

AKD™

User Guide



Edition: Revision C, April 2011

Valid for Hardware Revision C

Patents Pending

Part Number 903-200006-00

Original Documentation



Keep all manuals as a product component during the life span of the product.
Pass all manuals to future users/owners of the product.

KOLLMORGEN®

Record of Document Revisions:

| Revision | Remarks |
|----------------|--|
| 6/2010 | Combined User Guide and Parameter and Command Reference Guide (released in WorkBench only). |
| 9/2010, Rev- | Combined User Guide, CANopen, and EtherCAT manuals. Updated for Release 1.3. |
| 11/2010, Rev A | Usability improvements, minor topic edits. |
| 12/2010, Rev B | Document part number update. |
| 04/2011, Rev C | Updated for Release 1.4. Added Modbus, modulo, new homing modes, registration moves, W&S improved, new parameters, PLS enhanced, velocity unit changes, motion tasking revisions, switch bounce conditioning (DINx.FILTER), linear motor parameters. |

Hardware Revision (HR)

| Hardware Revision | Usable Firmware | Usable Workbench | Remarks |
|-------------------|------------------|------------------|---------------|
| A | M_01-03-00-zzz | 1.0.3.zzz | Start version |
| C | ≥ M_01-03-00-011 | ≥ 1.3.0.zzzzz | STO certified |

EnDat is a registered trademark of Dr. Johannes Heidenhain GmbH

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

HIPERFACE is a registered trademark of Max Stegmann GmbH

WINDOWS is a registered trademark of Microsoft Corporation

AKD is a registered trademark of Kollmorgen Corporation

Current patents:

US Patent 5,646,496 (used in control card R/D and 1 Vp-p feedback interface)

US Patent 5,162,798 (used in control card R/D)

US Patent 6,118,241 (used in control card simple dynamic braking)

Technical changes which improve the performance of the device may be made without prior notice.

Printed in the United States of America

This document is the intellectual property of Kollmorgen. All rights reserved. No part of this work may be reproduced in any form (by photocopying, microfilm or any other method) or stored, processed, copied or distributed by electronic means without the written permission of Kollmorgen.

Table of Contents

| | |
|---|-----------|
| 1 About this User Guide | 29 |
| 1.1 About this User Guide | 29 |
| 1.2 Abbreviations | 29 |
| 2 AKD Models | 31 |
| 2.1 CC Drive Models | 31 |
| 3 Basic Drive Setup | 33 |
| 3.1 Basic Drive Setup | 34 |
| 3.2 Display Codes | 34 |
| 3.3 AKD Setup Wizard | 35 |
| 4 Connecting the Drive | 37 |
| 4.1 Connected and Disconnected States | 38 |
| 4.2 Disconnected | 38 |
| 4.3 Device Not Shown | 38 |
| 4.4 Find and Enter IP Address | 39 |
| 4.5 Check Communications | 40 |
| 4.6 Connect To Another Drive | 40 |
| 5 Communicating with the Drive | 41 |
| 5.0.1 Overview | 42 |
| 5.0.2 Identifying the Drive IP Address | 42 |
| 5.0.2.1 Automatic (Dynamic) IP Addressing | 42 |
| 5.0.2.2 Static IP Addressing | 42 |
| 5.1 Check Communications | 43 |
| 5.2 Troubleshooting Connection and Communication Problems | 45 |
| 5.2.1 No Drives are Shown | 46 |
| 5.2.2 Device Not Shown | 46 |
| 5.2.3 Find and Enter IP Address | 46 |
| 5.3 TCP/IP Communication Protocols | 48 |
| 5.4 IP Address | 48 |
| 5.4.1 Port Number | 48 |
| 5.4.2 MAC Address | 48 |
| 6 Using WorkBench | 49 |
| 6.1 Welcome Screen | 49 |
| 6.1.1 Online | 49 |
| 6.1.2 Offline | 50 |
| 6.2 AKD Overview | 50 |
| 6.3 Online and Offline | 51 |
| 6.3.1 Online Drive | 51 |
| 6.3.2 Offline Drive | 51 |
| 6.3.3 Switching Between Online and Offline | 52 |
| 6.4 Watch | 52 |
| 6.5 Settings | 52 |
| 6.5.1 Navigation Tree | 52 |
| 6.5.2 Settings View | 52 |

| | | |
|----------|---|-----------|
| 7 | Configuring Drive Power | 55 |
| 7.1 | Power | 56 |
| 7.1.1 | Drive Setup for Power and Bus | 56 |
| 7.1.2 | Direct DC Mains Operation | 56 |
| 7.2 | Regeneration | 58 |
| 7.2.1 | Overview | 58 |
| 7.2.2 | Regen Resistor Options | 58 |
| 7.2.3 | Calculating Motor Peak Energy and Regen Resistor Size | 58 |
| 7.2.4 | Selecting a Compatible Regen Resistor | 61 |
| 7.2.5 | Configuring Regen Parameter Values | 61 |
| 7.2.6 | Related Parameters | 62 |
| 8 | Configuring Motor Settings | 63 |
| 8.1 | Motor | 64 |
| 8.1.1 | Overview | 64 |
| 8.1.2 | Motor Setup | 64 |
| 8.1.3 | Using the Motor View | 64 |
| 8.1.4 | Selecting a Motor | 65 |
| 8.1.4.1 | Configuring Custom Motors | 66 |
| 8.1.4.2 | Validating Motor Parameters | 66 |
| 8.2 | Feedback 1 | 66 |
| 8.2.1 | Overview | 67 |
| 8.3 | Using Feedback Options | 67 |
| 8.3.1 | Auto | 67 |
| 8.3.2 | Incremental Encoder | 67 |
| 8.3.3 | Sine Encoder | 67 |
| 8.3.4 | Endat 2.1, Endat 2.2 | 67 |
| 8.3.5 | BiSS | 67 |
| 8.3.6 | Hiperface | 67 |
| 8.3.7 | Resolver | 68 |
| 8.3.8 | SFD | 68 |
| 8.3.9 | Using Wake and Shake (WS) | 68 |
| 8.3.9.1 | Overview | 68 |
| 8.3.9.2 | Configuring WS | 68 |
| | Wake and Shake, More View | 69 |
| | Special Cases for WS | 70 |
| | Operation with Motor Brake | 70 |
| | End of Travel Limits | 70 |
| | Large Load Inertia or High Friction System | 70 |
| 8.3.9.3 | Using WS: Advanced | 70 |
| | Maximizing WS Reliability | 71 |
| 8.3.9.4 | Troubleshooting WS | 72 |
| 8.3.9.5 | Related Parameters | 73 |
| 8.3.10 | Commutation Alignment | 73 |
| 8.4 | Feedback 2 | 73 |

| | |
|---|-----------|
| 8.4.1 Encoder Emulation | 74 |
| 8.4.1.1 Overview | 74 |
| 8.4.1.2 Using Encoder Emulation | 74 |
| 8.4.1.3 Function Settings | 74 |
| Mode 0: Disable X9 | 74 |
| Output Modes 1 and 2 | 74 |
| Input Modes 3, 4, and 5 | 75 |
| 8.4.1.4 Resolution | 76 |
| 8.4.1.5 Related Parameters and Commands | 76 |
| 8.5 Non-Plug and Play Feedback Devices | 76 |
| 8.6 Parameters | 76 |
| 8.7 Calculations | 76 |
| 8.7.1 Current Loop | 77 |
| 8.7.2 Velocity Loop | 77 |
| 8.7.3 Position Loop | 77 |
| 8.7.4 Slider Tuning | 77 |
| 8.7.4.1 Input - Motor Data | 77 |
| 8.7.4.2 Constants | 77 |
| 8.7.4.3 Output - Control Loop Gains | 77 |
| 8.8 Foldback | 77 |
| 8.8.1 Drive Foldback | 78 |
| 8.8.2 Setting up motor foldback | 78 |
| 8.8.3 Setting Fault and Warning Levels | 78 |
| 8.8.4 Motor Peak Current Time | 79 |
| 8.8.5 Motor Foldback Ramp | 79 |
| 8.8.6 Motor Recovery | 80 |
| 8.9 Overall Foldback | 80 |
| 8.10 Brake | 80 |
| 8.11 Using Position Capture | 81 |
| 8.11.1 Overview | 81 |
| 8.11.2 Configuring Position Capture | 81 |
| 8.11.2.1 Setting the Capture Source (CAP0.TRIGGER) | 81 |
| 8.11.2.2 Setting the Capture Mode (CAP0.MODE) | 81 |
| 8.11.2.3 Arming and Retrieving the Capture Value (CAP0.EN and CAP0.T) | 82 |
| 8.11.2.4 Setting the Capture Edge (CAP0.EDGE) | 82 |
| 8.11.2.5 Setting the Pre-Condition Event: (CAP0.EVENT) | 82 |
| 8.11.2.6 Setting up a Pre-Condition for complex capture | 82 |
| Setting the Precondition Edge: (Terminal Command: CAP0.PREEDGE) | 82 |
| Setting the Pre-Condition Select: (Terminal Command: CAP0.PRESELECT) | 82 |
| 8.11.3 Related Parameters | 82 |
| 9 Selecting Units for Your Application | 83 |
| 9.1 Selecting and Saving Units | 84 |
| 9.2 Units Example | 84 |
| 9.2.1 Related Parameters | 86 |

| | | |
|-----------|---|-----------|
| 10 | Configuring General Drive Settings | 87 |
| 10.1 | Digital Inputs and Outputs | 88 |
| 10.1.1 | Overview | 88 |
| 10.1.2 | Using Digital I/O | 88 |
| 10.1.3 | Digital Inputs | 88 |
| 10.1.4 | Digital Outputs | 94 |
| 10.2 | Command Buffer | 96 |
| 10.2.1 | Overview | 96 |
| 10.2.2 | Editing the Command Buffers | 97 |
| 10.2.3 | Behavior of the Command Buffer | 98 |
| 10.2.4 | Delays for the Buffer | 99 |
| 10.3 | Digital Inputs (X7/X8) | 99 |
| 10.3.1 | Digital Inputs 1 and 2 | 102 |
| 10.3.2 | Digital Inputs 3 to 7 | 102 |
| 10.3.3 | Digital Input 8 (ENABLE) | 102 |
| 10.4 | Analog Input | 103 |
| 10.4.1 | Analog Input | 103 |
| 10.4.2 | Related Parameters and Commands | 103 |
| 10.5 | Analog Output | 103 |
| 10.6 | I/O Connection | 104 |
| 10.6.1 | I/O Connectors (X7 and X8) | 105 |
| 10.7 | Electronic Gearing | 106 |
| 10.7.1 | Overview | 106 |
| 10.7.2 | Limits | 106 |
| 10.7.3 | Determining Maximum Cable Length | 107 |
| 10.7.4 | Related Parameters: | 108 |
| 10.7.5 | Limits | 109 |
| 10.8 | Programmable Limit Switch | 109 |
| 10.8.1 | Overview | 109 |
| 10.8.2 | Using Programmable Limit Switches | 109 |
| 10.8.3 | Single Shot Mode | 111 |
| 10.8.4 | Related Parameters and Commands | 111 |
| 10.9 | Enable/Disable | 111 |
| 10.9.1 | Enable Modes | 111 |
| 10.9.1.1 | Hardware Enable Mode | 111 |
| 10.9.1.2 | Software Enable Default | 112 |
| 10.9.2 | Disable Modes | 112 |
| 10.9.3 | Drive Status | 112 |
| 10.9.4 | Controlled Stop | 113 |
| 10.9.5 | More/Less Button | 113 |
| 10.10 | Controlled Stop | 115 |
| 10.10.0.1 | Related Parameters and Commands | 116 |
| 10.10.1 | Related topics: | 117 |
| 10.11 | Dynamic Braking | 117 |
| 10.11.1 | Drive Regeneration | 117 |
| 10.11.1.1 | AKD-x00306 to AKD-x00606 | 117 |

| | | |
|--|--|------------|
| 10.11.1.2 | AKD-x01206 to AKD-x02406 and AKD-xzzz07..... | 117 |
| 10.12 | Emergency Stop..... | 118 |
| 10.13 | Safe Torque Off (STO)..... | 118 |
| 10.14 | Under Voltage Fault Behavior..... | 118 |
| 11 | Using Command Source and Operating Modes..... | 119 |
| 11.1 | Overview..... | 120 |
| 11.2 | Using Command Source and Operation Modes..... | 120 |
| 11.2.1 | Command Source..... | 120 |
| 11.2.1.1 | Service..... | 120 |
| 11.2.1.2 | Fieldbus..... | 120 |
| 11.2.1.3 | Electronic Gearing..... | 120 |
| 11.2.1.4 | Analog..... | 120 |
| 11.2.1.5 | Operation Mode..... | 120 |
| 11.2.2 | Related Parameters..... | 120 |
| 11.3 | Current Loop..... | 121 |
| 11.3.1 | Overview..... | 121 |
| 11.3.2 | Current Loop Gain..... | 121 |
| 11.3.3 | Current Loop Gain Scheduling..... | 122 |
| 11.3.3.1 | Using the Gain Scheduling View in WorkBench..... | 122 |
| 11.3.3.2 | Using the Terminal View for Gains Scheduling..... | 123 |
| 11.4 | Velocity Loop..... | 124 |
| 11.4.1 | Overview..... | 124 |
| 11.4.2 | Tabs in the Velocity Loop View..... | 124 |
| 11.4.3 | Velocity Loop Default Settings and Changes..... | 125 |
| 11.4.3.1 | Velocity Loop Changes Based on Slider Tuning..... | 125 |
| 11.4.3.2 | Velocity Loop Changes Based on PST..... | 126 |
| 11.5 | Position Loop..... | 126 |
| 11.5.1 | Overview..... | 126 |
| 11.5.2 | Tabs in the Position Loop View..... | 126 |
| 11.5.3 | Position Loop Default Behavior and Changes..... | 127 |
| 11.5.3.1 | Position Loop Changes Based on Slider Tuning..... | 127 |
| 11.5.3.2 | Position Loop Changes Based on PST..... | 127 |
| 11.5.4 | Modulo Position..... | 127 |
| 11.5.4.1 | Setting up the modulo axis in WorkBench..... | 128 |
| 11.5.4.2 | Setting up the modulo axis from the Terminal..... | 128 |
| 11.5.4.3 | Parameters affected by the modulo axis..... | 128 |
| 11.5.4.4 | Drive functions affected by modulo axis..... | 129 |
| Software limit switch..... | | 129 |
| Programmable limit switch..... | | 129 |
| Digital output Modes 5 and 6..... | | 129 |
| Motion tasking to absolute target positions..... | | 129 |
| 11.5.4.5 | Using the modulo position feature with multiturn encoders..... | 129 |
| 12 | Creating Motion..... | 133 |
| 12.1 | Homing..... | 134 |
| 12.1.1 | Overview..... | 134 |

| | |
|--|-----|
| 12.1.2 Using Homing..... | 134 |
| 12.1.2.1 Home Default Window..... | 134 |
| Mode Selection:..... | 135 |
| Settings:..... | 135 |
| Controls:..... | 136 |
| 12.1.2.2 Selecting and Using Homing Modes..... | 136 |
| Homing Mode 0: Home Using Current Position..... | 136 |
| Homing Mode 1: Find Limit Input..... | 136 |
| Homing Mode 2: Find Input Limit then Find Zero Angle..... | 137 |
| Homing Mode 3: Find Input Limit then Find Index..... | 138 |
| Homing Mode 4: Find Home Input..... | 139 |
| Homing Mode 5: Find Home Input then Find Zero Angle..... | 139 |
| Homing Mode 6: Find Home Input then Find Index..... | 140 |
| Homing Mode 7: Find Zero Angle..... | 141 |
| Homing mode 8: Move Until Position Error Exceeded..... | 141 |
| Homing Mode 9: Move Until Position Error Exceeded then Find Zero Angle..... | 141 |
| 12.1.2.3 Homing Mode 10: Move Until Position Error Exceeded then Find Index..... | 142 |
| Homing Mode 11: Find Index Signal..... | 143 |
| 12.1.3 Using Homing: Advanced..... | 144 |
| 12.1.3.1 Related Parameters and Commands..... | 144 |
| 12.1.3.2 Related Topics..... | 144 |
| 12.2 Motion Tasks..... | 144 |
| 12.2.1 Overview..... | 144 |
| 12.2.2 Motion Task Input Table..... | 144 |
| 12.2.3 Using Motion Tasks..... | 145 |
| 12.2.3.1 Motion Profiles..... | 147 |
| 12.2.3.2 Motion Types..... | 147 |
| Absolute motion task..... | 148 |
| Motion task relative to command position (PL.CMD)..... | 148 |
| Motion task relative to previous target position..... | 148 |
| 12.2.4 Using Motion Tasks: Advanced..... | 148 |
| 12.2.4.1 Joining multiple tasks..... | 148 |
| 12.2.4.2 Start Conditions..... | 149 |
| 12.2.4.3 Blending..... | 149 |
| 12.3 Registration Moves (Motion Task Blending)..... | 149 |
| 12.3.1 Configuring Registration Moves in WorkBench..... | 150 |
| 12.3.2 Configuring Registration Moves from the Terminal View..... | 150 |
| 12.4 Service Motion..... | 152 |
| 12.5 Jog Move..... | 153 |
| 12.6 Drive Motion Status..... | 153 |

| | |
|--|------------|
| 13 Saving Your Drive Configuration | 155 |
| 13.1 Save Options | 156 |
| 13.2 Save On Exit | 156 |
| 13.3 Save On Disconnect | 157 |
| 13.4 Save On Firmware Download | 158 |
| 14 Tuning Your System | 159 |
| 14.1 Slider Tuning | 160 |
| 14.1.0.1 Gentle, Medium, and Stiff | 160 |
| 14.1.0.2 The Slider | 160 |
| 14.1.0.3 Inertia Ratio | 160 |
| 14.2 Load Inertia | 160 |
| 14.3 Using the Performance Servo Tuner | 160 |
| 14.3.1 Overview | 160 |
| 14.3.2 Using the PST | 161 |
| 14.3.2.1 Saving and Emailing Bode Plots | 162 |
| 14.3.2.2 Importing a Frequency Response | 163 |
| 14.3.3 Measurement Options | 163 |
| 14.3.3.1 Using Manual Excitation Levels | 163 |
| 14.3.3.2 Taking a Bode Measurement without the PST | 164 |
| 14.3.4 Using the Performance Servo Tuner: Advanced | 164 |
| 14.3.4.1 Typical Cases for Advanced PST Use | 165 |
| Tuning Systems with Low-Frequency Resonances | 165 |
| Tuning Systems with High-Frequency Resonances | 166 |
| Tuning systems with noisy frequency responses | 167 |
| 14.3.4.2 PST Options | 169 |
| Enable BiQuad 1 Tuner | 170 |
| Enable BiQuad 2 Tuner | 170 |
| Biquad Type | 170 |
| Tune Acceleration FF | 170 |
| Enable Stability Motion | 170 |
| Phase and Gain Margins | 170 |
| Tune Velocity Integral | 170 |
| Enable Low Pass Search | 170 |
| 14.3.4.3 Measurement Options | 170 |
| Current Amplitude | 171 |
| Velocity Amplitude | 171 |
| Smooth % | 171 |
| Measurement | 174 |
| Injection Point | 174 |
| Excitation Type | 175 |
| FFT Points | 175 |

| | |
|--|------------|
| Excite Gap..... | 175 |
| Number Points..... | 175 |
| Velocity Max..... | 176 |
| 14.3.4.4 Plot Options..... | 177 |
| Coherence..... | 177 |
| 14.3.4.5 Cursors..... | 177 |
| 14.3.4.6 Resizing Bode Plots..... | 179 |
| Reading and Understanding the Bode Plot..... | 180 |
| Using the Performance Servo Tuner to Manually Tune Systems..... | 183 |
| Using the Tuning Simulation..... | 184 |
| Using the Performance Servo Tuner Manual Tuning Interface..... | 184 |
| Simulating Modified Loop Gains with the Performance Servo Tuner..... | 185 |
| Simulating Filters with the Performance Servo Tuner..... | 188 |
| Using Filters to Reduce Noise..... | 189 |
| 14.4 Tuning Guide..... | 189 |
| 14.4.1 Overview..... | 189 |
| 14.4.2 Determining Tuning Criteria..... | 190 |
| 14.4.3 Before You Tune..... | 190 |
| 14.4.4 Closed Loop Tuning Methods..... | 190 |
| 14.4.4.1 Tuning the Velocity Loop..... | 191 |
| 14.4.4.2 Tuning the Position Loop..... | 193 |
| 14.4.5 Torque Feedforward Tuning Methods..... | 193 |
| 14.4.5.1 Shape Based Feedforward Tuning..... | 193 |
| 14.4.6 Using Anti-Resonance Filters..... | 194 |
| 14.4.6.1 Biquad Calculations..... | 202 |
| 14.4.6.2 Common Uses Of Anti Resonance Filters..... | 204 |
| 15 Scope..... | 205 |
| 15.1 Overview..... | 205 |
| 15.2 Using the Scope..... | 205 |
| 15.2.1 Scope Channels Tab..... | 205 |
| 15.2.1.1 Source Column..... | 205 |
| 15.2.1.2 Color Column..... | 206 |
| 15.2.1.3 Hide Column..... | 206 |
| 15.2.1.4 Y-Axis Column..... | 206 |
| 15.2.1.5 Filter and Filter Frequency Column..... | 206 |
| 15.2.2 Scope Time-base and Trigger Tab..... | 206 |
| 15.2.2.1 Scope Time-base and Trigger, More View..... | 207 |
| 15.2.2.2 Trigger Type..... | 208 |
| 15.2.2.3 Trigger Position..... | 208 |
| 15.2.2.4 Trigger Value..... | 209 |
| 15.2.2.5 Effects of Recorder Gap..... | 210 |
| 15.2.2.6 Trigger Slope..... | 211 |

| | | |
|-----------|---|------------|
| 15.3 | Scope Settings | 211 |
| 15.3.1 | Load a setting (preset) to Scope screen | 212 |
| 15.3.2 | Create a new preset | 212 |
| 15.3.3 | Save or delete preset | 212 |
| 15.3.4 | Import preset | 212 |
| 15.3.5 | Export preset | 213 |
| 15.3.6 | Scope axis scaling and zooming | 214 |
| 15.3.7 | Manual range per axis | 215 |
| 15.3.8 | Unit display on Y axis | 215 |
| 16 | Using Parameters and the Terminal Screen | 217 |
| 16.1 | Terminal | 218 |
| 16.1.1 | Overview | 218 |
| 16.1.2 | Using the Terminal | 218 |
| 16.1.3 | Macros | 219 |
| 16.1.3.1 | Creating a Macro from Terminal commands | 219 |
| 16.1.3.2 | Macro Editor | 220 |
| 16.2 | Viewing Parameters | 221 |
| 16.3 | Parameter List | 221 |
| 16.4 | Parameter Load/Save | 221 |
| 16.5 | Parameter Comparer | 221 |
| 16.5.1 | Reference Parameter Selection | 222 |
| 16.5.2 | Target Parameter Selection | 222 |
| 16.5.3 | Display the comparison | 223 |
| 16.5.4 | Motion Task Comparison | 224 |
| | Summary of Parameters and Commands | 225 |
| 17 | Faults and Warnings | 239 |
| 17.1 | Fault and Warning Messages | 240 |
| 17.2 | Clearing Faults | 249 |
| 17.3 | Parameter and Command Error Messages | 250 |
| 17.4 | CANopen Emergency Messages and Error Codes | 253 |
| 17.5 | Unknown Fault | 256 |
| 17.5.0.1 | Remedies | 256 |
| 18 | Troubleshooting the AKD | 257 |
| 19 | Firmware and Firmware Updates | 258 |
| 19.1 | Downloading Firmware | 259 |
| 19.2 | Firmware Compatibility | 259 |
| 19.3 | Invalid Firmware | 259 |
| 19.4 | Forcing the drive into firmware download mode | 260 |
| 19.5 | Forced download of AKD firmware | 260 |
| | Connection Diagrams | 263 |
| 19.6 | Connection Diagram, AKD-x00306 to x00606 | 264 |
| 19.7 | Connection Diagram, AKD-x01206 | 265 |
| 19.8 | Connection Diagram, AKD-x02406 and AKD-xzzz07 | 266 |
| 19.9 | 24 V Auxiliary Supply (X1) | 267 |
| 19.10 | Motor Connection | 268 |
| 19.11 | External Regen Resistor (X3) | 269 |

| | |
|---|------------|
| 19.12 DC Bus Link (X3)..... | 270 |
| 19.13 Mains Supply Connection (X3, X4)..... | 271 |
| 19.13.1 Three Phase connection (all AKD types)..... | 271 |
| 19.13.2 Single phase connection (AKD-xzzz06 only)..... | 272 |
| 19.14 I/O Connection..... | 273 |
| 19.14.1 I/O Connectors (X7 and X8)..... | 273 |
| 19.15 Analog Output (X8)..... | 274 |
| 19.16 Analog Input (X8)..... | 275 |
| 19.17 Command encoder signal connection..... | 276 |
| 19.17.1 Incremental encoder input 5 V (X9)..... | 276 |
| 19.17.2 Incremental encoder input 24 V (X7)..... | 276 |
| 19.17.3 Emulated Encoder Output (EEO) - A quad B (X9)..... | 277 |
| 19.18 Pulse / Direction signal connection..... | 278 |
| 19.18.1 Pulse / Direction input 5 V (X9)..... | 278 |
| 19.18.2 Pulse / Direction input 24 V (X7)..... | 278 |
| 19.19 Up / Down signal connection..... | 278 |
| 19.19.1 Up / Down input 5 V (X9)..... | 278 |
| 19.19.2 Up / Down input 24 V (X7)..... | 279 |
| 19.20 Feedback Connector (X10)..... | 279 |
| Block Diagrams | 281 |
| 19.21 Block Diagram for Current Loop..... | 282 |
| 19.22 Block Diagram for Position/Velocity Loop..... | 282 |
| Appendix A | 1 |
| 20 Modbus | 3 |
| 20.1 Overview..... | 3 |
| 20.2 Installation and Setup..... | 3 |
| 20.3 Overview of Messaging..... | 3 |
| 20.4 Supported Functions..... | 3 |
| 20.5 Read Holding Registers (0x03)..... | 3 |
| 20.6 Write Multiple Registers (0x10)..... | 4 |
| 20.7 Exception Response Codes..... | 5 |
| 20.8 Modbus..... | 6 |
| 20.8.1 Overview..... | 6 |
| 20.8.2 Modbus Installation and Setup..... | 6 |
| 20.8.3 Overview of Messaging..... | 6 |
| 20.8.4 Supported Functions..... | 6 |
| 20.8.5 Read Holding Registers (0x03)..... | 6 |
| 20.8.6 Write Multiple Registers (0x10)..... | 8 |
| 20.8.7 Exception Response Codes..... | 8 |
| 20.9 Modbus Dynamic Mapping..... | 9 |
| 20.9.1 Configuring Dynamic Mapping..... | 9 |
| 20.9.2 Saving and Resetting Dynamic Mapping..... | 10 |
| 20.9.3 Scaling parameters..... | 10 |
| 20.9.4 Modbus specific registers (Parameters)..... | 11 |
| 20.9.5 32-bit versus 16-bit Values..... | 11 |
| 20.9.6 Mapping of 64-bit Parameters to 32-bit Parameters..... | 11 |

| | | |
|-----------|--|-----------|
| 20.9.7 | Fault Registers..... | 12 |
| 20.9.8 | Mapping Table..... | 12 |
| 20.10 | Modbus Parameter Table..... | 13 |
| 20.11 | Modbus 64-bit Parameters to 32-bit Mapping..... | 23 |
| 21 | Ethercat Manual..... | 27 |
| 21.1 | General..... | 58 |
| 21.1.1 | About this Manual..... | 59 |
| 21.1.2 | Target Group..... | 59 |
| 21.1.3 | Abbreviations Used..... | 60 |
| 21.2 | Safety..... | 61 |
| 21.2.1 | Safety Instructions..... | 62 |
| 21.2.2 | Use as directed..... | 62 |
| 21.2.3 | Prohibited use..... | 62 |
| 21.3 | Installation and Setup..... | 63 |
| 21.3.1 | Safety Instructions..... | 64 |
| 21.3.2 | EtherCAT Onboard..... | 65 |
| 21.3.2.1 | LED functions..... | 65 |
| 21.3.2.2 | Connection technology..... | 65 |
| 21.3.2.3 | Network Connection Example..... | 65 |
| 21.3.3 | Guide to Setup..... | 66 |
| 21.3.4 | Setup via TwinCAT NC/PTP System Manager..... | 66 |
| 21.3.4.1 | Scan devices..... | 67 |
| 21.3.4.2 | Select the device..... | 67 |
| 21.3.4.3 | Scan for boxes..... | 68 |
| 21.3.4.4 | Add Slaves to NC tasks..... | 68 |
| 21.3.4.5 | Enable the network configuration..... | 69 |
| 21.3.4.6 | Enable the axis and move the axis..... | 70 |
| 21.3.5 | Setup WorkBench over TwinCAT..... | 71 |
| 21.3.5.1 | TwinCAT and WorkBenchconfiguration..... | 72 |
| 21.3.5.2 | Connecting to a drive using WorkBench..... | 73 |
| 21.3.5.3 | Configuring and enabling a drive..... | 76 |
| 21.3.6 | Setup via KAS IDE..... | 77 |
| 21.4 | EtherCAT Profile..... | 78 |
| 21.4.1 | Slave Register..... | 79 |
| 21.4.2 | AL Event (Interrupt Event) and Interrupt Enable..... | 80 |
| 21.4.2.1 | Interrupt Enable Register (Address 0x0204:0x0205)..... | 80 |
| 21.4.2.2 | AL Event Request (Address 0x0220:0x0221)..... | 81 |
| 21.4.3 | Phase Run-Up..... | 82 |
| 21.4.3.1 | AL Control (Address 0x0120:0x0121)..... | 82 |
| 21.4.3.2 | AL Status (Address 0x0130:0x0131)..... | 82 |
| 21.4.3.3 | AL Status Code (Address 0x0134:0x0135)..... | 83 |
| 21.4.3.4 | EtherCAT communication phases..... | 83 |
| 21.4.4 | CANopen over EtherCAT (CoE) Status Machine..... | 84 |
| 21.4.4.1 | Status Description..... | 84 |
| 21.4.4.2 | Commands in the Control Word..... | 85 |
| 21.4.4.3 | Status Machine Bits (status word)..... | 86 |

| | | |
|-----------|---|------------|
| 21.4.5 | Fixed PDO Mappings..... | 87 |
| 21.4.6 | Supported Cyclical Setpoint and Actual Values..... | 88 |
| 21.4.7 | Supported Operation Modes..... | 88 |
| 21.4.8 | Adjusting EtherCAT Cycle Time..... | 89 |
| 21.4.9 | Maximum Cycle Times depending on operation mode..... | 89 |
| 21.4.10 | Synchronization..... | 90 |
| 21.4.10.1 | Synchronization behavior with distributed clocks (DC) enabled..... | 90 |
| 21.4.10.2 | Synchronization behavior with distributed clocks (DC) disabled..... | 90 |
| 21.4.11 | Latch Control Word and Latch Status Word..... | 91 |
| 21.4.12 | Mailbox Handling..... | 92 |
| 21.4.12.1 | Mailbox Output..... | 93 |
| 21.4.12.2 | Mailbox Input..... | 94 |
| 21.4.12.3 | Example: Mailbox Access..... | 95 |
| 21.4.13 | Fieldbus Parameters..... | 96 |
| 21.5 | Index..... | 98 |
| 22 | CANopen Manual..... | 104 |
| 22.1 | General..... | 135 |
| 22.1.1 | About this Manual..... | 136 |
| 22.1.2 | Target Group..... | 136 |
| 22.1.3 | Abbreviations used..... | 137 |
| 22.2 | Safety..... | 138 |
| 22.2.1 | Safety Instructions..... | 139 |
| 22.2.2 | Use As Directed..... | 139 |
| 22.2.3 | Prohibited Use..... | 139 |
| 22.3 | Installation and Setup..... | 140 |
| 22.3.1 | Safety Instructions..... | 141 |
| 22.3.2 | Guide to Setup..... | 142 |
| 22.4 | CANopen Basics..... | 142 |
| 22.4.1 | Basic Features implemented by CANopen..... | 143 |
| 22.4.1.1 | Setup and general functions:..... | 143 |
| 22.4.1.2 | Positioning functions:..... | 143 |
| 22.4.1.3 | Data transfer functions:..... | 143 |
| 22.4.2 | Transmission Rate and Procedure..... | 143 |
| 22.4.3 | Response to BUSOFF Communication Faults..... | 144 |
| 22.4.4 | Important Configuration Parameters..... | 144 |
| 22.5 | CANopen Communication Profile..... | 144 |
| 22.5.1 | General Description of CAN..... | 145 |
| 22.5.2 | Construction of the Communication Object Identifier..... | 146 |
| 22.5.3 | Definition of the Used Data Types..... | 147 |
| 22.5.3.1 | Basic data types..... | 147 |
| | Unsigned Integer..... | 147 |
| | Signed Integer..... | 148 |
| 22.5.3.2 | Mixed data types..... | 148 |
| 22.5.3.3 | Extended data types..... | 149 |
| | Octet String..... | 149 |

| | |
|--|-----|
| Visible String..... | 149 |
| 22.5.4 Communication Objects..... | 149 |
| 22.5.4.1 Network Management Objects (NMT)..... | 150 |
| 22.5.4.2 Synchronization Object (SYNC)..... | 150 |
| 22.5.4.3 Time-Stamp Object (TIME)..... | 150 |
| 22.5.4.4 Emergency Object (EMCY)..... | 151 |
| Application of the Emergency Object..... | 151 |
| Composition of the Emergency Object..... | 151 |
| 22.5.4.5 Service Data Objects (SDO)..... | 152 |
| Composition of the Service Data Object..... | 152 |
| Initiate SDO Download Protocol..... | 154 |
| Download SDO Segment Protocol..... | 154 |
| Initiate SDO Upload Protocol..... | 154 |
| Upload SDO Segment Protocol..... | 154 |
| Abort SDO Protocol..... | 154 |
| 22.5.4.6 Process Data Object (PDO)..... | 155 |
| Transmission modes..... | 156 |
| Trigger modes..... | 156 |
| 22.5.4.7 Nodeguard..... | 157 |
| 22.5.4.8 Heartbeat..... | 158 |
| 22.6 CANopen Drive Profile..... | 159 |
| 22.6.1 CANopen Emergency Messages and Error Codes..... | 160 |
| 22.6.2 General Definitions..... | 163 |
| 22.6.2.1 General objects..... | 163 |
| Object 1000h: Device Type (DS301)..... | 163 |
| Object 1001h: Error register (DS301)..... | 164 |
| Object 1002h: Manufacturer Status Register (DS301)..... | 165 |
| Object 1003h: Predefined Error Field (DS301)..... | 166 |
| Object 1005h: COB-ID of the SYNC Message (DS301)..... | 167 |
| Object 1006h: Communication Cycle Period (DS301)..... | 167 |
| Object 1008h: Manufacturer Device Name (DS301)..... | 168 |
| Object 1009h: Manufacturer Hardware Version..... | 168 |
| Object 100Ah: Manufacturer Software Version (DS301)..... | 168 |
| Object 100Ch: Guard Time (DS301)Response monitoring..... | 169 |
| Object 100Dh: Lifetime Factor (DS301)..... | 169 |
| Object 1010h: Store Parameters (DS301)..... | 170 |
| Object 1014h: COB-ID for Emergency Message (DS301)..... | 171 |
| Object 1016h: Consumer Heartbeat Time..... | 171 |

| | |
|--|-----|
| Object 1017h: Producer Heartbeat Time..... | 172 |
| Object 1018h: Identity Object (DS301)..... | 172 |
| Object 1026h: OS Prompt..... | 174 |
| 22.6.2.2 Manufacturer specific objects..... | 175 |
| Object 2014-2017h: 1st-4th Mask 1 to 4 for Transmit-PDO..... | 175 |
| Object 2018h: Firmware Version..... | 176 |
| Object 2026h: ASCII Channel..... | 177 |
| Object 20A0h: Latch position 1, positive edge..... | 178 |
| Object 20A1h: Latch position 1, negative edge..... | 178 |
| Object 20A2h: Latch position 2, positive edge..... | 178 |
| Object 20A3h: Latch position 2, negative edge..... | 179 |
| Object 20A4h: Latch Control Register..... | 179 |
| Object 20A5h: Latch Status Register..... | 180 |
| Object 20A6h: Latch position 1, positive or negative edge..... | 180 |
| Object 20B8h: Reset of changed input information..... | 181 |
| 22.6.2.3 Profile specific objects..... | 182 |
| Object 60FDh: Digital inputs (DS402)..... | 182 |
| Object 60FEh: Digital outputs (DS402)..... | 182 |
| Object 6502h: Supported drive modes (DS402)..... | 183 |
| 22.6.3 PDO Configuration..... | 184 |
| 22.6.3.1 Receive PDOs (RXPDO)..... | 185 |
| Objects 1400-1403h: 1st - 4th RXPDO communication parameter (DS301)..... | 185 |
| Objects 1600-1603h: 1st - 4th RXPDO mapping parameter (DS301)..... | 186 |
| Default RXPDO definition..... | 187 |
| 22.6.3.2 Transmit PDOs (TXPDO)..... | 188 |
| Objects 1800-1803h: 1st - 4th TXPDO communication parameter (DS301)..... | 188 |
| Objects 1A00-1A03h: 1st - 4th TXPDO mapping parameter (DS301)..... | 190 |
| Default TXPDO definition..... | 191 |
| 22.6.4 Device Control (dc)..... | 192 |
| 22.6.4.1 Status Machine (DS402)..... | 192 |
| States of the Status Machine..... | 193 |
| Transitions of the status machine..... | 194 |
| 22.6.4.2 Object Description..... | 195 |
| Object 6040h: Control word (DS402)..... | 195 |
| Object 6041h: Status word (DS402)..... | 196 |
| Object 6060h: Modes of Operation (DS402)..... | 198 |

| | |
|---|-----|
| Object 6061h: Modes of Operation Display (DS402)..... | 199 |
| 22.6.5 Factor Groups (fg) (DS402)..... | 199 |
| 22.6.5.1 General Information..... | 199 |
| Factors..... | 199 |
| Relationship between Physical and Internal Units..... | 199 |
| 22.6.5.2 Objects for velocity scaling..... | 200 |
| Object 204Ch: pv scaling factor..... | 200 |
| 22.6.5.3 Objects for position calculation..... | 201 |
| Object 608Fh: Position encoder resolution (DS402)..... | 201 |
| Object 6092h: Feed constant (DS402)..... | 202 |
| 22.6.6 Profile Velocity Mode (pv) (DS402)..... | 203 |
| 22.6.6.1 General Information..... | 203 |
| Objects that are defined in this section..... | 203 |
| Objects that are defined in other sections..... | 203 |
| 22.6.6.2 Object description..... | 203 |
| Object 606Ch: velocity actual value (DS402)..... | 203 |
| Object 60FFh: target velocity (DS402)..... | 204 |
| 22.6.7 Profile Torque Mode (tq) (DS402)..... | 205 |
| 22.6.7.1 General Information..... | 205 |
| Objects that are defined in this section..... | 205 |
| Objects that are defined in other sections..... | 205 |
| 22.6.7.2 Object description..... | 205 |
| Object 6071h: Target torque (DS402)..... | 205 |
| Object 6073h: Max current (DS402)..... | 205 |
| Object 6077h: Torque actual value (DS402)..... | 206 |
| 22.6.8 Position Control Function (pc) (DS402)..... | 207 |
| 22.6.8.1 General Information..... | 207 |
| Objects that are defined in this section..... | 207 |
| Objects that are defined in other sections..... | 207 |
| 22.6.8.2 Object Description..... | 207 |
| Object 6063h: position actual value* (DS402)..... | 207 |
| Object 6064h: position actual value (DS402)..... | 208 |
| Object 6065h: Following error window..... | 208 |
| Object 60F4h: Following error actual value (DS402)..... | 208 |
| 22.6.9 Interpolated Position Mode (ip) (DS402)..... | 209 |
| 22.6.9.1 General information..... | 209 |
| Objects defined in this section..... | 209 |

| | |
|--|-----|
| Objects defined in other sections..... | 209 |
| 22.6.9.2 Object description..... | 209 |
| Object 60C0h: Interpolation sub mode select..... | 209 |
| Object 60C1h: Interpolation data record..... | 210 |
| Object 60C2h: Interpolation time period..... | 211 |
| Object 60C4h: Interpolation data configuration..... | 212 |
| 22.6.10 Homing Mode (hm) (DS402)..... | 214 |
| 22.6.10.1 General information..... | 214 |
| Objects that are defined in this section..... | 214 |
| Objects that are defined in other sections..... | 214 |
| 22.6.10.2 Object Description..... | 214 |
| Object 607Ch: homing offset (DS402)..... | 214 |
| Object 6098h: homing method (DS402)..... | 214 |
| Object 6099h: homing speeds (DS402)..... | 216 |
| Object 609Ah: homing acceleration (DS402)..... | 216 |
| Homing Mode Sequence..... | 217 |
| 22.6.11 Profile Position Mode (pp)..... | 218 |
| 22.6.11.1 General Information..... | 218 |
| Objects that are defined in this section..... | 218 |
| Objects that are defined in other sections..... | 218 |
| 22.6.11.2 Object Description..... | 219 |
| Object 607Ah: target position (DS402)..... | 219 |
| Object 607Dh: Software position limit (DS402)..... | 219 |
| Object 6081h: profile velocity (DS402)..... | 220 |
| Object 6083h: profile acceleration (DS402)..... | 220 |
| Object 6084h: profile deceleration (DS402)..... | 220 |
| Functional Description..... | 221 |
| 22.7 Appendix..... | 223 |
| 22.7.1 Object Dictionary..... | 223 |
| 22.7.1.1 Float Scaling..... | 223 |
| 22.7.1.2 Communication SDOs..... | 223 |
| 22.7.1.3 Manufacturer specific SDOs..... | 226 |
| 22.7.1.4 Profile specific SDOs..... | 234 |
| 22.7.2 Examples..... | 236 |
| 22.7.2.1 Examples, setup..... | 236 |
| Basic testing of the connection to the AKD controls..... | 236 |
| Example: Operating the Status Machine..... | 237 |
| Example: Jog Mode via SDO..... | 238 |

| | |
|---|------------|
| Example: Torque Mode via SDO..... | 238 |
| Example: Jog Mode via PDO..... | 239 |
| Example: Torque Mode via PDO..... | 241 |
| Example: Homing via SDO..... | 242 |
| Example: Using the Profile Position Mode..... | 244 |
| Example: ASCII Communication..... | 247 |
| Test for SYNC telegrams..... | 248 |
| 22.7.2.2 Examples, special applications..... | 249 |
| Example: External Trajectory with Interpolated Position Mode..... | 249 |
| 22.8 Index..... | 255 |
| Appendix B..... | 261 |
| | 262 |
| About the Parameter and Command Reference Guide..... | 263 |
| Parameter and Command Naming Conventions..... | 265 |
| Summary of Parameters and Commands..... | 265 |
| AIN Parameters..... | 279 |
| AIN.CUTOFF..... | 280 |
| AIN.DEADBAND..... | 281 |
| AIN.DEADBANDMODE..... | 283 |
| AIN.ISCALE..... | 285 |
| AIN.OFFSET..... | 286 |
| AIN.PSCALE..... | 287 |
| AIN.VALUE..... | 288 |
| AIN.VSCALE..... | 289 |
| AIN.ZERO..... | 290 |
| AIO Parameters..... | 291 |
| AIO.ISCALE..... | 291 |
| AIO.PSCALE..... | 292 |
| AIO.VSCALE..... | 293 |
| AOUT Parameters..... | 295 |
| AOUT.DEBUGADDR..... | 295 |
| AOUT.DEBUGSCALE..... | 296 |
| AOUT.ISCALE..... | 297 |
| AOUT.MODE..... | 298 |
| AOUT.OFFSET..... | 299 |
| AOUT.PSCALE..... | 300 |
| AOUT.VALUE..... | 302 |
| AOUT.VALUEU..... | 303 |
| AOUT.VSCALE..... | 304 |
| BODE Parameters..... | 305 |
| BODE.EXCITEGAP..... | 305 |
| BODE.FREQ..... | 306 |
| BODE.IAMP..... | 307 |
| BODE.IFLIMIT..... | 308 |

| | |
|---|------------|
| BODE.IFTHRESH..... | 309 |
| BODE.INJECTPOINT..... | 310 |
| BODE.MODE..... | 311 |
| BODE.MODETIMER..... | 314 |
| BODE.PRBDDEPTH..... | 316 |
| BODE.VAMP..... | 317 |
| BODE.VFLIMIT..... | 318 |
| BODE.VFTHRESH..... | 319 |
| CAP Parameters..... | 320 |
| CAP0.EDGE, CAP1.EDGE..... | 320 |
| CAP0.EN, CAP1.EN..... | 321 |
| CAP0.EVENT, CAP1.EVENT..... | 322 |
| CAP0.FILTER, CAP1.FILTER..... | 325 |
| CAP0.MODE, CAP1.MODE..... | 326 |
| CAP0.PLFB, CAP1.PLFB..... | 327 |
| CAP0.PREEDGE, CAP1.PREEDGE..... | 328 |
| CAP0.PREFILTER, CAP1.PREFILTER..... | 329 |
| CAP0.PRESELECT, CAP1.PRESELECT..... | 330 |
| CAP0.STATE, CAP1.STATE..... | 331 |
| CAP0.T, CAP1.T..... | 332 |
| CAP0.TRIGGER, CAP1.TRIGGER..... | 333 |
| CS Parameters..... | 334 |
| CS.DEC..... | 334 |
| CS.STATE..... | 336 |
| CS.TO..... | 337 |
| CS.VTHRESH..... | 337 |
| DIN Parameters..... | 339 |
| DIN.HCMD1 TO DIN.HCMD4..... | 340 |
| DIN.LCMD1 to DIN.LCMD4..... | 341 |
| DIN.ROTARY..... | 341 |
| DIN.STATES..... | 343 |
| DIN1.FILTER TO DIN7.FILTER..... | 344 |
| DIN1.INV TO DIN7.INV..... | 345 |
| DIN1.MODE TO DIN7.MODE..... | 346 |
| DIN1.PARAM TO DIN7.PARAM..... | 348 |
| DIN1.STATE TO DIN7.STATE..... | 349 |
| DOUT Parameters..... | 351 |
| DOUT.CTRL..... | 351 |
| DOUT.RELAYMODE..... | 352 |
| DOUT.STATES..... | 353 |
| DOUT1.MODE AND DOUT2.MODE..... | 354 |
| DOUT1.PARAM AND DOUT2.PARAM..... | 355 |
| DOUT1.STATE AND DOUT2.STATE..... | 356 |
| DOUT1.STATEU AND DOUT2.STATEU..... | 357 |
| DRV Parameters and Commands..... | 359 |
| DRV.ACC..... | 359 |

| | |
|-----------------------|-----|
| DRV.ACTIVE..... | 361 |
| DRV.BLINKDISPLAY..... | 362 |
| DRV.CLRFAULTHIST..... | 363 |
| DRV.CLRFAULTS..... | 364 |
| DRV.CMDDELAY..... | 365 |
| DRV.CMDSOURCE..... | 366 |
| DRV.CRASHDUMP..... | 367 |
| DRV.DBILIMIT..... | 368 |
| DRV.DEC..... | 369 |
| DRV.DIR..... | 370 |
| DRV.DIS..... | 372 |
| DRV.DISMODE..... | 373 |
| DRV.DISSOURCES..... | 374 |
| DRV.DISTO..... | 375 |
| DRV.EMUEDIR..... | 376 |
| DRV.EMUEMODE..... | 377 |
| DRV.EMUEMTURN..... | 378 |
| DRV.EMUERES..... | 379 |
| DRV.EMUEZOFFSET..... | 380 |
| DRV.EN..... | 381 |
| DRV.ENDEFAULT..... | 382 |
| DRV.FAULTHIST..... | 383 |
| DRV.FAULTS..... | 384 |
| DRV.HANDWHEEL..... | 385 |
| DRV.HELP..... | 386 |
| DRV.HELPALL..... | 387 |
| DRV.HWENMODE..... | 388 |
| DRV.ICONT..... | 389 |
| DRV.INFO..... | 390 |
| DRV.IPEAK..... | 391 |
| DRV.IZERO..... | 392 |
| DRV.LIST..... | 393 |
| DRV.LOGICVOLTS..... | 394 |
| DRV.MEMADDR..... | 395 |
| DRV.MEMDATA..... | 396 |
| DRV.MOTIONSTAT..... | 397 |
| DRV.NAME..... | 398 |
| DRV.NVLIST..... | 399 |
| DRV.NVLOAD..... | 400 |
| DRV.NVSAVE..... | 401 |
| DRV.ONTIME..... | 402 |
| DRV.OPMODE..... | 403 |
| DRV.READFORMAT..... | 404 |
| DRV.RSTVAR..... | 405 |
| DRV.RUNTIME..... | 406 |
| DRV.SETUPREQBITS..... | 407 |

| | |
|----------------------------------|------------|
| DRV.SETUPREQLIST..... | 408 |
| DRV.STOP..... | 409 |
| DRV.TEMPERATURES..... | 410 |
| DRV.TYPE..... | 411 |
| DRV.VER..... | 412 |
| DRV.VERIMAGE..... | 413 |
| DRV.WARNINGS..... | 414 |
| DRV.ZERO..... | 415 |
| FB1 Parameters..... | 416 |
| FB1.BISSBITS..... | 416 |
| FB1.ENCRES..... | 417 |
| FB1.HALLSTATE..... | 418 |
| FB1.HALLSTATEU..... | 419 |
| FB1.HALLSTATEV..... | 420 |
| FB1.HALLSTATEW..... | 421 |
| FB1.IDENTIFIED..... | 422 |
| FB1.INITSIGNED..... | 423 |
| FB1.MECHPOS..... | 424 |
| FB1.MEMVER..... | 425 |
| FB1.OFFSET..... | 426 |
| FB1.ORIGIN..... | 427 |
| FB1.PFIND..... | 429 |
| FB1.PFINDCMDU..... | 430 |
| FB1.POLES..... | 431 |
| FB1.PSCALE..... | 432 |
| FB1.RESKTR..... | 433 |
| FB1.RESREFPHASE..... | 434 |
| FB1.SELECT..... | 435 |
| FB1.TRACKINGCAL..... | 437 |
| FB2 Parameters..... | 438 |
| FB2.ENCRES..... | 438 |
| FB2.MODE..... | 439 |
| FB2.SOURCE..... | 440 |
| FBUS Parameters..... | 441 |
| FBUS.PARAM1 TO FBUS.PARAM20..... | 441 |
| FBUS.PLLSTATE..... | 443 |
| FBUS.PLLTHRESH..... | 444 |
| FBUS.SAMPLEPERIOD..... | 445 |
| FBUS.SYNCACT..... | 446 |
| FBUS.SYNCDIST..... | 447 |
| FBUS.SYNCWND..... | 448 |
| FBUS.TYPE..... | 449 |
| GEAR Parameters..... | 450 |
| GEAR.ACCTMAX..... | 450 |
| GEAR.DECMAX..... | 451 |
| GEAR.IN..... | 452 |

| | |
|-----------------------------|------------|
| GEAR.MODE..... | 453 |
| GEAR.MOVE..... | 455 |
| GEAR.OUT..... | 456 |
| GEAR.VMAX..... | 457 |
| GUI Parameters..... | 458 |
| GUI.DISPLAY..... | 458 |
| GUI.PARAM01..... | 459 |
| GUI.PARAM02..... | 459 |
| GUI.PARAM03..... | 459 |
| GUI.PARAM04..... | 460 |
| GUI.PARAM05..... | 460 |
| GUI.PARAM06..... | 460 |
| GUI.PARAM07..... | 461 |
| GUI.PARAM08..... | 461 |
| GUI.PARAM09..... | 461 |
| GUI.PARAM10..... | 462 |
| HOME Parameters..... | 463 |
| HOME.ACC..... | 463 |
| HOME.AUTOMOVE..... | 465 |
| HOME.DEC..... | 466 |
| HOME.DIR..... | 467 |
| HOME.DIST..... | 468 |
| HOME.FEEDRATE..... | 469 |
| HOME.IPEAK..... | 470 |
| HOME.MODE..... | 471 |
| HOME.MOVE..... | 472 |
| HOME.P..... | 473 |
| HOME.PERRTHRESH..... | 474 |
| HOME.REQUIRE..... | 475 |
| HOME.SET..... | 476 |
| HOME.V..... | 477 |
| HWLS Parameters..... | 478 |
| HWLS.NEGSTATE..... | 478 |
| HWLS.POSSTATE..... | 479 |
| IL Parameters..... | 481 |
| IL.BUSFF..... | 481 |
| IL.CMD..... | 482 |
| IL.CMDU..... | 483 |
| IL.DIFOLD..... | 484 |
| IL.FB..... | 485 |
| IL.FF..... | 486 |
| IL.FOLDFTHRESH..... | 487 |
| IL.FOLDFTHRESHU..... | 488 |
| IL.FOLDWTHRESH..... | 489 |
| IL.FRCTION..... | 490 |
| IL.IFOLD..... | 491 |

| | |
|----------------------------------|------------|
| IL.IUFB..... | 492 |
| IL.IVFB..... | 493 |
| IL.KACFF..... | 494 |
| IL.KBUSFF..... | 495 |
| IL.KP..... | 496 |
| IL.KPDRATIO..... | 497 |
| IL.KPLOOKUPINDEX..... | 498 |
| IL.KPLOOKUPVALUE..... | 499 |
| IL.KPLOOKUPVALUES..... | 500 |
| IL.KVFF..... | 501 |
| IL.LIMITN..... | 502 |
| IL.LIMITP..... | 503 |
| IL.MFOLDD..... | 504 |
| IL.MFOLDR..... | 505 |
| IL.MFOLDT..... | 506 |
| IL.MIFOLD..... | 507 |
| IL.OFFSET..... | 508 |
| IL.VCMD..... | 509 |
| IL.VUFB..... | 510 |
| IL.VVFB..... | 511 |
| 23 LOAD.INERTIA..... | 512 |
| 24 MODBUS Parameters..... | 513 |
| MODBUS.PIN AND MODBUS.POUT..... | 514 |
| MODBUS.PSCALE..... | 515 |
| MODBUS.UNITLABEL..... | 516 |
| MOTOR Parameters..... | 517 |
| MOTOR.AUTASET..... | 517 |
| MOTOR.BRAKE..... | 518 |
| MOTOR.BRAKERLS..... | 519 |
| MOTOR.BRAKESTATE..... | 520 |
| MOTOR.CTF0..... | 521 |
| MOTOR.ICONT..... | 522 |
| MOTOR.IDDATAVALID..... | 523 |
| MOTOR.INERTIA..... | 524 |
| MOTOR.IPEAK..... | 525 |
| MOTOR.KE..... | 526 |
| MOTOR.KT..... | 527 |
| MOTOR.LQLL..... | 528 |
| MOTOR.NAME..... | 529 |
| MOTOR.PHASE..... | 530 |
| MOTOR.PITCH..... | 531 |
| MOTOR.POLES..... | 532 |
| MOTOR.R..... | 533 |
| MOTOR.RTYPE..... | 534 |
| MOTOR.TBRAKEAPP..... | 535 |
| MOTOR.TBRAKERLS..... | 536 |

| | |
|--|------------|
| MOTOR.TEMP..... | 537 |
| MOTOR.TEMPFAULT..... | 538 |
| MOTOR.TEMPWARN..... | 539 |
| MOTOR.TYPE..... | 540 |
| MOTOR.VMAX..... | 541 |
| MOTOR.VOLTMAX..... | 542 |
| MOTOR.VOLTMIN..... | 543 |
| MOTOR.VOLTRATED..... | 544 |
| MOTOR.VRATED..... | 545 |
| MT Parameters and Commands..... | 547 |
| MT.ACC..... | 547 |
| MT.CLEAR..... | 549 |
| MT.CNTL..... | 550 |
| MT.CONTINUE..... | 553 |
| MT.DEC..... | 554 |
| MT.EMERGMT..... | 556 |
| MT.HOMEREQUIRE..... | 557 |
| MT.LIST..... | 558 |
| MT.LOAD..... | 559 |
| MT.MOVE..... | 560 |
| MT.MTNEXT..... | 561 |
| MT.NUM..... | 562 |
| MT.P..... | 563 |
| MT.PARAMS..... | 564 |
| MT.SET..... | 565 |
| MT.TNEXT..... | 566 |
| MT.TNUM..... | 567 |
| MT.TNVSAVE..... | 568 |
| MT.TPOSWND..... | 569 |
| MT.TVELWND..... | 570 |
| MT.V..... | 571 |
| MT.VCMD..... | 572 |
| PL Parameters..... | 573 |
| PL.CMD..... | 573 |
| PL.ERR..... | 574 |
| PL.ERRFTHRESH..... | 575 |
| PL.ERRMODE..... | 577 |
| PL.ERRWTHRESH..... | 578 |
| PL.FB..... | 579 |
| PL.FBSOURCE..... | 580 |
| PL.INTINMAX..... | 581 |
| PL.INTOUTMAX..... | 582 |
| PL.KI..... | 583 |
| PL.KP..... | 584 |
| PL.MODP1..... | 585 |
| PL.MODP2..... | 586 |

| | |
|---|------------|
| PL.MODPDIR..... | 587 |
| PL.MODPEN..... | 588 |
| PLS Parameters and Commands..... | 589 |
| PLS.EN..... | 589 |
| PLS.MODE..... | 590 |
| PLS.P1 TO PLS.P8..... | 591 |
| PLS.RESET..... | 592 |
| PLS.STATE..... | 593 |
| PLS.T1 TO PLS.T8..... | 594 |
| PLS.UNITS..... | 595 |
| PLS.WIDTH1 TO PLS.WIDTH8..... | 597 |
| REC Parameters and Commands..... | 599 |
| REC.ACTIVE..... | 599 |
| REC.CH1 to REC.CH6..... | 600 |
| REC.DONE..... | 601 |
| REC.GAP..... | 602 |
| REC.NUMPOINTS..... | 603 |
| REC.OFF..... | 604 |
| REC.RECPRMLIST..... | 605 |
| REC.RETRIEVE..... | 606 |
| REC.RETRIEVEDATA..... | 607 |
| REC.RETRIEVEFRMT..... | 608 |
| REC.RETRIEVEHDR..... | 609 |
| REC.RETRIEVESIZE..... | 610 |
| REC.STOPTYPE..... | 611 |
| REC.TRIG..... | 612 |
| REC.TRIGPARAM..... | 613 |
| REC.TRIGPOS..... | 614 |
| REC.TRIGPRMLIST..... | 615 |
| REC.TRIGSLOPE..... | 616 |
| REC.TRIGTYPE..... | 617 |
| REC.TRIGVAL..... | 618 |
| REGEN Parameters..... | 619 |
| REGEN.POWER..... | 619 |
| REGEN.REXT..... | 620 |
| REGEN.TEXT..... | 621 |
| REGEN.TYPE..... | 622 |
| REGEN.WATTEXT..... | 623 |
| SM Parameters..... | 625 |
| SM.I1..... | 625 |
| SM.I2..... | 626 |
| SM.MODE..... | 627 |
| SM.MOVE..... | 630 |
| SM.T1..... | 631 |
| SM.T2..... | 632 |
| SM.V1..... | 633 |

| | |
|--------------------------------|------------|
| SM.V2 | 634 |
| STO Parameters | 635 |
| STO.STATE | 635 |
| SWLS Parameters | 636 |
| SWLS.EN | 636 |
| SWLS.LIMIT0 | 637 |
| SWLS.LIMIT1 | 638 |
| SWLS.STATE | 639 |
| UNIT Parameters | 641 |
| UNIT.ACCLINEAR | 641 |
| UNIT.ACCROTARY | 642 |
| UNIT.LABEL | 643 |
| UNIT.PIN | 644 |
| UNIT.PLINEAR | 645 |
| UNIT.POUT | 646 |
| UNIT.PROTARY | 647 |
| UNIT.VLINEAR | 648 |
| UNIT.VROTARY | 649 |
| VBUS Parameters | 650 |
| VBUS.OVFTHRESH | 650 |
| VBUS.OVWTHRESH | 651 |
| VBUS.RMSLIMIT | 652 |
| VBUS.UVFTHRESH | 653 |
| VBUS.UVMODE | 654 |
| VBUS.UVWTHRESH | 655 |
| VBUS.VALUE | 656 |
| VL Parameters | 657 |
| VL.ARPF1 TO VL.ARPF4 | 657 |
| VL.ARPQ1 TO VL.ARPQ4 | 659 |
| VL.ARTYPE1 TO VL.ARTYPE4 | 660 |
| VL.ARZF1 TO VL.ARZF4 | 661 |
| VL.ARZQ1 TO VL.ARZQ4 | 662 |
| VL.BUSFF | 663 |
| VL.CMD | 664 |
| VL.CMDU | 665 |
| VL.ERR | 666 |
| VL.FB | 667 |
| VL.FBFILTER | 668 |
| VL.FBSOURCE | 669 |
| VL.FBUNFILTERED | 670 |
| VL.FF | 671 |
| VL.GENMODE | 672 |
| VL.KBUSFF | 673 |
| VL.KI | 674 |
| VL.KO | 675 |
| VL.KP | 676 |

| | |
|---------------------------|------------|
| VL.KVFF..... | 677 |
| VL.LIMITN..... | 678 |
| VL.LIMITP..... | 679 |
| VL.LMJR..... | 679 |
| VL.MODEL..... | 681 |
| VL.OBSBW..... | 682 |
| VL.OBSMODE..... | 683 |
| VL.THRESH..... | 684 |
| WS Parameters..... | 685 |
| WS.ARM..... | 685 |
| WS.DISARM..... | 686 |
| WS.DISTMAX..... | 687 |
| WS.DISTMIN..... | 688 |
| WS.IMAX..... | 689 |
| WS.MODE..... | 690 |
| WS.NUMLOOPS..... | 691 |
| WS.STATE..... | 692 |
| WS.T..... | 693 |
| WS.TDELAY1..... | 694 |
| WS.TDELAY2..... | 695 |
| WS.TDELAY3..... | 696 |
| WS.VTHRESH..... | 697 |
| 25 Index..... | 700 |

1 About this User Guide

This guide describes the operation and use of the AKD drive. Each section details a specific topic related to the use of the product in basic terms which will help you get the most from the product. Each section include examples to help guide you in setting up and use the various features available in the drive.

This guide is for users who have installed and tested the drive according to the *AKD Installation Manual*. The *AKD Installation Manual* is included on the product CD and contains critical safety information.

1.1 About this User Guide

This guide describes the operation and use of the AKD drive. Each section details a specific topic related to the use of the product in basic terms which will help you get the most from the product. Each section include examples to help guide you in setting up and use the various features available in the drive.

This guide is for users who have installed and tested the drive according to the *AKD Installation Manual*. The *AKD Installation Manual* is included on the product CD and contains critical safety information.

1.2 Abbreviations

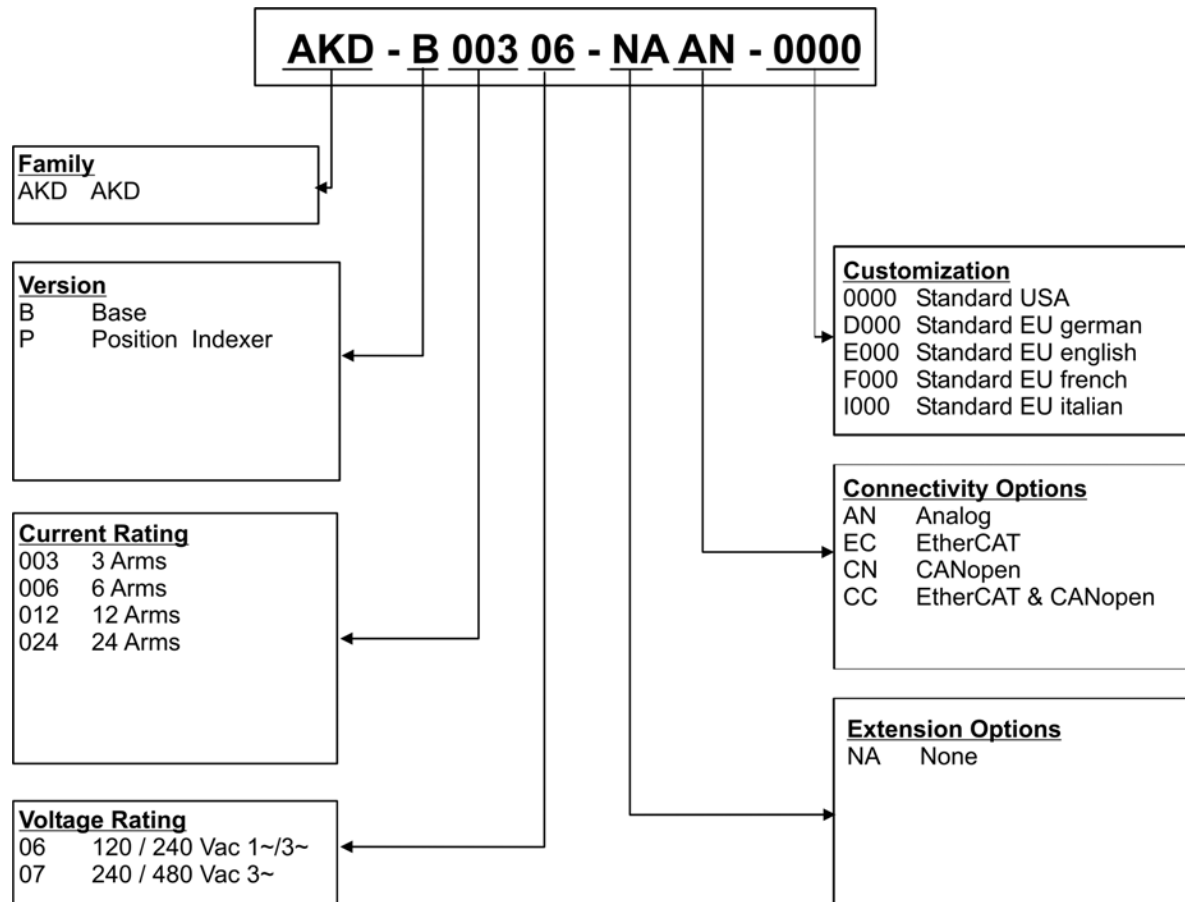
| Abbreviation | Meaning |
|---------------------------------|---|
| AGND | Analog ground |
| CE | Communauté Européenne |
| COM | Serial interface for a personal computer |
| DCOMx | Communication line for digital inputs (with x=7 or 8) |
| Disk | Magnetic storage (diskette, hard disk) |
| EEPROM | Electrically erasable programmable memory |
| EMC | Electromagnetic compatibility |
| F-SMA | Fiber optic cable connector according to IEC 60874-2 |
| LED | Light-emitting diode |
| LSB | Low significant byte (or bit) |
| MSB | Main significant byte (or bit) |
| NI | Zero pulse |
| PC | Personal computer |
| PE | Protective earth |
| PLC | Programmable logic control |
| PLL | Phase locked loop |
| PLS | Programmable limit switch |
| PWM | Pulse-width modulation |
| RAM | Random access memory (volatile memory) |
| $R_{\text{Brake}}/R_{\text{B}}$ | regen resistor (also called a regen resistor) |
| RBext | External regen resistor |
| RBint | Internal regen resistor |
| RCD | Residual current device |
| RES | Resolver |
| ROD | Incremental encoder (A quad B) |
| S1 | Continuous operation |
| STO | Safe torque off |
| Vac | Volts, alternating current |

| Abbreviation | Meaning |
|--------------|-----------------------|
| Vdc | Volts, direct current |

2 AKD Models

AKD drive models are available in a variety of combinations of features. The part number identifies the features included in your model.

The figure below shows part number identification for drive features:



The customization code includes language version of printed material for European countries:

- D000 for German
- E000 for English
- F000 for French
- I000 for Italian

2.1 CC Drive Models

CC drive models allow you to select between EtherCAT, CANopen, or analog operation. This drive model is identified with a new model number of the form AKD-Pxxxxx-NACC-0000 (the CC is the unique identifier).

The CC drive model is fitted with both the EtherCAT (X5 and X6) and CANopen (X12 and X13) fieldbus connectors and a new software parameter (DRV.TYPE) allows you to select which features the drive supports; you cannot use EtherCAT and CANopen at the same time.

This page intentionally left blank.

3 Basic Drive Setup

| | |
|----------------------------|----|
| 3.1 Basic Drive Setup..... | 34 |
| 3.2 Display Codes..... | 34 |
| 3.3 AKD Setup Wizard..... | 35 |

3.1 Basic Drive Setup

The [AKD Quick Start Guide](#) provides details for basic drive setup. Basic drive setup consists of the following general steps:

Hardware Installation:

1. Install the drive in on your conductive panel and connect the Protective Earth ground.
2. Connect the logic power you will need to operate all of the control logic to X1.
3. Connect the motor power to X2.
4. Connect the feedback to X10.
5. Connect the inputs and outputs you will be using on X7 and X8.
6. Bring AC power to the unit and connect AC power to X3 or X4.
7. Connect drive communications to X11.
8. Confirm that you can communicate with the drive and that your PC is linked to the AKD.

Software Installation and Drive Communication Setup:

1. Install and start the interface software (WorkBench).
2. Set the drive IP address using the S1 and S2 switches.
3. Configure the drive using the **Setup Wizard**.

WorkBench System Requirements

Required Components: Microsoft .NET Framework 2.0

Supported Operating Systems:

- Windows XP
- Windows Vista
- Windows 7

3.2 Display Codes

During drive operation, the drive display shows the following codes, depending on the drive status.

| Display Code | Status |
|----------------------------|---|
| o0 | Normal operation, current mode, no faults |
| o1 | Normal operation, velocity mode, no faults |
| o2 | Normal operation, position mode, no faults |
| F [3 digit code, flashing] | Fault (see Fault and Warning Messages) |
| n [3 digit code, flashing] | Warning (see Fault and Warning Messages) |
| I,P [IP address] | Displaying drive IP address |
| [.] | Drive enabled |
| [.] (flashing) | Drive in an internal dynamic brake mode (DRV.ACTIVE = 3). |
| d2 | Firmware download: corrupted operational FPGA; resident FPGA is functional. |
| d3 | Firmware download: HW download (HW switch was pressed - Rev 3 and higher). |
| d4 | Firmware download: Corrupted operational FW. |
| d5 | Firmware download: SW download (download command was issued from the operational FW). |
| dL | Loading image process is running. |
| dF (flashing) | Failure during firmware download. |
| Sb | Special mode: Burn-in |

3.3 AKD Setup Wizard

The Setup Wizard contains step by step instructions for configuring a drive for the first time and generating a simple test motion. You can access the Setup Wizard from the AKD Overview screen or by right clicking on the drive name.

The Setup Wizard is useful when you first set up your system. The wizard confirms your connection with the drive and then leads you through a series of steps to quickly get your drive up and running. With plug and play feedback devices, several steps are skipped (feedback, brake) because the drive automatically configures these settings. For all systems, you can select the units you want to use, configure your operation mode, tune the system, and perform some simple jog moves within the wizard. After you are comfortable with the basic system setup, you can save your settings to the drive and exit the wizard.

This page intentionally left blank.

4 Connecting the Drive

| | |
|--|----|
| 4.1 Connected and Disconnected States..... | 38 |
| 4.2 Disconnected..... | 38 |
| 4.3 Device Not Shown..... | 38 |
| 4.4 Find and Enter IP Address..... | 39 |
| 4.5 Check Communications..... | 40 |
| 4.6 Connect To Another Drive..... | 40 |

4.1 Connected and Disconnected States

WorkBench always starts disconnected from any drives. The **Disconnected** view opens when you start WorkBench and offers two choices:

- **Connect:** Opens the **Connect to a Drive** view.
- **Delete:** Opens a list of available drives and allows you to delete a drive from WorkBench.

While WorkBench is trying to establish [communications](#) with the drive, WorkBench is in the connecting state. Normally, WorkBench will be in the connecting state for a few moments before the connection is established. If WorkBench cannot establish communications correctly, then a five second timeout occurs and WorkBench returns to the disconnected state.

4.2 Disconnected

When WorkBench is [disconnected](#) from a drive, no communication exists between your PC and the drive.

The drive becomes disconnected because of one of the following conditions:

- When WorkBench starts it remembers which drives you were using previously but it does not initially connect to these drives.
- If WorkBench detects that it can no longer communicate with the drive, it will automatically go to this disconnected state. Common causes include a network cable being disconnected or the drive being turned off.
- You pressed the disconnect command.

To restore communication:

1. Clicking Connect will start communication with the drive. If WorkBench cannot find the drive, it will immediately return to the disconnected state.
2. Pressing select will show a window where you can select a different drive you would like to use.
3. Pressing delete will remove this drive from the navigation tree on the left hand side of the main window.

4.3 Device Not Shown

If your specific drive is not shown in the list, then WorkBench has not been able to find the drive.

Common reasons why your drive is not shown in the list include the following:

- The drive is not powered on.
- One of the network cables between your PC and the drive are not connected correctly. You can check if the cable is connected to the drive by checking that the link LED on the Ethernet connector is on continuously. If your PC has a link LED, then you should check that this LED is also continuously lit (usually, this LED is next the RJ45 socket on your PC).
- A router on the network between your PC and the drive is blocking the drive discovery messages. Make sure that port 5002 is not blocked by any routers or firewalls. You can enter the IP address of your drive directly into WorkBench if a router or firewall is blocking port 5002. Often, firewalls are the cause of a blocked connection.
- Your PC and the drive are on different subnets. Networks, especially those with many devices on them, are split up into multiple subnets. The discovery protocol used to find drives will only work if your PC and the drive are both on the same subnet. You can enter the IP address of the drive directly into WorkBench if this is the case.
- The network mask defines more than 512 possible addresses. In this case, WorkBench will not ping all of these addresses, so you must unblock discover ports or specify directly the IP address of your drive.

When an adapter has such a network mask, its background will be shown as yellow to warn that this network will not be discovered with all discover protocols.

4.4 Find and Enter IP Address

You can view the drive IP address on the drive display by pressing the button shown below. The display shows the digits of the IP address in sequence, with dots separating the numbers. You should see four numbers separated by three dots, for example, 192.168.1.5.

You can enter the IP address if you press **More** and check the **Specify Address** box.



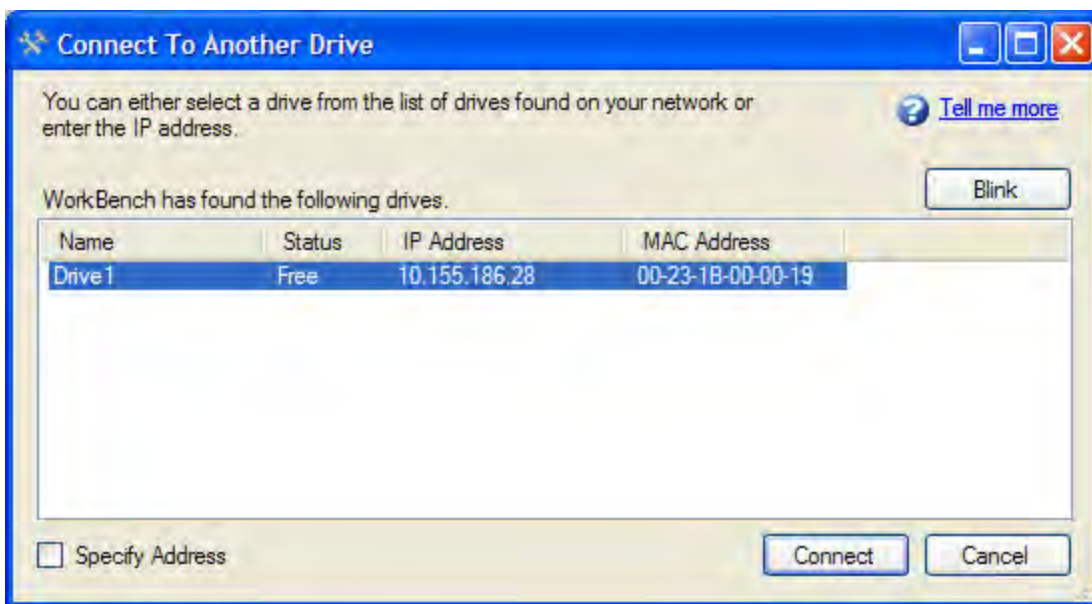
4.5 Check Communications

If you have more than one drive connected to your network, then you can confirm that the new drive is connected to the correct network as follows:

1. A two-digit, seven-segment LED display is located on the front of the drive near the top. If you can see the display, then press the **Blink Display** button on the drive and drive will flash the seven-segment display on and off.
2. If it is difficult to see the display, then you can check the MAC address in the WorkBench list against the MAC address on the label of the drive. The drive is connected if the numbers displayed in WorkBench match the numbers printed on the label on the side of the drive.

4.6 Connect To Another Drive

This window allows you to change the drive that you are using.



| Button or Dialog Box | Description |
|------------------------|--|
| Name | Displays the drive name. By default the name is "No_Name". You can change the name by connecting to the drive and navigating to the top item in the navigation tree. |
| Status | Only one user at a time can connect to a drive. If someone else is connected to the drive, it is Busy . If Free is displayed, then you can connect. |
| Blink | Clicking Blink forces the display on the selected drive to alternate between the whole display being on and the whole display being off for 20 seconds. |
| MAC Address | Displays the MAC address of the drive. The MAC address is unique and is also printed on the label on the side of the drive. |
| IP Address | Displays the IP address of the drive. You can enter a raw IP address (1.2.3.4) or a DNS name. You can also specify a port number different from the default (port 23) by appending the IP address (for example, 1.2.3.4:1000). |
| Specify Address | If your drive does not appear in the list, you can enter its IP address (for example, 1.2.3.4) or a DNS name. You can also specify a port number different from the default (port 23) by appending the IP address (for example, 1.2.3.4:1000). |

5 Communicating with the Drive

| | |
|--------------------------------|----|
| 5.1 Check Communications | 43 |
|--------------------------------|----|

5.0.1 Overview

In order to use the drive, you must be able to communicate with the drive using WorkBench and an Ethernet connection. With some basic network knowledge, you can quickly establish communication with your drive. The drive uses TCP/IP (a worldwide standard for high-speed communication); both the AKD and your PC need to understand each other through this standard in order to communicate. This section explains how to establish a TCP/IP link between your PC and a drive.

5.0.2 Identifying the Drive IP Address

The first step in establishing communication with the drive is to identify the drive IP address. WorkBench and the drive find each other using this IP address, which tells your PC where to look for the drive in order to make the communication connection. You can establish communication through the IP address with two types of connections:

- **Automatic:** Allow the drive and PC to link automatically.
- **Direct:** Connect to a drive directly based on a known IP address.

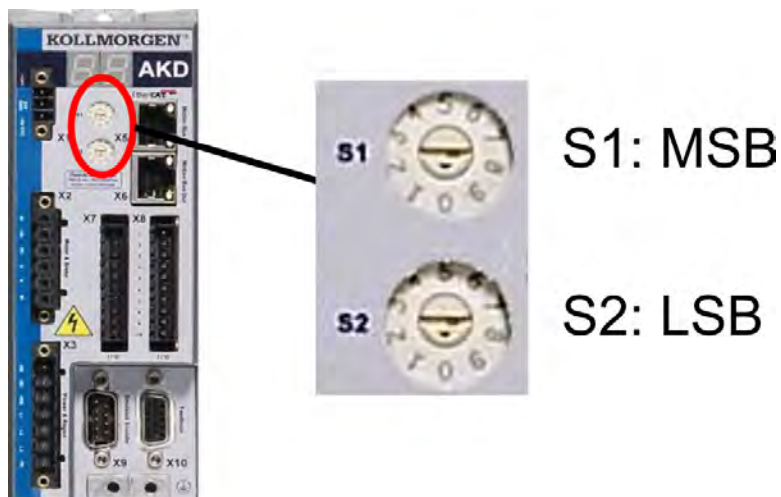
5.0.2.1 Automatic (Dynamic) IP Addressing

Automatic (also called “dynamic”) addressing is performed using the Dynamic Host Configuration Protocol (DHCP). This protocol makes it easy for a device to attach to a network. The drive is set in automatic IP mode by setting the two rotary switches to zero (S1 and S2, located on the front of the drive). Your PC is set in automatic mode by configuring the TCP/IP screen to “Obtain an IP address automatically”

When first communicating with the drive, conflicts might exist with other programs or devices connected to your computer that are competing for IP addresses. If you have a problem recognizing a drive, then try turning off other devices (especially a wireless device or remote network connection). If you still have problems connecting with the drive, check in the troubleshooting area of this manual.

5.0.2.2 Static IP Addressing

Another option in connecting to the drive is via a static IP connection. In this case you are assigning a specific IP address to the drive and you are modifying your pc network configuration to be able to recognize the static address. The drive IP address can be set using the two rotary switches on the front of the drive.



The address will then be set as 192.168.0.S1S2, with S1 representing the 10's digit and S2 the 1's digit. As you turn the switches, the drive displays the S1 and S2 values.

Example:

S1 is set to 3, S2 is set to 5, the address now is set to: 192.168.0.35.

In order for the drive to connect to the PC, the PC network configuration must find this address. First, identify which network port you are using to communicate with the drive. Once you have identified the port, you can access the properties area of the network connection (on your PC) and set up the proper masking to allow the two devices to communicate. The configuration is set up in the "Use the following IP address:". Set the IP address to 192.168.0.100 and the Subnet mask to 255.255.255.0. This allows the two devices to recognize each other and connect point to point (note that S1 = 0 and S2 = 0 is automatic (dynamic) IP addressing).

5.1 Check Communications

If you have more than one drive connected to your network, then you can confirm that the new drive is connected to the correct network as follows:

1. A two-digit, seven-segment LED display is located on the front of the drive near the top. If you can see the display, then press the **Blink Display** button on the drive and drive will flash the seven-segment display on and off.
2. If it is difficult to see the display, then you can check the MAC address in the WorkBench list against the MAC address on the label of the drive. The drive is connected if the numbers displayed in WorkBench match the numbers printed on the label on the side of the drive.

This page intentionally left blank.

5.2 Troubleshooting Connection and Communication Problems

5.2.1 No Drives are Shown

If no drives are shown in the list, then WorkBench has not found any drives.

Common reasons why drives may not appear in the list include the following:

- The drive is not powered on.
- One of the network cables between your PC and the drive is not connected correctly. If the cable is connected correctly, then the LINK LED on the Ethernet connector is lit continuously. If your PC has a LINK LED you should check that this is also continuously lit (normally this LED is next the RJ45 socket on your PC).
- A router on the network between your PC and the drive is blocking the drive discovery messages. You can enter the IP address directly into WorkBench if this is the case (to find the drive IP address, see 5.2.2 Device Not Shown).

5.2.2 Device Not Shown

If your specific drive is not shown in the list, then WorkBench has not been able to find the drive.

Common reasons why your drive is not shown in the list include the following:

- The drive is not powered on.
- One of the network cables between your PC and the drive are not connected correctly. You can check if the cable is connected to the drive by checking that the link LED on the Ethernet connector is on continuously. If your PC has a link LED, then you should check that this LED is also continuously lit (usually, this LED is next the RJ45 socket on your PC).
- A router on the network between your PC and the drive is blocking the drive discovery messages. Make sure that port 5002 is not blocked by any routers or firewalls. You can enter the IP address of your drive directly into WorkBench if a router or firewall is blocking port 5002. Often, firewalls are the cause of a blocked connection.
- Your PC and the drive are on different subnets. Networks, especially those with many devices on them, are split up into multiple subnets. The discovery protocol used to find drives will only work if your PC and the drive are both on the same subnet. You can enter the IP address of the drive directly into WorkBench if this is the case.
- The network mask defines more than 512 possible addresses. In this case, WorkBench will not ping all of these addresses, so you must unblock discover ports or specify directly the IP address of your drive. When an adapter has such a network mask, its background will be shown as yellow to warn that this network will not be discovered with all discover protocols.

5.2.3 Find and Enter IP Address

You can view the drive IP address on the drive display by pressing the button shown below. The display shows the digits of the IP address in sequence, with dots separating the numbers. You should see four numbers separated by three dots, for example, 192.168.1.5.

You can enter the IP address if you press **More** and check the **Specify Address** box.



5.3 TCP/IP Communication Protocols

5.4 IP Address

The IP address of a drive uniquely defines the drive on the network. Ethernet requires that every device on a network segment must have a unique IP address.

5.4.1 Port Number

The AKD drive supports the following protocols using the following port numbers:

TCP: Port 23. Telnet for normal communications with the drive.

Port 502. Modbus TCP.

UDP: Ports 5002 and 5003. These ports are used to discover AKD drives on the local subnet.

5.4.2 MAC Address

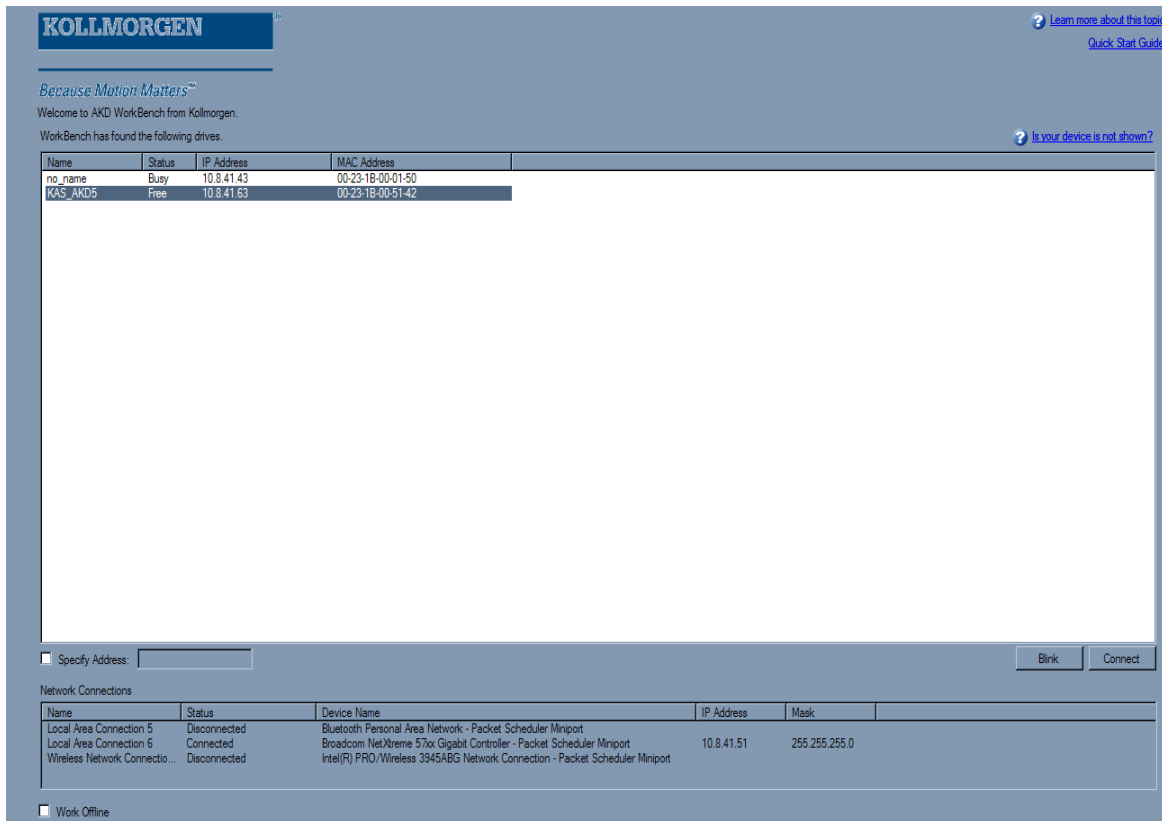
Ethernet also requires that every device must have a globally unique identifier called the MAC address. The MAC address is a 48 bit number normally shown as a series of six hexadecimal numbers (for example, 00:AA:11:BB:22:CC).

Every AKD drive is given a unique MAC address when it is manufactured and this MAC address cannot be changed. The MAC address of every drive is printed on the sticker on the side of the drive.

6 Using WorkBench

6.1 Welcome Screen

This view lets you select which AKD drive you wish to work with. You can work with a physical drive via the Ethernet port of your PC ([online](#)) or with a drive simulation ([offline](#)).



6.1.1 Online

Select **Online** to display a list of the AKD drives that WorkBench has found on your local network. You can select one of these drives from the list and click **Next** to continue. This will connect you to the drive and you will be given the option to use a wizard to setup the drive.

| Button or Dialog Box | Description |
|----------------------|---|
| Name | The name that someone has given the drive. By default, the name is "No_Name". You can change the name by connecting to the drive and navigating to the top item in the navigation tree. |
| Status | Only one user can connect to an AKD at a time. If someone else is connected to the drive, then the status is Busy . If no one is connected to the drive, then the status is Free and you will be able to connect. |
| Blink | Clicking Blink will force the display on the selected drive to repeatedly flash the display LEDs. |
| MAC Address | This is the MAC address of the drive. The MAC address is unique and is also printed on the label on the side of the drive. |
| IP Address | This is the IP address of the drive. |

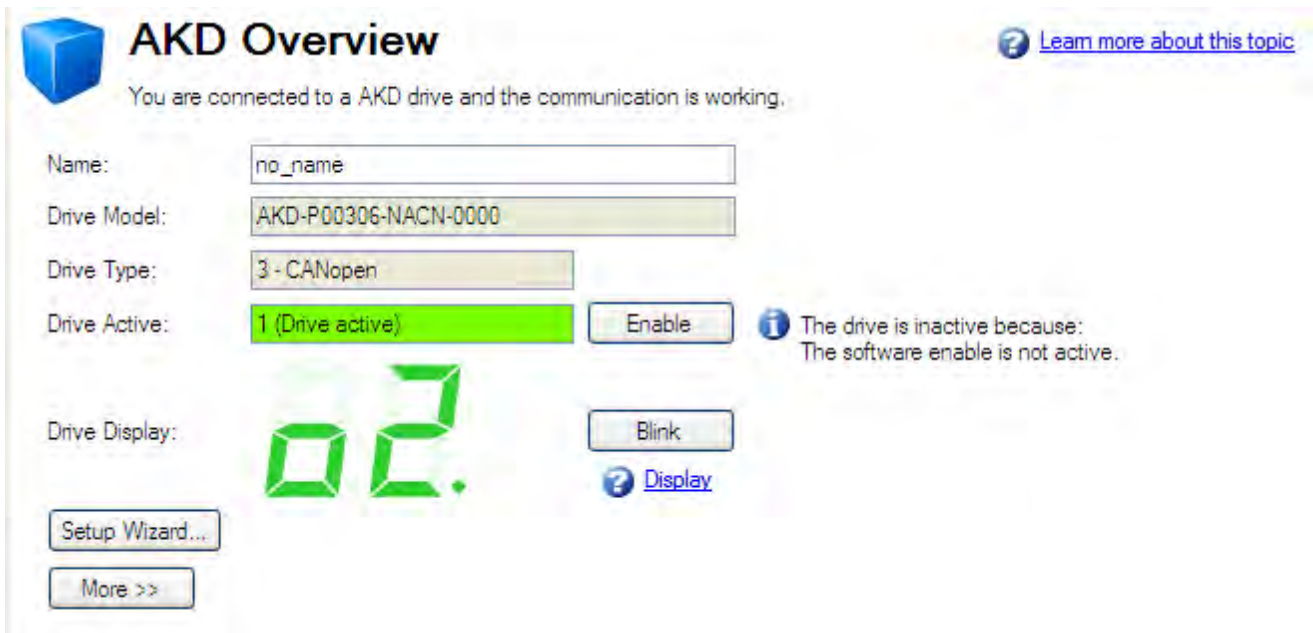
| Button or Dialog Box | Description |
|------------------------|---|
| Specify Address | If your drive does not appear in the list, you can enter its IP address (e.g. 1.2.3.4) or a DNS name. You can optionally specify a different port number than the default port 23) by appending it (for example, 1.2.3.4:1000 would be port 1,000). |

6.1.2 Offline

Select **Offline** to display a list of the different models that WorkBench can simulate. Once you have made your selection, click **Next** and the **Overview** screen for the Offline drive opens.

6.2 AKD Overview

Once your drive is connected, the **AKD Overview** shows a summary of the drive that you are using.



You can view or edit the following information from the Overview window.

| Button or Dialog Box | Description | Parameter |
|----------------------|---|------------|
| Name | Names each drive in use with a unique identifier. | DRV.NAME |
| Drive Model | Displays the model number of this drive. The model number is also on the label on the side of the drive. If you are offline , then you can change the type of drive that you are simulating. | DRV.INFO |
| Drive Type | Selects the operational fieldbus for your drive. | DRV.TYPE |
| Drive Active | The drive is active when it is enabled and also supplies voltage to the motor. | DRV.ACTIVE |
| Enable | Click Enable to turn on the power stage in the drive and apply voltage to the motor. This command may fail for many reasons; see "DRV.EN" (=> p. 381) for further details. | DRV.EN |
| Disable | Click Disable to turn off the power stage and remove the voltage applied to the motor. | DRV.DIS |

| Button or Dialog Box | Description | Parameter |
|---------------------------|---|------------------|
| Drive Display | This graphic replicates the two-digit seven-segment display located on the front face of the drive. The seven-segment display shows a code that indicates the state of the drive and any faults that may be present. WorkBench shows a copy of what the drive display currently shows. A key to the display is here . | |
| Blink | Click Blink to force the display to alternate between the whole display being on and the whole display being off for 20 seconds. You can use this button to confirm that you are communicating with the correct drive hardware. | DRV.BLINKDISPLAY |
| Setup Wizard | The Setup wizard takes you through the essential configuration steps so that you can control the motor movement. | |
| More | Click More to display Serial Number, Firmware Version, Cumulative On Time, and Update Firmware. | |
| Serial Number | This text box displays the unique serial number of the drive you are communicating with. The serial number is also shown on the label on the side of the drive. | DRV.INFO |
| Firmware Version | This text box displays the version of the firmware code running inside the drive. | DRV.VER |
| Download | Click Download to retrieve the latest AKD firmware from Kollmorgen. See 19.1 Downloading Firmware | |
| Cumulative On Time | This text box displays the cumulative time this drive has been powered on. When the drive is powered on, this value continues counting from the value it had when the drive was last turned off. | DRV.RUNTIME |
| Update Firmware | Use this box to select the firmware version you want your drive to run. | |
| Drive IP Address | This text box displays the IP address of the drive that you are communicating with. | |
| MAC Address | This text box displays the MAC address of the drive that you are communicating with. The MAC address is unique and is also printed on the label on the side of the drive. | |

6.3 Online and Offline

WorkBench allows you to work online (working with a real drive) or offline (working without any drive hardware).

6.3.1 Online Drive

An "online drive" means that WorkBench is working with a specific physical drive on your network.

Each online drive can either be connected to WorkBench (WorkBench has an active connection with the drive and data is being passed between WorkBench and the drive) or it can be disconnected (there is no communication between WorkBench and the drive). If communications are lost (for example a network cable is disconnected) with a drive then WorkBench will switch the drive to the disconnected state.

Only one PC can be connected to a drive at a time.

6.3.2 Offline Drive

An offline drive allows you to use WorkBench without having any drive hardware. The parameters of a drive are simulated within WorkBench. An offline drive allows you to create a drive configuration as well as exploring the

different screens within WorkBench. Because this is a simulation there are a number of operations that are not possible (for example commanding motion).

6.3.3 Switching Between Online and Offline

WorkBench does not allow you to change an instance drive between offline or online. If you wish to move a configuration between two devices that you can save the parameters to a file and then import this parameter file into a new drive you have created.

6.4 Watch

This window allows you to view the current value of selected information from the drive. You can toggle the window on/off by clicking the glasses on the tool bar.

By default, the watch window shows position, velocity, and current for the selected drive. The list can be customized as shown below.

| Button or Dialog Box | Description |
|----------------------|--|
| Add | Adds a new parameter to the watch list. |
| Edit | Allows you to modify the selected item. |
| Delete | Removes the selected items from the list. |
| Move Up | Moves the selected items up one place in the list. |
| Move Down | Moves the selected items down one place in the list. |

6.5 Settings

6.5.1 Navigation Tree

The Settings view allows you to configure the drive to fit your specific application. When you click Settings in the navigation pane located to the left of the WorkBench screen, additional views for configuring your drive appear. By default, only settings applicable to your current drive operation mode and command source will appear in this tree. To show all settings available for the AKD (even if these settings are not used with the current operation mode and command source), right click on **Settings** and select **Show all Settings**.

6.5.2 Settings View

From the main Settings view, you can configure the following drive settings:

Settings

[? Learn more about this topic](#)

Select which mode of operation and command source the drive should work in. The AKD drive can work in many different modes, each mode is suited to different types of application.

Command Source:

0 - Service
▼

Operation Mode:

2 - Mode Position
▼

Home

Motion Task

Service Motion

| Button or Dialog Box | Description |
|---|---|
| Command Source | Selects where the command is being provided: 0-Service: You communicate with the drive using the TCP/IP service channel. 1-Fieldbus: The drive is being controlled by commands coming over the field-bus. 2-Gearing: The position is proportional to the secondary feedback. 3-Analog: The analog input provides a current, velocity, or position command. |
| Operation Mode | Selects the control loop being commanded from the source: 0-Torque Mode: Drive controls are based on the current passing through the motor. For a rotary motor, this value is proportional to the torque of the motor; for linear motors, this value is proportional to the force the motor generates. 1-Velocity Mode: Drive controls are based on the velocity of the motor. 2-Position Mode: Drive controls are based on the position of the motor. |
| Motion and Control Loop Graphics | Allows you to select the details related to each specific loop from a graphical interface. |

Related Topics

- 11 Using Command Source and Operating Modes for details on configuring the drive for your application.

This page intentionally left blank.

7 Configuring Drive Power

| | |
|-----------------------|----|
| 7.1 Power..... | 56 |
| 7.2 Regeneration..... | 58 |

7.1 Power

7.1.1 Drive Setup for Power and Bus

The **Power** screen allows you to confirm Power Bus settings and accommodate external regeneration needs if required. Nothing is required for this screen if you have no regeneration requirements. Review the data on the screen to be certain the bus voltage is at the appropriate levels you expect (approximate input line AC voltage * 1.4). The other values are the appropriate limits for over voltage and under voltage for the particular drive. You can select the undervoltage fault mode to trigger either only when the drive is enabled or always.

From the **Power** screen, you can view and configure the drive power settings as follows:

| Button or Dialog Box | Description | Parameter |
|------------------------------------|---|------------------------|
| Measured Bus Voltage | Reads the current DC bus voltage. | VBUS.VALUE |
| Over Voltage Fault Level | Reads the over voltage fault level. | VBUS.OVFTHRESH |
| Under Voltage Fault Level | Reads the under voltage fault level. | VBUS.UVFTHRESH |
| Under Voltage Fault Mode | Sets under voltage mode. | VBUS.UVMODE |
| Regen Resistor Type | Sets the regen resistor type to either -1-External Regen or 0-Internal Regen (if available) . | REGEN.TYPE |
| Regen Power | Reads the regen power (only visible for external regen). | REGEN.POWERREGEN.POWER |
| External Regen Resistance | Sets the external, user-defined regen resistor resistance (only visible for external regen). | REGEN.REXT |
| External Regen Heat Up Time | Sets the external regen resistor thermal protection time constant (only visible for external regen). | REGEN.TEXT |
| External Regen Power | Sets the regen resistor's power fault level for an external regen resistor (only visible for external regen). | REGEN.WATTEXT |

See 7.2 Regeneration for more information about regen resistors and sizing regen resistors.

7.1.2 Direct DC Mains Operation

Direct DC input is available on all standard AKD models. The DC input should be run into the AC input connection. Positive and negative DC lines should use L1 and L2 connections (polarity is not critical). L1 and L2 connections are found on either the X3 connector or the X4 connector depending on the model.

(see "Mains Supply Connection (X3, X4)" (=> p. 271) for more information on this connection).

The nominal level of DC voltage applied must be compatible with the voltage fault levels in the drive. You must also consider voltage variations in the DC power supply above and below the nominal value so that nuisance faults are avoided.


When you determine the maximum nominal DC voltage applied to the drive you should also consider the regeneration circuit, in addition to the over voltage level. Running the drive slightly below the over voltage level is not possible because the drive does not have the capability to dissipate regenerated energy. This practice can also be harmful to the regen circuit. A good practice is not to exceed the nominal DC voltage produced by a standard AC installation. For the AKD-zzzzz06, 340 Vdc is the equivalent DC voltage for an 240 Vac supply and for the AKD-xxxxx07, 680 Vdc is the equivalent DC voltage for a 480 Vac supply.

The voltage fault levels are also shown in the **Power** screen and depend on the voltage level of drive used.

Voltage ranges are as follows:

| Model | Under Voltage Level | Over Voltage Level |
|------------|---------------------|--------------------|
| AKD-zzzz06 | 90 Vdc | 420 Vdc |
| AKD-zzzz07 | 380 Vdc | 840 Vdc |

You can view bus voltage values in the **Power** screen as shown below:



Power

Monitor the Bus Voltage and configure the Regen. Resistor.

Bus Voltage

Measured Bus Voltage: V

Over Voltage Fault Level: V

Under Voltage Fault Level: V

Under Voltage Fault Mode: ▾

Regen. Resistor

Regen. Resistor Type: ▾

Regen. Power: W

7.2 Regeneration

7.2.1 Overview

Regeneration, or "shunting", dissipates energy from the DC bus during deceleration of the motor load. During deceleration, the motor acts as a generator that pumps energy back into the system. If this energy is not dissipated, then the bus energy level can exceed acceptable levels (VBUS.OVFTHRESH). If the system exceeds the maximum bus voltage, then the drive generates an over voltage fault (FF501) and shuts down. A regeneration resistor, or regen resistor, is an external device that dissipates excess energy and allows the drive to function normally during deceleration.

Follow these steps to determine the regen needs for your system and to configure the drive for regen:

1. Calculate motor peak and continuous regenerative energy and use this value to size the regen resistor.
2. Select a compatible regen resistor.
3. Configure regen parameter values in WorkBench.

7.2.2 Regen Resistor Options

In the **Power** screen, you can select from a variety of pre-sized regen resistors using the **Regen Resistor Type** box.

Regen. Resistor

| | | | |
|------------------------------|---------------------|------------------|----------------|
| Regen. Resistor Type: | -1 - External Regen | Select Resistor: | <User Defined> |
| External Regen Resistance: | 0 | | |
| External Regen Heat Up Time: | 100.000 | | |
| External Regen Power: | 1,000 | | |
| Regen. Power: | 0 | | |

| | |
|----------------|-----------|
| BAR-250-33 | DE-106254 |
| BAR-500-33 | DE-106255 |
| BAR-1500-33 | DE-106258 |
| BAS-3000-33 | DE-201407 |
| <User Defined> | |

When you select -1-External Regen, the **Select resistor** box appears and contains the pre-sized resistors for your drive. Select the model regen resistor that you are using and the drive will populate the remainder of the fields. If you are using a nonstandard resistor, then choose **<User defined>** and fill in the appropriate values for your resistor.

NOTE

If you use a nonstandard resistor, contact Kollmorgen technical support to confirm that the nonstandard resistor will work correctly with your system.

7.2.3 Calculating Motor Peak Energy and Regen Resistor Size

In order to determine whether or not your system requires a regen resistor, you must calculate the peak kinetic energy that the motor generates during deceleration and the continuous regeneration energy created by the motor. If this energy exceeds the drive capacity, then you need a regen resistor. In many cases, peak or continuous regenerated energy does not exceed the drive capacity and no regen resistor is needed.

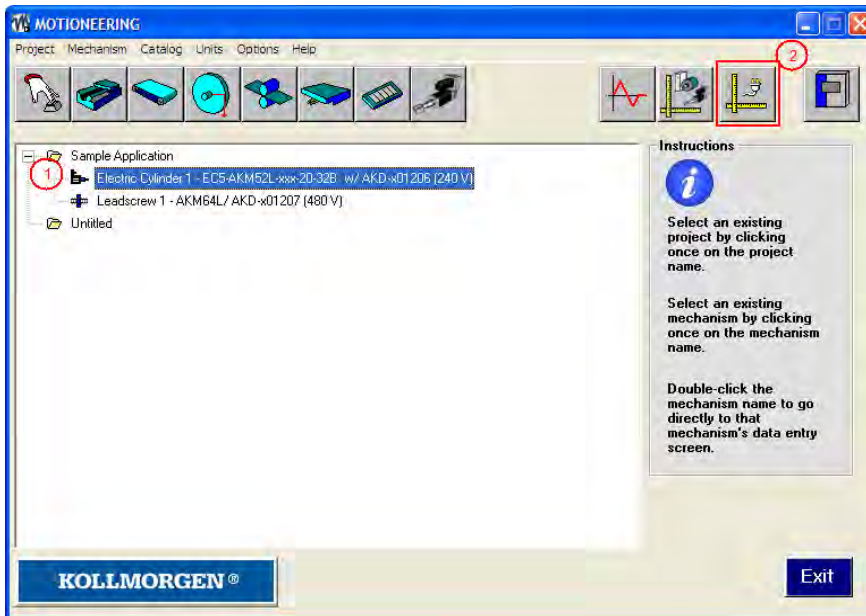
The calculation for peak kinetic energy requires values for several factors that affect the generation of energy in a motion system:

- Load inertia
- Motor inertia
- Motor speed from which deceleration occurs
- Time required to decelerate

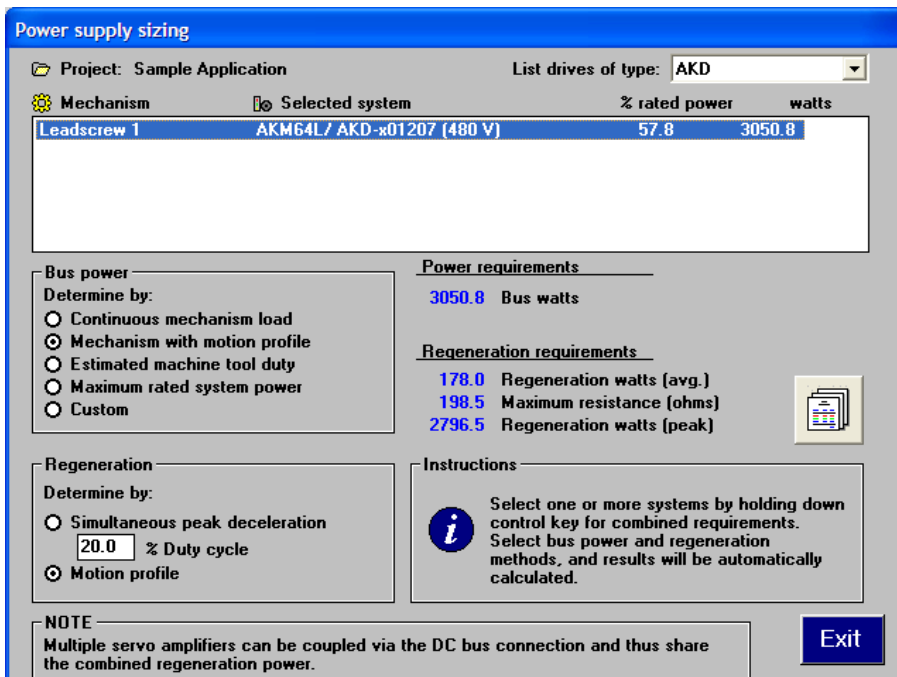
Regen resistor sizing information for your particular application can be calculated using the Motioneering® Application Engine. You can download this program here:

http://www.kollmorgen.com/website/com/eng/support/design_tools/motioneering.php

After you install and set up this program, highlight your application (1) and then click on the Power Supply Sizing icon (2).



The application then displays the regen sizing tool; see the application help for further sizing assistance.



7.2.4 Selecting a Compatible Regen Resistor

After you calculate the appropriate resistor size, compare the results with the capabilities of the drive and, if necessary, select an external regeneration resistor which matches these capabilities from the chart below. The resistors shown below are included in the WorkBench setup. If you do not find a match for your application, please contact the Kollmorgen customer support team for further assistance.

| NA Part Number | EU Part Number | Resistor Type | AKD- x- 00306 | AKD- x- 00606 | AKD- x- 01206 | AKD- x- 02406 | AKD- x- 00307 | AKD- x- 00607 | AKD- x- 01207 | AKD- x- 02407 |
|----------------|----------------|---------------------------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| BAFP-100-33 | DE-201437 | External Resistor, 100 W, 33 ohms | x | x | x | | | | | |
| BAFP-200-33 | DE-201438 | External Resistor, 200 W, 33 ohms | x | x | x | | | | | |
| BAR-250-33 | DE-106254 | External Resistor, 250 W, 33 ohms | x | x | x | | x | x | x | |
| BAR-500-33 | DE-106255 | External Resistor, 500 W, 33 ohms | x | x | x | | x | x | x | |
| BAR-1500-33 | DE-106258 | External Resistor, 1500 W, 33 ohms | x | x | x | | x | x | x | |
| BAS-3000-33 | DE-201407 | External Resistor, 3000 W, 33 ohms | x | x | x | | x | x | x | |
| BAR-600-23 | DE-200613 | External Resistor, 600 W, 23 ohms | | | | | | | | x |
| BAR-1000-23 | DE-200614 | External Resistor, 1000 W, 23 ohms | | | | | | | | x |
| BAS-2000-23 | DE-200615 | External Resistor, 2000 W, 23 ohms | | | | | | | | x |
| BAS-3000-23 | DE-200616 | External Resistor, 3000 W, 23 ohms | | | | | | | | x |
| BAS-4000-23 | DE-200617 | External Resistor, 4000 W, 23 ohms | | | | | | | | x |
| BAR-500-15 | DE-201439 | External Resistor, 500 W, 15 ohms | | | | x | | | | |
| BAR-1000-15 | DE-201440 | External Resistor, 1000 W, 15 ohms | | | | x | | | | |
| BAS-2000-15 | DE-103871 | External Resistor, 2000 W, 15 ohms | | | | x | | | | |
| BAS-3000-15 | DE-103872 | External Resistor, 3000 W, 15 ohms | | | | x | | | | |
| BAS-6000-15 | DE-103873 | External Resistor, 6000 W, 15 ohms | | | | x | | | | |

7.2.5 Configuring Regen Parameter Values

If you use an external regen resistor, then you must also include additional information about the regen resistor in the **PowerView**. These values are automatically inserted in WorkBench. The chart below summarizes these three additional values:

- **External Regen Resistance** (REGEN.REXT, ohms)
- **External Regen Heat Up Time** (REGEN.TEXT, seconds)
- **External Regen Power** (REGEN.WATTEXT, watts)

These parameters allow the regen resistor to function properly and dissipate power appropriately for your system.

| NA Part Number | EU Part Number | Resistor Type (all resistors UL recognized) | Resistance ohms (REGEN.REXT) | Heat Up Time s (REGEN.TEXT) | Power Rating W (REGEN.WATTEXT) |
|----------------|----------------|--|------------------------------------|-----------------------------------|--------------------------------------|
| BAFP-100-33 | DE-201437 | External Resistor, 100 W, 33 ohms | 33 | 16.5 | 100 |
| BAFP-200-33 | DE-201438 | External Resistor, 200 W, 33 ohms | 33 | 27.5 | 200 |
| BAR-250-33 | DE-106254 | External Resistor, 250 W, 33 ohms | 33 | 22.0 | 250 |
| BAR-500-33 | DE-106255 | External Resistor, 500 W, 33 ohms | 33 | 33.0 | 500 |
| BAR-1500-33 | DE-106258 | External Resistor, 1500 W, 33 ohms | 33 | 25.7 | 1,500 |
| BAS-3000-33 | DE-201407 | External Resistor, 3000 W, 33 ohms | 33 | 77.0 | 3,000 |
| BAR-600-23 | DE-200613 | External Resistor, 600 W, 23 ohms | 23 | 27.5 | 600 |
| BAR-1000-23 | DE-200614 | External Resistor, 1000 W, 23 ohms | 23 | 27.5 | 1,000 |
| BAS-2000-23 | DE-200615 | External Resistor, 2000 W, 23 ohms | 23 | 77.0 | 2,000 |
| BAS-3000-23 | DE-200616 | External Resistor, 3000 W, 23 ohms | 23 | 84.3 | 3,000 |
| BAS-4000-23 | DE-200617 | External Resistor, 4000 W, 23 ohms | 23 | 77.0 | 4,000 |
| BAR-500-15 | DE-201439 | External Resistor, 500 W, 15 ohms | 15 | 33.0 | 500 |
| BAR-1000-15 | DE-201440 | External Resistor, 1000 W, 15 ohms | 15 | 27.5 | 1,000 |
| BAS-2000-15 | DE-103871 | External Resistor, 2000 W, 15 ohms | 15 | 77.0 | 2,000 |
| BAS-3000-15 | DE-103872 | External Resistor, 3000 W, 15 ohms | 15 | 84.3 | 3,000 |
| BAS-6000-15 | DE-103873 | External Resistor, 6000 W, 15 ohms | 15 | 91.7 | 6,000 |

7.2.6 Related Parameters

REGEN Parameters

VBUS.OVWTHRESH

VBUS.VALUE

8 Configuring Motor Settings

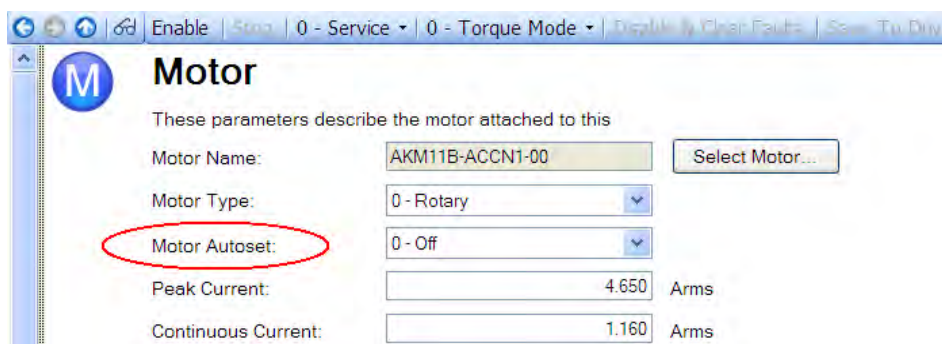
| | |
|---|----|
| 8.1 Motor..... | 64 |
| 8.2 Feedback 1..... | 66 |
| 8.3 Using Feedback Options..... | 67 |
| 8.4 Feedback 2..... | 73 |
| 8.5 Non-Plug and Play Feedback Devices..... | 76 |
| 8.6 Parameters..... | 76 |
| 8.7 Calculations..... | 76 |
| 8.8 Foldback..... | 77 |
| 8.9 Overall Foldback..... | 80 |
| 8.10 Brake..... | 80 |
| 8.11 Using Position Capture..... | 81 |

8.1 Motor

8.1.1 Overview

The **Motor** screen is used to set up or confirm the parameters of the motor that is connected to the drive. In certain cases, based on the feedback type, the motor parameters will automatically be set. The drive will auto-detect feedback devices that are preset with the appropriate feedback and motor parameters when MOTOR.AUTOSSET is set to 1 (default). The values the drive uses for commutation and current and velocity loop gains will be populated automatically.

If your motor does not have a plug and play feedback device, then you must turn off the motor autoselect feature as shown in the screenshot below (MOTOR.AUTOSSET = 0) and select the appropriate motor from the motor parameter database.



All of the appropriate Kollmorgen motors compatible with the AKD drive are contained in the motor database. For motors that are not listed, click **Select Motor** to open a custom motor view in which you can input the appropriate motor parameters.

8.1.2 Motor Setup

For SFD, Endat, and BiSS feedback devices, the drive will automatically be detected and the correct motor parameters will be automatically set in the AKD drive. If your motor is detected automatically, the parameters in the **Motor** view are shaded and not accessible. If you have a non-plug and play standard device (such as an incremental encoder or resolver), you can use this screen to enter the standard AKM, Cartridge motor, DDR, or DDL motor. Other motors can be entered using the **Custom Motor** selection and setting up the parameters (see section on non memory motors). Select the standard motor from the drop down list and select ok. The AKD drive will now show all of the appropriate parameters needed for the motor to operate properly.

8.1.3 Using the Motor View

The **Motor** view displays parameters related to the specific motor attached to the drive as follows:

- **Motor Name:** The motor part number read from the autoselect device, or the name from the motor database. When entering a custom motor name, the motor name should not contain any spaces.
- **Motor Type:** This field allows you to select the proper parameters for a rotary motor or a linear motor (linear motors are a future feature).
- **Motor Autoselect:** This setting allows the drive to automatically set up a plug and play motor (MOTOR.AUTOSSET = 1). With **Motor Autoselect** turned off (MOTOR.AUTOSSET = 0), you can access the motor database to select a catalog or custom motor.

The next set of parameters displayed are specific to the electrical and mechanical characteristics of the motor connected to the AKD drive.

- Peak Current: Motor peak current rating in Amps rms.
- Continuous Current: Motor continuous current rating in Amps rms.
- Inertia: Motor rotor inertia in Kg-cm².
- Torque Constant: Motor torque constant in Nm/Arms.
- Inductance: Motor rated inductance in milliHenries
- Motor Poles: Number of motor poles.
- Maximum Speed: Motor maximum rated speed
- Motor Resistance: Motor winding resistance in Ohms.
- Maximum Voltage: Motor maximum rated voltage in Volts rms.
- Motor Phase: Motor phase offset (used to set motor commutation as required - for most devices this is set to 0).
- Coil Thermal Constant: Motor coil thermal time constant in mHz.

8.1.4 Selecting a Motor

The **Select Motor** button opens a screen in which you can configure a non-plug and play motor or custom motor.

When this screen is opened, WorkBench displays by default the motor matched to the current motor name attached to the drive. WorkBench searches matching motor as follows:

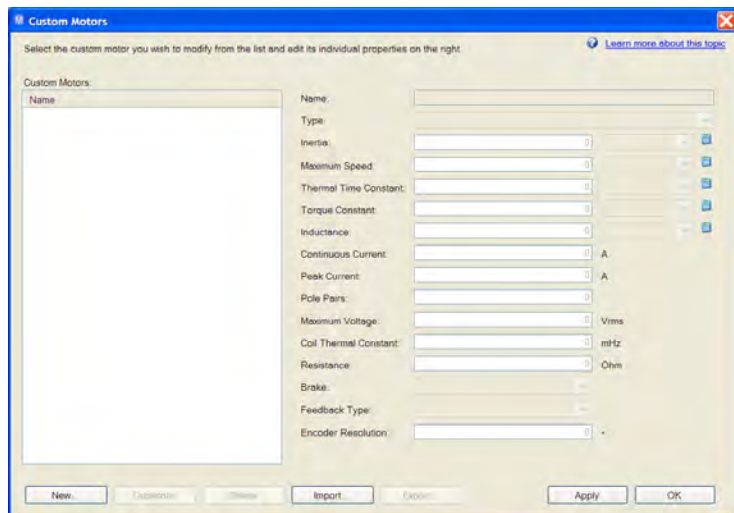
1. WorkBench first checks the motor name with custom motors for a match.
2. If a match is not found, then WorkBench checks the name with the standard motors database for a match.
3. If a match is not found, then an AKM motor is selected.

For non-plug and play motors, a database of catalog motors is available based on the different Kollmorgen motor families. When you select a motor family, a part number is displayed according to the selected motor family. You can change the part number as needed, and the complete motor name will be displayed according to your selection. This complete motor name is sent to drive. The portions of the part number labeled in bold are required values.

The **More/Less** button displays and hides motor temperature settings.

8.1.4.1 Configuring Custom Motors

Click on **Custom Motor** to create and edit custom motors in the following screen:



In this view, you can import or export a motor parameter file, or create a custom motor of your choice. The appropriate parameters must be chosen as listed. Several of the parameters allow you to select an alternate unit of measure. When building a custom motor file, do not use blank spaces in the name you choose. Once you have configured one or more custom motors, if you select a custom motor from the list and click **OK**, then the selected custom motor will be displayed in the **Select Motor** screen.

The actions available in the custom motor screen include:

- **New.** Allows you to start a new custom motor (with default values) or load a catalog motor you may want to modify.
- **Duplicate.** Makes a copy of the highlighted motor in the custom motor list.
- **Delete.** Deletes the highlighted motor in the customer motor list.
- **Import.** Allows you to import a motor file (*.motor) from another location
- **Export.** Allows you to save the highlighted motor file (*.motor) to another location
- **Apply.** Accepts the values you have entered for the specific motor files you are entering.
- **OK.** Returns you to the **Motor Selection** screen.

When entering any of the motor data, be certain the units are correct. The AKD drive uses the motor parameters to set up the various feedback loops and limits associated with the motor selected.

Note that if you select a custom motor from the list and click Ok button then that selected custom motor will be displayed in the Select Motor screen.

8.1.4.2 Validating Motor Parameters

When you click **OK** in the **Select Motor** screen, WorkBench validates the range with the drive. If any error is found, an error screen is displayed. Click **Continue** to set the motor parameters in the drive. Click **Cancel** to close this screen.

If errors occur while setting the motor parameters, an error screen indicates which parameters require additional attention.

Related Parameters

MOTOR Parameters

8.2 Feedback 1

The Feedback view allows you to configure the primary position feedback device fitted to your motor.

When you select your feedback device from the Feedback Selection list, the appropriate feedback configuration choices appear below the dial.

8.2.1 Overview

The AKD offers a variety of feedback solutions, which allows you to optimize your system based on your specific machine needs. The available feedback options include resolver, SFD, sine-cosine encoder (Endat 2.1, BiSS, Hiperface), incremental encoder, as well as line-count, single, and absolute variations. Your motor model number will indicate the type of feedback that you have. With some incremental encoders, parameters are set up in the encoder itself, and the AKD drive recognizes the feedback automatically and sets up the drive accordingly. This automatic recognition is called "plug and play". Currently, SFD and Endat are plug and play encoders. Other feedback types require that you enter parameters manually.

8.3 Using Feedback Options

Use the **Feedback** screen to set up your system to match the proper feedback device. By default, the drive uses the **Auto** setting to detect feedback devices. This setting allows the drive to interrogate the feedback device to see if it is a recognized plug and play device. If the drive recognizes the device, then all the parameters for that device and motor are loaded into the drive. Both the feedback and the motor information are now present in the drive and the system is operable.

If the feedback is a non-plug and play device, then you can choose from the list of supported devices in **Feedback Selection** list and then enter the line count manually. The following sections describe each supported device available in the **Feedback Selection** list and the input information required to configure each device.

8.3.1 Auto

This is the default setting and is used to determine if a plug and play device is available. If a plug and play device is available, the **Auto** mode is replaced by the feedback device detected, along with the appropriate resolution settings.

8.3.2 Incremental Encoder

The incremental encoder is a non-plug and play device. Incremental encoders are available in a variety of line counts. If you select an incremental encoder option, the encoder resolution must be entered into the **Rotary Encoder Resolution** box. The units for this field are in counts per revolution, which is post-quadrature (multiplies the lines per revolution by 4).

8.3.3 Sine Encoder

Sine encoders are offered with different data communication protocols. These include Endat, BiSS, Hiperface, and others. A standard sine-cosine encoder with simple analog communication is not a plug and play device. As with the incremental encoder, the line count is entered in the **Rotary Encoder Resolution** box.

8.3.4 Endat 2.1, Endat 2.2

Endat-based sine encoders are plug and play compatible, and the system will properly recognize these encoders. With the AKD set in **Auto**, this encoder type is detected and the feedback and motor parameters are loaded automatically.

8.3.5 BiSS

BiSS will be plug and play in a future release. Currently, the device is programmed with the motor and feedback information and once selected will set up the feedback and motor parameters in the AKD.

8.3.6 Hiperface

Hiperface is a plug and play device that will be supported in a future release.

8.3.7 Resolver

The resolver feedback option is not plug and play. When selecting the resolver option, three specific parameters are set by default for the standard AKM resolver: phase lag, transformation ratio, and feedback poles. Currently, the AKD does not support non-Kollmorgen standard resolver options.

8.3.8 SFD

Smart Feedback Device (SFD) is Kollmorgen's most popular plug and play device. SFD allows for quick and easy setup from the **Auto** mode, which automatically configures the drive with the motor and feedback parameters.

8.3.9 Using Wake and Shake (WS)

8.3.9.1 Overview

Wake and shake (WS) is used to establish commutation in drives with the following types of feedback:

- Incremental encoders without Halls or commutation channels.
- Sine encoders without Halls or commutation channels.

When controlling a brushless DC (BLDC) motor, you must know the electrical position of the motor shaft. Without absolute position data, it is impossible for the drive to know which sequence of coils to energize to produce motion. Absolute feedback devices, such as resolvers and absolute encoders, can detect position directly. Incremental devices, such as incremental encoders and sine encoders without a commutation channel, must determine electrical position indirectly at start up. The drive uses the WS feature to determine electrical position by sending the motor short bursts of current and measuring the resulting incremental motion. The drive uses this measurement to estimate electrical position accurately enough to control the motor.

8.3.9.2 Configuring WS

You can configure WS after your motor has been connected to the AKD according to the *AKD Installation Manual*. The WS procedure is initiated automatically when both the hardware and software enable signals become logic high.

Before attempting to enable the drive, the drive must be compensated for the motor and the AKD servo loops must be stable.

Compensation values for many rotary motors are included in a database already loaded into the drive.

NOTE

An unstable system will not function properly during or after the WS process.

Use the default **Wake and Shake** view to configure your system:

Wake and Shake [Learn more about this topic](#)

Wake and Shake is used to establish commutation in drives with certain types of feedback.

Mode
Sets the method used for Wake and Shake

Standard Wake and Shake
 Commutation Alignment

Arm
Motor Phase: deg

Settings

| | | |
|---|--------------------------------------|--------|
| Number of Wake and Shake Loops: | <input type="text" value="5"/> | Counts |
| Time the current vector applied to motor: | <input type="text" value="1"/> | ms |
| Maximum allowed velocity: | <input type="text" value="100.000"/> | rpm |
| Maximum allowed current: | <input type="text" value="2.796"/> | Arms |

[More >>](#)

Mode

The information in this help section primarily applies to **Standard Wake and Shake** (Mode 0), which is the default mode for Wake and Shake. **Commutation Alignment** (Mode 1) is a simple Wake and Shake mode which only uses the WS.IMAX parameter. In Mode 1, The windings are energized at for 1 second at WS.IMAX and the resulting position is used to calculate MOTOR.PHASE. The motion during **Commutation Alignment** is generally large (up to 180 electrical degrees). Select the mode (WS.MODE) according to your application needs.

Arm

Click **Arm** to set WS to start at the next drive enable (WS.ARMWS.ARM). This area also shows the current status of the wake and shake process. See WS.STATE for a detailed explanation of the possible states.

WS.ARM is not restricted to any feedback type.

Settings

- **Number of Wake and Shake Loops.** The WS feature uses the mean of all wake and shake repetitions, called "loops", to establish commutation (WS.NUMLOOPS; see Using WS: Advanced for a discussion of loops). If fewer than five loops are used, commutation may be incorrect, possibly causing poor performance or stability.
- **Time the current vector applied to motor.** This box specifies specifies the duration of the current pulse used for commutation. Increasing this value (WS.T) increases the movement of the system.
- **Maximum allowed velocity.** If a velocity (VL.FB) higher than this value (WS.VTHRESH) is detected while WS is running, then a fault will be generated.
- **Maximum allowed current.** This value (WS.IMAX) is directly proportional to the movement. A value that is too low may fail to cause movement; a value that is too high value may cause an over speed fault.
- **Maximum allowed movement (WS.DISTMAX).** If the total motion from the starting position (the position at the time the drive is enabled after a WS.ARM command) exceeds WS.DISTMAX a fault will occur. Setting WS.DISTMAX to zero disables this feature.
- **Minimum allowed movement (WS.DISTMIN).** If the total motion from the starting position is less than WS.DISTMIN a fault will occur. This will prevent poor initialization from broken wires, incorrect current settings, very high friction, etc. Setting WS.DISTMIN to zero disables this feature.

Wake and Shake, More View

To configure additional WS settings, click **More** at the bottom of the default view to display the following options:

Less <<

| Wake and Shake Movement | | |
|--|-----------|-------------|
| Minimum Distance: | 182.043 | Counts16Bit |
| Maximum Distance: | 2,730.665 | Counts16Bit |
| Delay Time | | |
| Delay Time between current steps: | 5 | ms |
| Delay Time between coarse angle to fine angle: | 50 | ms |
| Delay Time between loops in Mode 0: | 100 | ms |

Wake and Shake Movement

Use these boxes to set values for the maximum (WS.DISTMAX) and minimum (WS.DISTMIN) movement required for finding commutation.

Delay Times

Delay time is the time that elapses when switching different current vectors. Use these boxes to set specific time delays for current steps (WS.TDELAY1), coarse to fine angle (WS.TDELAY2) and time between loops in mode 0 (WS.TDELAY3).

Special Cases for WS

Operation with Motor Brake

An amplifier with a motor brake operates the WS procedure similar to an amplifier without a brake. All precautions and behavioral descriptions above also apply in this case. It is important to note that the brake is automatically applied (motor brake, not holding brake) after the WS process is complete. The brake may cause unexpected movement if the DRV.OPMODE used prior to WS does not retain position. If a force component is present parallel to the track on a linear motor (gravity, load, etc.), or tangential on a rotary motor, the motor may move from the startup position after WS completes and the brake is applied.

If the application requires that the startup position be retained, have the controller system ready to take control immediately after WS is complete. One way to set this control is to have the drive in DRV.OPMODE 1 (digital velocity) or DRV.OPMODE 2 (position mode) on power-up. This precaution keeps the motor stationary after enable.

End of Travel Limits

If anything restricts the motion of the motor, a commutation fault can occur. Examples of situations that may result in faults include the following:

1. If the motor is resting against a rigid endstop, the movement of the motor may be impeded below the minimum threshold set by WS.DISTMIN. This lack of movement causes a fault.
2. If the motor is actuating a limit sensor/switch, the system (PLC, SWLS.LIMIT0 and SWLS.LIMIT1) may be preventing the AKD from producing motion. If descriptive motion is not achieved, the system faults.

Large Load Inertia or High Friction System

Systems with a large load mismatch may need more current than the default setting for correct commutation. Begin with the default value for WS.IMAX and gradually increase or decrease as needed. If adjusting WS.IMAX does not result in a successful commutation, the width of the search pulse can be increased by increasing WS.T.

8.3.9.3 Using WS: Advanced

WS is performed upon enable in order to establish a valid value for MOTOR.PHASE at startup. MOTOR.PHASE is used to calculate electrical phase. With absolute feedback devices, MOTOR.PHASE is a fixed offset between absolute mechanical position and the electrical position. With incremental devices, position is accumulated relative to an initial MOTOR.PHASE. However, at startup, MOTOR.PHASE is invalid since the initial position is random, thus the requirement for the WS process.

WS is a two-step process:

1. Coarse Phase. The drive sequentially pulses a user-specified current, `WS.IMAX`, at each electrical quadrant (0°, 90°, 180°, 270°). Based on the resulting observed movement, an approximate location is calculated.
2. Fine Phase. The drive makes small adjustments to the coarse phase while monitoring movement during velocity mode (command velocity = 0) to find a precise position.

The amplitude of the current pulses in this process equals `WS.IMAX`. The drive repeats these two steps for a user-specified number of times (`WS.NUMLOOPS`) to produce a more accurate estimate of the electrical phase.

The drive normally indicates warning F478 before WS is initiated and successful. If WS fails, the commutation is not valid and the drive indicates one of the following faults:

- F473: Insufficient movement. The maximum movement during WS was less than `WS.DISTMIN`.
- F475: Excessive movement. The movement during WS exceeded `WS.DISTMAX`.
- F476: Fine-Coarse delta too large. The phase calculated during the fine phase and coarse phase differed by more than 10 degrees.
- F478: Overspeed. The feedback velocity (`VL.FB`) exceeded `WS.VTHRESH` during WS.
- F479: Loop angle delta too large. The difference between the phase determined in different cycles (loops) exceeded 30 degrees.
- F482: Commutation not initialized. WS is required (feedback is one of the types listed in the Overview) but WS has not been successfully performed.
- F483 to F485: U, V, or W phase missing . Intermittent or broken motor connection.

Maximizing WS Reliability

The following suggestions will help you achieve successful commutation:

- Incorrect determination of `MOTOR.PHASE` may cause a system runaway. Since the typical movement during correct operation of WS is very small, you can use the velocity overspeed parameters (`WS.VTHRESH` and `DRV.VTHRESH`) to prevent a runaway. Prior to enabling the drive, set `DRV.VTHRESH` 100 mm/s for linear motors or 200 rpm for rotary motors. After a successful enable, `DRV.VTHRESH` can be returned to the normal operating value.
- Set `WS.IMAX` to its default value, `WS.IMAX=0.5*min(MOTOR.IPEAK, DRV.IPEAK)`.
- Set `WS.NUMLOOPS` 20 for best results in many applications.
- `WS.T` specifies the duration for which the search current is applied. With a stable velocity loop, most applications work well with the default value of `WS.T`. The default value causes the software to calculate the width of the search pulse based on the velocity loop proportional gain, `VL.KP`. Increasing `WS.T` effectively increases the movement of the motor during WS, which may be necessary for systems with a low-resolution feedback or high load inertia.
- `WS.IMAX` specifies the amplitude of the current pulse used during the initial/rough commutation. A `WS.IMAX` value that is too low may result in a fault by failing to cause enough movement for commutation. If the value is too high, the preset movement threshold could be exceeded, also resulting in a fault. If the default value is producing faults for too little movement, gradually adjust this parameter to overcome excessive friction and/or load on the system. `WS.IMAX` also specifies the maximum current used in the second stage of commutation. The initial current is 25% of `WS.IMAX`, then steps up to 100% of `WS.IMAX`.
- `FB1.SELECT` selects the type of feedback used by the amplifier. The WS feature is used only for `FB1.SELECT = 11, 21`. The feedback must be configured prior to initiating the WS procedure.
- `MOTOR.BRAKE` If your amplifier has a motor brake, set `MOTOR.BRAKE = 1`. For motors without a brake, set `MOTOR.BRAKE = 0`.

⚠ CAUTION

- Adjust WS.T with extreme caution. Increasing WS.T increases the movement of the system. Applying an incorrect value of WS.T may cause erratic drive behavior.
- When initiating WS, the motor may experience a runaway. Stay clear of all moving parts. Ensure there are properly operating safety devices such as hardware limit switches and suitable end-of-travel limits.
- If WS.NUMLOOPS is less than 5, commutation may be incorrect. This condition may cause faults and/or adversely affect performance or stability. Set WS.NUMLOOPS = 20.

8.3.9.4 Troubleshooting WS

| Problem | Possible Cause | Remedy |
|-----------------------------------|--|--|
| Excessive Movement | <ul style="list-style-type: none"> • Brake slips on vertical system. • External forces on motor too great. | <ul style="list-style-type: none"> • Check brake. • Remove forces acting on motor. • Lower WS.IMAX. |
| Insufficient Movement | <ul style="list-style-type: none"> • Motor brake too rigid. • Motor resting on rigid end-stop. • Too much friction on motor track. • Foreign objects impeding movement of motor. • Motor load very large, and impedes sufficient movement. • WS.DISTMIN manually set too high • WS.IMAX too low | <ol style="list-style-type: none"> a. Check brake b. Check physical location of motor c. Check friction and cleanliness of motor track d. Increase WS.IMAX value |
| Excessive Movement | <ul style="list-style-type: none"> • Brake slips on vertical system. • External forces on motor too great. | <ul style="list-style-type: none"> • Check brake. • Remove forces acting on motor. • Lower WS.IMAX. |
| Insufficient Movement | <ul style="list-style-type: none"> • Motor brake too rigid. • Motor resting on rigid end-stop. • Too much friction on motor track. • Foreign objects impeding movement of motor. • Motor load very large, and impedes sufficient movement. WS.DISTMIN manually set too high • WS.IMAX too low | <ul style="list-style-type: none"> • Check brake. • Check physical location of motor. • Check friction and cleanliness of motor track • Increase WS.IMAX value |
| U, V, or W Phase Missing Fault | Intermittent or broken motor connection. | Check connections to motor phases. |
| Commutation Not Initialized Fault | Wake and Shake is required but WS procedure has previously been canceled (WS.DISARM) or has failed. | Correct errors and rerun WS procedure. |
| Other | <ul style="list-style-type: none"> • Drive not configured correctly. | <ul style="list-style-type: none"> • Check drive compensation • Verify amplifier feedback |

8.3.9.5 Related Parameters

WS Parameters

DRV.IPEAK

FB1.SELECT

MOTOR.BRAKE

MOTOR.PHASE

MOTOR.IPEAK

8.3.10 Commutation Alignment

8.4 Feedback 2

The Feedback 2 screen helps configure how you will be using either connector X9 or X7. The screen uses the term feedback in the labels, but you can also think of this as the “signal” source depending on how you use these connection points.

The **Feedback source** box allows you to choose from three feedback sources:

Source 0 – This simply indicates you are not using either connector as a feedback source.

Source 1 – This is tied to use of connector X9 (consider for emulated encoder connection).

Source 2 – This is tied to use of connector X7 (consider for the high speed opto inputs on the I/O Connector).

Feedback Mode offers three selections as well, depending on how you are using the connection source above.

Mode 0 – The input is configured as A/B signals.

Mode 1 – The input is configured as Step and Direction signals.

Mode 2 – The input is configured as Up-down signals.

The **Resolution** box sets the resolution of the device you have indicated as your input of signal source.

Encoder Emulation Output

The drive offers the flexibility to use connector X9 for an encoder emulation output. This output can be set up using the Encoder Emulation view (see 8.4.1 Encoder Emulation).

The **Emulation Mode** box offers three settings for Encoder Emulation mode:

Mode 0 – The connector is free to use as an input

Mode 1 – The X9 connector is now set up as an encoder emulation with a once per rev index pulse (resolution is set in the next parameter (Emulation Resolution)).

Mode 2 – The X9 connector is now set up as an absolute index pulse

Emulation Resolution sets the resolution desired for the output of the emulated encoder.

The **Index or Absolute offset** is dependent on which emulation mode is selected, allowing you to set an offset for either the index pulse or absolute pulse.

A check box allows you to consider the motor direction based on the encoder output.

8.4.1 Encoder Emulation

8.4.1.1 Overview

The encoder emulation (EEO) connector (X9) can be used as either an output or an input. As an output, you can use this connector for position feedback to an analog controller and for configuring the drive as the master in a master/slave system. You can also use X9 connector as an input for controlling the drive through an A/B, pulse and direction, or up/down command. Common applications for this input include using the drive with a stepper controller and configuring the drive as the slave in a master/slave system.

8.4.1.2 Using Encoder Emulation

You can configure the X9 connector from the **Encoder Emulation** screen by setting the connector function, resolution, and (where applicable) input position. The **Electronic Gearing** screen also has provisions for setting the function of the X9 connector.

8.4.1.3 Function Settings

Function settings for the X9 connector are assigned through the "DRV.EMUEMODE" (=> p. 377) parameter.

Mode 0: Disable X9

Input (No EEO Output). The EEO connector (X9) is configured as an input. See Feedback 2 (FB2) for definitions for the EEO Connector (X9). Note that this mode is also deprecated and will behave as it did with firmware versions prior to M_01-03-00-000.

In earlier firmware versions, this mode indicates that the EEO connector is not operative.

Output Modes 1 and 2

In the output modes, the drive calculates the motor shaft position from the cyclic- absolute signals of the primary feedback, generating incremental-encoder compatible pulses from this information. Pulse outputs on the SubD connector X9 are three signals: A, B and index, with 90° phase difference (i.e. in quadrature, hence the alternative term "A quad B" output), with a zero pulse.

If you are using the AKD as a master, the slave drives use the master's encoder output signals as command input and follows these commands (velocity and direction). The drives operate from an internal supply voltage.

Mode 1– Aquad B with once per rev index pulse

[Output Mode 1 - A quad B with Once per Rev Index Pulse Connection Diagram](#)

This output mode simulates an encoder signal from the X9 port to another AKD or external controller. EEO resolution ("DRV.EMUERES" (=> p. 379)) defines how many counts are outputted for one revolution of the primary feedback.

Index offset ("DRV.EMUEZOFFSET" (=> p. 380)) determines the point during the revolution of the primary feedback when the index pulse (X9 pins 7&8) is output through the X9 port. The pulse will occur once every revolution of the primary feedback at the positive value of the offset.



Encoder Emulation

The encoder emulation page is used to configure the X9 connector on the drive.

| | |
|---------------|---|
| Function: | <input type="text" value="1 - Output - With once per rev index pulse"/> |
| Resolution: | <input type="text" value="2,048"/> |
| Index Offset: | <input type="text" value="1,024"/> |

In this case, the Index is offset 180 degrees, or halfway through the revolution of the primary feedback.

Mode 2– A quad B with absolute index pulse

[Output Mode 2 - A quad B with Absolute Index Pulse Connection Diagram](#)

When mode 2 is chosen, a box will appear for an absolute index point to be entered ("DRV.EMUEMTURN" (=> p. 378)). The absolute index pulse will be output when the X9 output position reaches the assigned value. The index will be output when the feedback position in counts matches this parameter. In the example below, the absolute index pulse output will occur after 10 positive revolutions of the motor. The absolute index is referenced from the zero point of the primary feedback and is reset when the drive is homed.

| | |
|------------------|---|
| Function: | <input type="text" value="2 - Output - With absolute index pulse"/> |
| Resolution: | <input type="text" value="2,048"/> |
| Absolute Offset: | <input type="text" value="20,480"/> |

Input Modes 3, 4, and 5

The X9 connector is also capable of input modes. These input modes correspond to the signal types described below. The **Electronic Gearing** screen also includes provisions for setting the function of the X9 connector for input modes.

Mode 3-A quad B signals

[Input Mode 3 - A quad B Signals Connection Diagram](#)

Input mode 3 allows an A quad B encoder or the encoder emulation output of another drive to be connected and used as a commander encoder, dual loop feedback, gearing, or camming input.

Mode 4-Pulse/direction signals

[Input Mode 4 - Pulse/direction Signals Connection Diagram](#)

Input mode 4 allows the drive to be connected to a third-party stepper-motor controller. The number of steps can be adjusted so that the drive can be adapted to match the step-direction signals of any stepper controller.

Mode 5-Up/down signals

[Input Mode 5 - Up/down Signals Connection Diagram](#)

The drive can be connected to a third-party controller which delivers up-down signals.

Mode 6-Output – with once per rev index and Input – Step and Direction

This mode allows you to output emulated encoder signals from the X9 (Emulated Encoder) connector, and input a step handwheel signal into the X7 (High Speed Opto IO) connector simultaneously. Mode 6 and 7 are identical, with the exception that the emulated encoder has an incremental Z pulse (Z pulse occurs every revolution) in mode 6, and an absolute Z pulse (Z pulse occurs at one absolute position) in mode 7.

Mode 7 – Output – with absolute Index and Input – Step and Direction

This mode allows you to output emulated encoder signals from the X9 (Emulated Encoder) connector, and input a step handwheel signal into the X7 (High Speed Opto IO) connector simultaneously. Mode 6 and 7 are identical, with the exception that the emulated encoder has an incremental Z pulse (Z pulse occurs every revolution) in mode 6, and an absolute Z pulse (Z pulse occurs at one absolute position) in mode 7.

8.4.1.4 Resolution

The resolution setting defines how many counts are output for one revolution of the primary feedback (when X9 is configured as an output), or how many counts will be considered a full revolution of the input signal from an external controller (when X9 is configured as an input). The resolution value is post-quadrature; for example, a 1,000 count encoder has a resolution of 4,000 counts.

Note: If the resolution value is set to 0, then the X9 connector will not produce a command.

8.4.1.5 Related Parameters and Commands

The DRV.EMUEMODE parameter sets the EEO output and input modes. The resolution (before multiplication) is set by the DRV.EMUERES function. The DRV.EMUEZOFFSET parameter adjusts and saves the zero pulse position within one mechanical turn. DRV.EMUEMTURN sets the absolute index point in mode 2 and DRV.HANDWHEEL sets the position at which the index pulse is output in output mode 2.

DRV.EMUEDIR

DRV.EMUEMODE

DRV.EMUEMTURN

DRV.EMUERES

DRV.EMUEZOFFSET

DRV.HANDWHEEL

8.4.1 Encoder Emulation

8.5 Non-Plug and Play Feedback Devices

To set up an AKD drive with a non-plug and play feedback device (for example a resolver or an encoder), you must select a motor from the list of standard or custom motors or enter the motor parameters manually. Once the motor data is entered into WorkBench, an initial set of parameters can be calculated and downloaded to the drive.

8.6 Parameters

The following parameters can be initialized to their default values or calculated from user-supplied motor data:

| | | | |
|-------------|-----------------|---------------------------|--------------------------------------|
| IL.KP | IL.LIMITN | VL.KP | PL.KI = 0 (Default Value) |
| IL.KFFACC | IL.LIMITP | VL.KI | PL.INTINMAX = 0.419 (Default Value) |
| IL.KBUSFF | IL.PWMFREQ | VL.LIMITN | PL.INTOUTMAX = 0.419 (Default Value) |
| IL.FRICTION | IL.KVFF | VL.LIMITP | MOTOR.IPEAK |
| IL.OFFSET | IL.FOLDFTHRESHU | VL.THRESH | MOTOR.ICONT |
| IL.INTEN | IL.FOLDWTHRESH | VL.KVFF | MOTOR.PITCH |
| IL.IVFB | IL.MFOLDD | PL.KP | MOTOR.POLES |
| IL.KPDRATIO | IL.MFOLDT | PL.KD = 0 (Default Value) | MOTOR.TYPE |

8.7 Calculations

WorkBench uses the following equations to calculate parameter values.

8.7.1 Current Loop

The current loop proportional gain (IL.KP) must be such that the current loop closed loop crossover frequency/bandwidth (BW) nominal is the lesser of 2000 Hz or (PWM Frequency/4).

Then, with this frequency:

$$IL.KP = 2 * \pi * (\text{desired bandwidth in Hz}) * (\text{motor L line-line in H})$$

Setting the D and Q components

8.7.2 Velocity Loop

$$VL.KP = (2 * \pi * 75) * (2 * Jm / Kt) = 300 * \pi * Jm / Kt$$

$$VL.KI = 5$$

8.7.3 Position Loop

$$PL.KP = ??$$

8.7.4 Slider Tuning

The slider tuning algorithm in WorkBench currently uses the following algorithm.

8.7.4.1 Input - Motor Data

The values for inertia, J_m (Kg / cm²), and torque constant, K_t (Nm/A), are obtained from either the SFD or the motor model number that you select.

8.7.4.2 Constants

Velocity Loop Bandwidth – BW = The default value is 75Hz.

Input - Inertia Ratio – Q = The default is 1.

8.7.4.3 Output - Control Loop Gains

$$VL.KP = 2 * \pi * BW * Jm * (1+Q) * 0.0001 / Kt$$

$$VL.KPI = BW * 0.08 - 1 \quad (\text{minimum of } 1)$$

$$PL.KP = BW / 5$$

8.8 Foldback

The foldback feature in the AKD protects both the motor and the drive from overheating. Two current foldback algorithms run in parallel in the drive: the drive foldback algorithm and the motor foldback algorithm. Each algorithm uses different sets of parameters. Each algorithm has its own foldback current limit, IL.DIFOLD (drive foldback) and IL.MIFOLD (motor foldback). The overall foldback current limit is the minimum of the two at any given moment.

$$IL.IFOLD = \min(IL.DIFOLD, IL.MIFOLD).$$

Foldback is not the same as current limits. Instantaneous current limits for the drive are set by the positive peak current (IL.LIMITP) and negative peak current (IL.LIMITN) in the Limits view in WorkBench. The foldback algorithms may reduce the current output to the motor in spite of the current limit settings.

8.8.1 Drive Foldback

The drive foldback algorithm monitors current feedback; since this is a monitoring function, the drive foldback parameters are not user configurable. If the current feedback exceeds the continuous current rating of the drive (DRV.ICONT), then the algorithm decreases the current to the DRV.ICONT level. For example, under a step command input condition, the foldback algorithm allows maximum peak current (DRV.IPEAK) output from the drive for a short period of time (up to IL.DFOLDD time), after which the drive begins an exponential foldback (with time constant of IL.DFOLDT) of the current to the drive's continuous current.

It takes a few seconds for the exponential decay to drop from the drive's peak current to its continuous level. A recovery time, where the feedback current is below DRV.ICONT level, is required to allow current above DRV.ICONT level again. A recovery time of IL.DFOLDR with 0 current allows the drive to apply DRV.IPEAK current for IL.DFOLDD time.

8.8.2 Setting up motor foldback

Motor foldback is set up automatically when using a plug and play motor or when a particular motor is selected from the WorkBench database.

If you are using a custom motor, use the Motor view in WorkBench to set custom values needed for foldback configuration. The parameter entries required for the drive to apply motor foldback protection properly are coil thermal constant (MOTOR.CTFO), peak current of the motor (MOTOR.IPEAK), and continuous current of the motor (MOTOR.ICONT). These values are used to setup the algorithm for motor foldback.

8.8.3 Setting Fault and Warning Levels

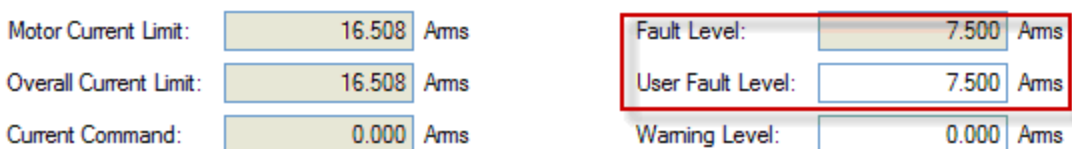
The **Motor Current Limit** and **Overall Current Limit** boxes show status variables that are constantly updated by the foldback algorithm. As current is applied above the drive or motor continuous rating, the capacity for the application of peak current starts to decrease. The motor current limit and overall current limit are actively decreased. If the move profile requires less than continuous current rating for a period of time, the Motor Current Limit and Overall Current Limit begin to increase until they reach maximum foldback capacity once again.

When Motor Current Limit or Overall Current Limit < Warning Level, an n524 status warning is triggered. When Motor Current Limit or Overall Current Limit < Fault Level, an F524 fault is triggered and the drive power stage is disabled. The load then coasts to a stop.

In the **Foldback** screen, setting the Warning Level to 0 turns off the warning feature. Otherwise, the Warning Level must be set above the Fault Level, but below the Motor Current Limit and Overall Current Limit to trigger a warning.

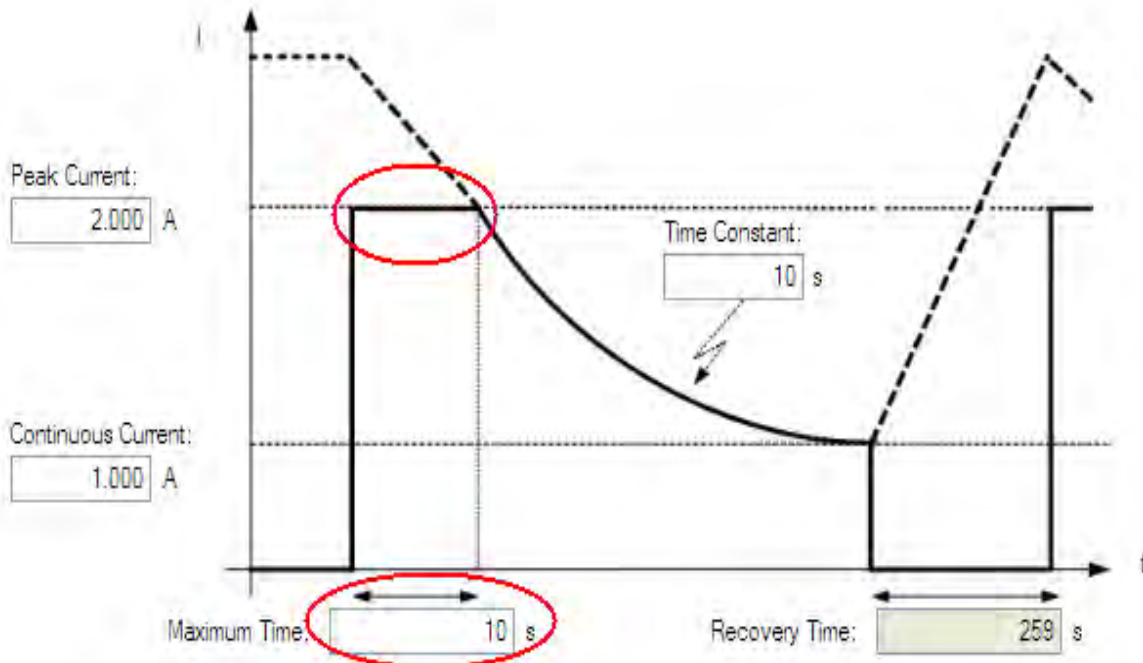


If User Fault Level is set above Fault Level, the User Fault Level will be ignored. The User Fault Level is used to increase the time the drive is operated in foldback mode without faulting. For instance if the default Fault Level is 9.000 Ams and a User Fault Level is set to 7.5 Ams, the Fault Level is changed to 7.5 Ams. This configuration effectively increases the time that foldback will be applied to the drive before faulting.



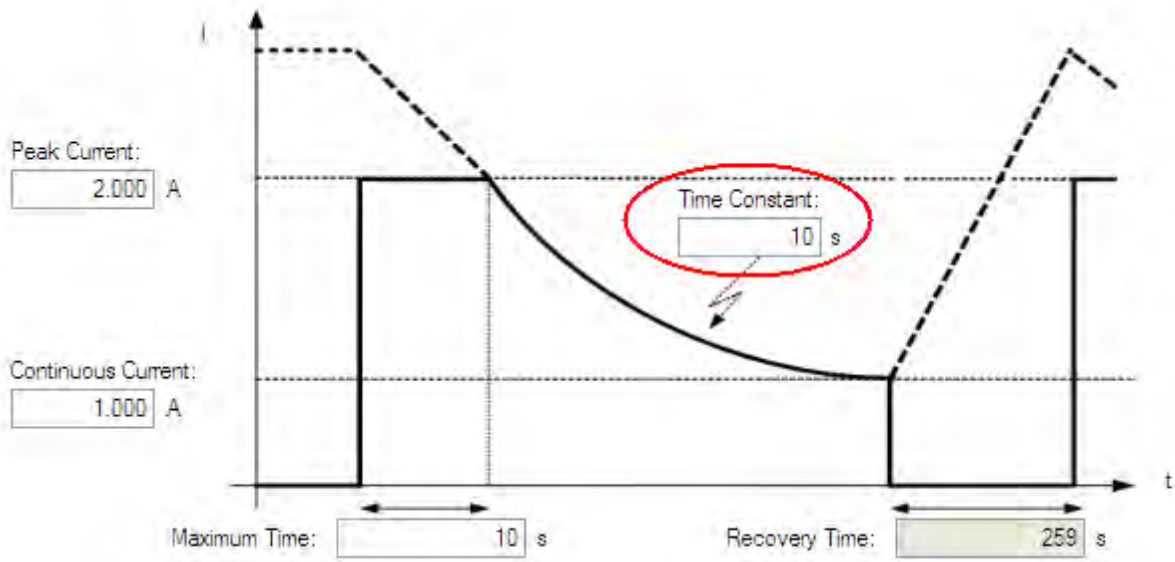
8.8.4 Motor Peak Current Time

Peak current (MOTOR.IPEAK) along with coil thermal constant (MOTOR.CTFO) are used to determine the maximum time the motor can sustain peak current. The maximum time (IL.MFOLDD) is displayed in the **Foldback** screen as shown below:



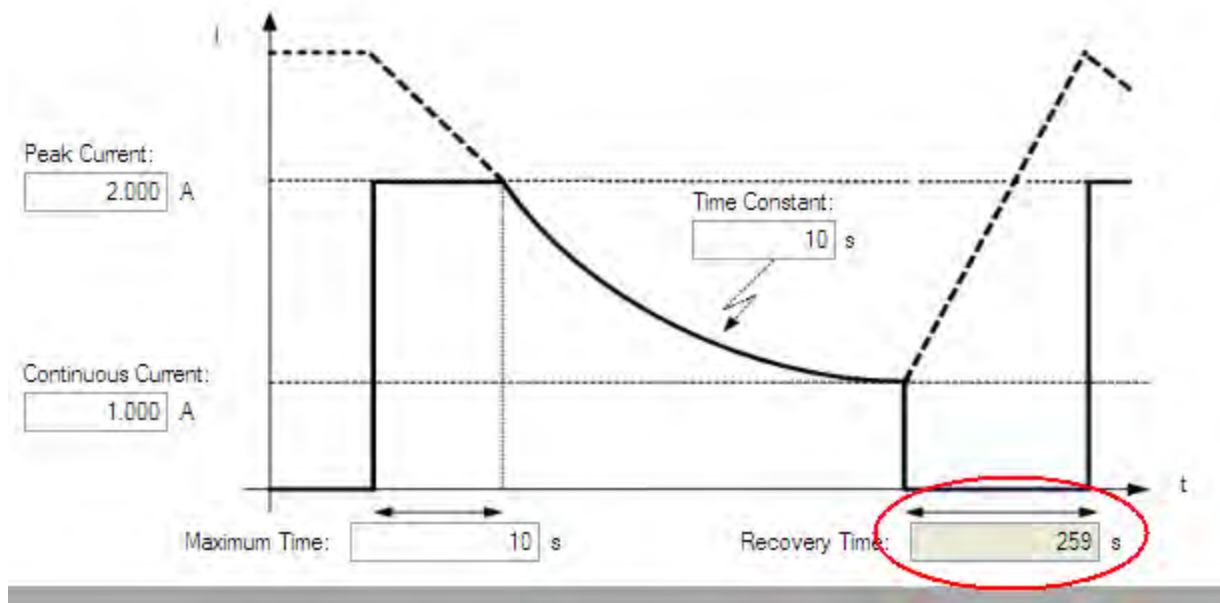
8.8.5 Motor Foldback Ramp

Once the maximum time for motor peak current has elapsed, if the move profile still demands peak current from AKD, the drive will exponentially lower the current applied to the motor. The Time Constant (IL.MFOLDT) dictates the profile. A smaller time constant represents a steeper decline in current applied to the motor.



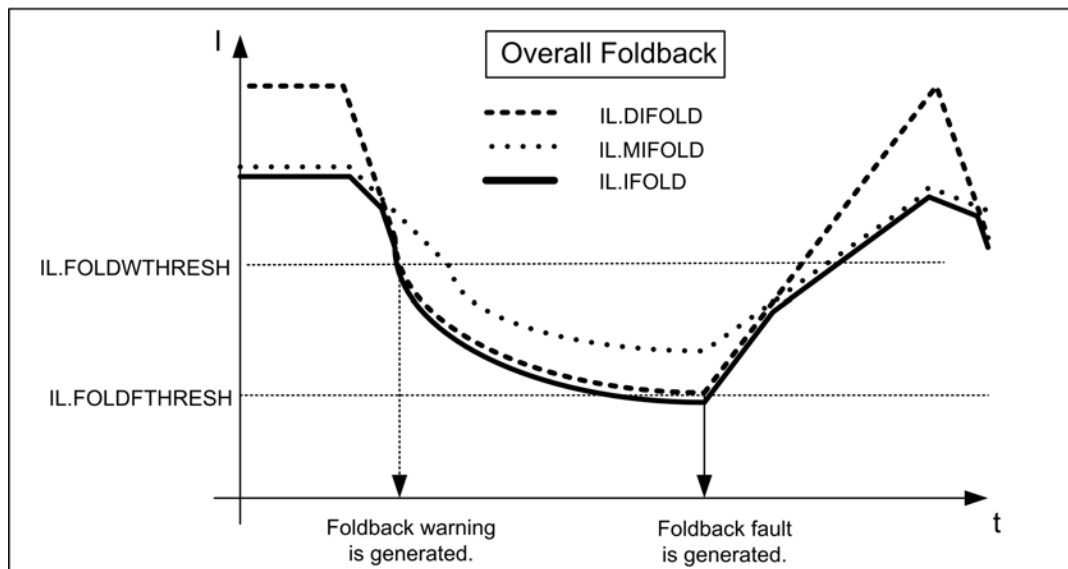
8.8.6 Motor Recovery

Once the peak motor current available has reached the continuous current of the motor, the motor needs Recovery Time (IL.MFOLDR) to cool down. Full Recovery Time (IL.MFOLDR) at 0 current is required for the motor to reach full maximum capacity in the shortest amount of time. The drive can command a current less than continuous current to continue driving the load, but the recovery time for full maximum capacity is increased.



8.9 Overall Foldback

The overall limit is the momentary minimum value between the drive foldback and the motor foldback. The overall foldback is shown in the diagram below. You can set the warning and the fault levels as shown in the diagram.



8.10 Brake

The brake output on connector X2 controls a mechanical brake that optionally may be fitted to a motor. The brake is applied and released relative to the **Drive Active** state of the drive. You can modify the release and apply delays using the parameters shown below.

| Button or Dialog Box | Description | Parameter |
|----------------------|--|---------------------------------|
| Brake Control | Does this motor have a brake? | MOTOR.BRAKE |
| Brake Release Delay | The time between the drive being active and the brake being released. | MOTOR.TBRAKERLS |
| Brake Apply Delay | The time between the brake being applied and the drive not being active. | MOTOR.TBRAKEAPP |

8.11 Using Position Capture

8.11.1 Overview

Position capture allows you to precisely determine what the motor position (or drive clock time) was when a specific event triggers. The AKD drive uses two independent captures, which operate similarly. The descriptions provided in this user guide refer to Capture 0, but also apply to Capture 1.

Position capture is used in precision environments, where the motor may be moving at very high velocities, an IO is triggered, and you must know exactly where the motor was when the event occurred. Homing algorithms often use position capture.

Position capture will capture the motor position (or drive clock time) when the capture trigger is activated. The capture engine operates on a faster clock than the 16 kHz clock used in the scope and recorder. This faster clock allows the position capture to obtain more accurate results than those obtained using the scope or recorder clock.

8.11.2 Configuring Position Capture

To configure the position capture, select **Position Capture** from the **Settings** group:

PC Position Capture [Learn more about this topic](#)

The drive will be able to capture the position of the axes

Position Capture 0

Capture Source: 0 - DIN 1

Capture Mode: 0 - Standard position

Arming

Captured Position: 0 Counts16Bit

Capture Parameters

Capture Edge:

Pre Conditions

Pre Condition: 0 - Trigger edge (ignore prec)

Pre Edge:

Pre Select:

Position Capture 1

Capture Source:

Capture Mode:

Arming

Captured Position: Parameter is not set

Capture Parameters

Capture Edge:

Pre Conditions

Pre Condition:

Pre Edge:

Pre Select:

8.11.2.1 Setting the Capture Source (CAP0.TRIGGER)

The capture source determines which input on the drive causes the position capture to trigger.

Capture Source Options:

- 0 - 6: These options trigger on the Digital Input 1 pin through Digital Input 7 pin, respectively.
- 7- 9: These options trigger on the X9 connector RS485 Input 1 pin through RS485 Input 3 pin, respectively.
- 10: This option triggers on the primary encoder index.

8.11.2.2 Setting the Capture Mode (CAP0.MODE)

The capture mode determines what information is saved on the drive when the capture triggers.

Capture mode options:

- 0 – Standard Position.** Captures the motor position in drive units

- 1 – Drive Internal Time.** Captures the time of the trigger in ns
- 2 – Distributed Clock Time.** Captures the network (Ethercat) distributed clock time
- 3 – Primary Encoder Signal.** Captures the motor position triggering on primary encoder index. This mode automatically rearms after each trigger.

If either **0 - Standard Position** or **3 - Primary Encoder Signal** is selected, delays may occur and are associated with feedback devices that are digital or interpolated .

8.11.2.3 Arming and Retrieving the Capture Value (CAP0.EN and CAP0.T)

CAP0.EN arms the capture and CAP0.T retrieves the capture value. Once you have configured the capture, you must arm it before it will trigger. Click **Arm** (1) to arm the capture.

Once the capture is armed, when it triggers, the captured value will be displayed below the Arm button (2).

8.11.2.4 Setting the Capture Edge (CAP0.EDGE)

The capture edge determines which input state change triggers the capture.

Capture Edge Options:

- 1 – Rising Edge.** Captures when the input signal goes high, from a low state.
- 2 – Falling Edge.** Captures when the input signal goes low, from a high state.
- 3 – Both Edges.** Captures any time the input signal changes state.

8.11.2.5 Setting the Pre-Condition Event: (CAP0.EVENT)

The Capture Pre-Condition Event gives the user more flexibility in setting what conditions must be present for the Capture to trigger.

Event Options:

- 0 – No precondition.** Capture triggers as soon as the capture edge occurs.
- 1 – Trigger Edge after precondition.** Captures triggers only when the precondition occurs before the capture edge occurs.
- 2 – Trigger Edge while precondition = 1.** Captures triggers only while the precondition is evaluated and is true while the capture edge occurs.
- 3 – Trigger Edge while precondition = 0.** Captures triggers only while the precondition is evaluated and is false while the capture edge occurs

8.11.2.6 Setting up a Pre-Condition for complex capture

Setting the Precondition Edge: (Terminal Command: CAP0.PREEDGE)

The pre-edge determines what input state change triggers the precondition. This feature operates the same as the capture edge described above.

Setting the Pre-Condition Select: (Terminal Command: CAP0.PRESELECT)

The preselect chooses what input source will trigger the precondition (based on the preedge setting, and the pre-filter setting). This feature operates the same as the capture source described above.

8.11.3 Related Parameters

CAP Parameters

9 Selecting Units for Your Application

| | |
|-------------------------------------|----|
| 9.1 Selecting and Saving Units..... | 84 |
| 9.2 Units Example..... | 84 |

9.1 Selecting and Saving Units

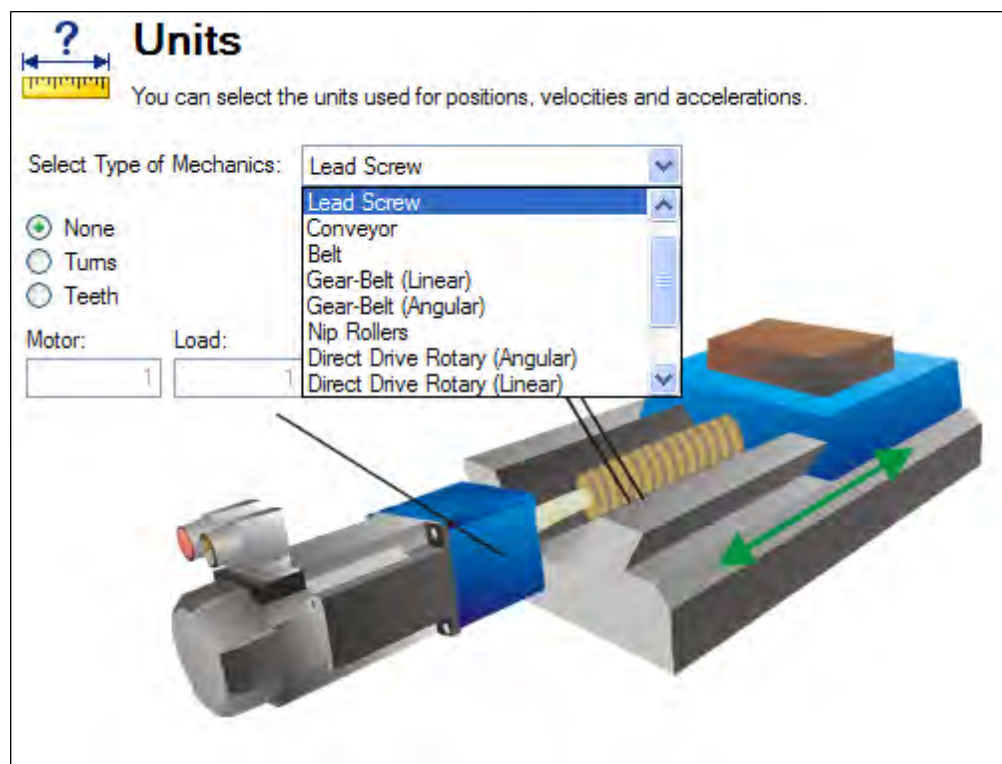
The drive uses three primary measures of motion: position, velocity, and acceleration. You must first choose units of measurement for each of these, and then enter the details of the mechanics in order to scale the chosen units appropriately.

The settings for units are automatically applied. The unit settings in the drive will reflect last settings made in the units screen before exiting. To save settings into the nonvolatile memory of the drive, click the **Save to Drive** button in the toolbar.

9.2 Units Example

1. Select type of mechanics

To set the units in the drive for a particular application, first select the type of mechanics present.



2. Select Position Units

By default, position is measured in counts. Counts are the smallest unit of position that can be represented in the drive. This unit of measure equates to 4,294,967,296 counts/revolution of the motor. You can use the units screen to change this measurement into a meaningful scale relating to the units of the application.


For position units, five selections are available:

- 0—Counts (4,294,967,296/rev)
- 1—Radians (2π / rev)
- 2—Degree (360/rev)
- 3—Custom (set by user according to mechanics of machine)
- 4—Counts (16-bit) (65,536 /rev)

Custom Units



The drive uses full 32-bit quantization for internal calculations regardless of unit settings. User units settings will not affect performance, resolution, or accuracy of the servo system.

Choose “3-Custom”, and then select your desired position units, for example, millimeters.

Position Unit: 

Velocity Unit:

Acceleration Unit:

Custom Position Unit:  

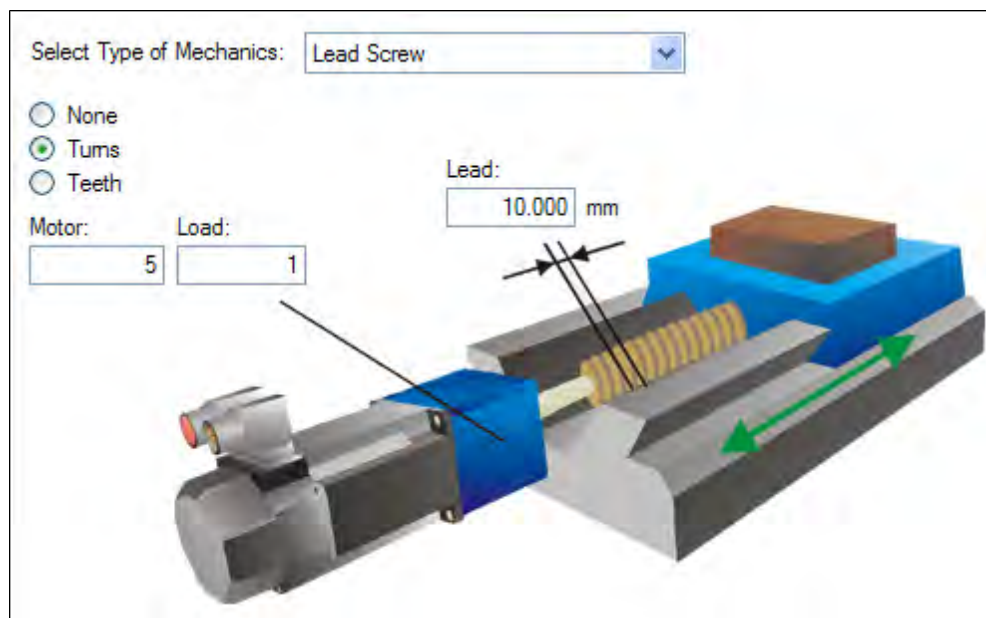
3. Select Velocity Units

For velocity, select **Custom/s** to set the measurement to mm/sec.

4. Select Acceleration Units

For acceleration, select **Custom/s^2** to set the measurement to mm/sec².

Once the chosen units of measurement are established, the details of the mechanics need to be entered to establish the scaling of the chosen units.



In this example, a lead screw with a 10 mm lead and a 5:1 gearbox has been selected. A 10 mm lead means the load will traverse 10 mm for every turn of the screw. A scale factor is applied based on the values entered for the chosen mechanics. This scaling is done through UNIT.PIN and UNIT.POUT parameters in the AKD, which are automatically adjusted when the mechanics values are entered.

In this example, scaling is adjusted as follows:

Custom

UNIT.PIN Label: = UNIT.POUT rev.

UNIT.PIN is calculated as follows:

$$10 \text{ mm/screw turn} * 1 \text{ screw turn/5 motor turns} = 2 \text{ mm/motor turn}$$

UNIT.PIN and UNIT.POUT can be directly entered using the **Motor Only** selection from the **Select Type of Mechanics** box.

9.2.1 Related Parameters

UNIT Parameters

DRV.NVSAVE

MOTOR.TYPE

10 Configuring General Drive Settings

| | |
|---|-----|
| 10.1 Digital Inputs and Outputs..... | 88 |
| 10.2 Command Buffer..... | 96 |
| 10.3 Digital Inputs (X7/X8)..... | 99 |
| 10.4 Analog Input..... | 103 |
| 10.5 Analog Output..... | 103 |
| 10.6 I/O Connection..... | 104 |
| 10.7 Electronic Gearing..... | 106 |
| 10.8 Programmable Limit Switch..... | 109 |
| 10.9 Enable/Disable..... | 111 |
| 10.10 Controlled Stop..... | 115 |
| 10.11 Dynamic Braking..... | 117 |
| 10.12 Emergency Stop..... | 118 |
| 10.13 Safe Torque Off (STO)..... | 118 |
| 10.14 Under Voltage Fault Behavior..... | 118 |

10.1 Digital Inputs and Outputs

10.1.1 Overview

The drive has programmable digital inputs and outputs that you can use to initiate motion, control auxiliary devices, or trigger other actions. The inputs and outputs should be wired according to the instructions in the drive Installation Manual. Note that the input on pin 4 of the X8 connector is dedicated as an enable input.

10.1.2 Using Digital I/O

Once wired correctly, digital inputs and outputs can be used for a variety of functions such as to trigger auxiliary devices, initiate homing moves or other motion tasks, or set travel limits. This section describes the specific functionality of the programmable I/O.

I/O Tip: When using I/O devices, you must carefully consider the type of device you use for switches. An unsuitable switch can cause switch bounce, which in turn can cause erroneous triggers to occur. For example a low cost xx switch, as it is toggled, will bounce a few times before it turns on or off. A device that is monitoring these inputs frequently may interpret the bounce as multiple triggers of that I/O. The drive has the ability to reduce this type of error using some debounce techniques to ignore sudden state changes caused by bounces.

10.1.3 Digital Inputs

Digital inputs can be set in different modes based on the desired function. These functions are outlined below:

Mode 0: Off

This mode is the non-use state and is the default setting for the drive. This mode is valid for all opmodes and command source combinations.

Mode 1: Fault Reset

When an input configured with this mode becomes active, the drive will try to clear all active faults. This mode is edge triggered, so the action occurs only once. If the condition that triggered the fault is still present, the fault condition will remain. See 17.1 Fault and Warning Messages for details regarding the behavior of individual faults.

This mode is valid for all opmodes and command source combinations.

Mode 2: Start Motion Task

This mode is used to start motion task number x, where x = the value of the associated input parameter. This input will trigger a motion task number as defined in the extra parameter field for this input.

This mode is valid for opmode 2 (position) and command source 0 (service) only.

Once started, the motion task will run until completed. Changes on the input that started the task will be ignored until the motion task is complete. If multiple inputs are configured to start a task, all of these inputs will be ignored until the task is complete. If a motion task is already active in the drive, changes on this input will be ignored.

Example:

```

->DIN1.MODE 2 - sets the input mode to be Start Motion Task
->DIN1.PARAM 1 - sets the Motion Task start to 1.
->MT.LIST - confirms that Motion Task 1 exists.
->10.000 [counts] 1000.000 [rpm] 0 1001.358 [rpm/s] 1001.358 [rpm/s] 0
0 0 [ms]
<Create a rising edge of the input>
<Motion Task 1 executed>

```

Mode 3: Motion Task Select Bit

This mode is used to select the motion tasks that are stored in the drive (numbers 1 to 127) or the reference traverse/homing (0). The motion task number is presented externally at the digital inputs. The motion task set by this mode will be executed when digital input assigned to mode 4 (motion task start selected) gets a rising edge.

This mode is valid for opmode 2 (position) and command source 0 (service) only.

Example

Assume:

DIN1.MODE = DIN2.MODE=DIN3.MODE =3

The state of input 1 and 3 is 1.

The state of input 2 is 0.

Motion task 5 ($5 = 2^0 + 2^2$) will be executed.

Mode 4: Motion Task Start Selected

This mode is used to start the motion task that is stored in the drive by giving the motion task number. This input uses a secondary variable for the motion task number to be started with the Input trigger. The secondary variable is set by mode 3 (Motion task select bit).

Motion task number "0" initiates homing/reference traverse. A rising edge starts the motion task and a falling edge cancels the motion task.

This mode is valid for opmode 2 (position) and command source 0 (service) only.

Mode 5: Start Home

This mode is used to start the homing motion task on the rising edge. The falling edge has no effect on this input mode of operation.

This mode is valid for opmode 2 (position) and command source 0 (service) only.

Mode 6: Start Jog

This mode is used to start a jog move. This input mode utilizes a secondary variable for the jog's velocity. The jog will start upon a rising edge. A falling edge stops the jog.

This mode is valid for opmodes 1 (velocity) and 2 (position) and command source 0 (service).

Mode 7: Reserved

Mode 8: Zero Latch

This mode is used to define the current drive position as the zero pulse for the drive EEO and sets the incremental encoder zero pulse offset. The current position, depending on the incremental encoder resolution that is set, is calculated at the rising edge and stored as an offset. An automatic save is then generated. This function is used to perform an automatic setting of the zero pulse in one turn of the motor.

This mode is valid for all opmodes and command source combinations.

Mode 9: Command Buffer

This mode is used to execute four different sets of command buffers. Each set contains two buffers: low and high, for a total of eight buffers. DINx.PARAM for this mode can be 1 to 4, and determines which set of buffers to use.

To set the high and low values of the eight buffers from the terminal screen, use the commands DIN.HCMDx and DIN.LCMDx ($1 \leq x \leq 4$). Use ";" to separate the two buffer commands. Each buffer contains up to 128 characters.

Example

->DIN1.MODE 9 (sets command buffer mode to digital input 1)

->DIN1.PARAM 1 (sets the first set of buffers to digital input 1)

->DIN.HCMD1 DRV.OPMOE 1; (sets high command buffer)

->DIN.LCMD1 DRV.OPMOE 0; (sets low command buffer)

Under this configuration, a rising edge in digital input 1 will set DRV.OPMODE to 1 and a falling edge will set DRV.OPMODE to 0.

You can also set the command buffers from the **Digital I/O** view in WorkBench; see 10.2 Command Buffer

This mode is valid for all opmodes and command source combinations.

Mode 10: Control Fault Relay

This mode is used to create an external fault.

Input state is 0 – drive regular behavior

Input state is 1 – “Fault 245 – external fault” is issued.

This mode is valid for all opmodes and command source combinations.

Mode 11: Home reference

This mode is used to receive a physical home reference switch located on the machine to use for the different Home Types.

This mode is valid for opmode 2 (position) and command source 0 (service) only.

Mode 12: Reserved

Mode 13: Controlled Stop

This mode is used to stop the motor using the deceleration variable ramp. If zero velocity is reached, the power stage is then disabled. Also see controlled stop (CS) parameters and commands and Controlled Stop.

This mode is valid for all opmodes and command sources.

Mode 14: Reserved

Mode 15: Quick Stop

This mode is used to stop the motor. It is equivalent to issuing a DRV.STOP command.

This mode is valid for all opmodes and command sources 0 (service) and 2 (electronic gearing).

Mode 16: Activate Electronic Gearing

This mode starts/activates an electronic gearing procedure upon a rising edge.

This mode is valid for opmode 2 (position) and command source 2 (electronic gearing).

Mode 17: Activate Electronic Gear Position Shift

This mode is used to add a position shift to the gearing upon a rising edge. The distance of the position shift is set by the secondary variable. The secondary variable is set by DINx.PARAM. The parameter is in position units and is used to incorporate a phase shift while operating in electronic gearing mode.

This mode is valid for opmode 2 (position) and command source 2 (electronic gearing).

Example

An input is set to add a 180 degree "phase shift" when triggered. As the drive is following the electronic gearing input, the input is triggered and the motor will follow the drive acceleration and deceleration rates to shift 180 degrees while maintaining the gearing synchronization.

Mode 18: Positive Limit Switch

This mode will cause the input to operate as the positive limit switch. If the positive limit switch input is triggered (goes low), the positive direction motion will then be stopped.

This mode is valid for all opmodes and command source combinations.

⚠ CAUTION

When setting up the hardware limit switches, you must be certain that the switch remains in the triggered state until you move off of the switch. A very low deceleration rate combined with a high approach velocity may overshoot the switch. This action will cause the position limit warning to be canceled. The warning is not latched, therefore if the switch is overshoot, additional movement in the same direction (if commanded) will be possible. This movement can cause machine damage.

Mode 19: Negative Limit Switch

This mode will cause the input to operate as the negative limit switch. If the negative limit switch input is triggered (goes low), the negative direction motion will then be stopped.

This mode is valid for all opmodes and command source combinations.

⚠ CAUTION

When setting up the hardware limit switches, you must be certain that the switch remains in the triggered state until you move off of the switch. A very low deceleration rate combined with a high approach velocity may overshoot the switch. This action cancels the position limit warning. The warning is not latched, therefore if the switch is overshoot, additional movement in the same direction (if commanded) will be possible. This movement can cause machine damage.

Mode 20: Brake Release

This mode is used to apply or release the brake when the drive is not active.

Input = 0: the drive controls the brake (regular drive behavior)

Input = 1: the user controls the brake (apply or release using commands)

This mode is valid for all opmodes and command source combinations.

Mode 21: Current Limit

This mode is used to limit the drive current. The current limit is set by a secondary variable; use DINx.PARAM to set the secondary variable.

This mode is valid for all opmodes and command source combinations.

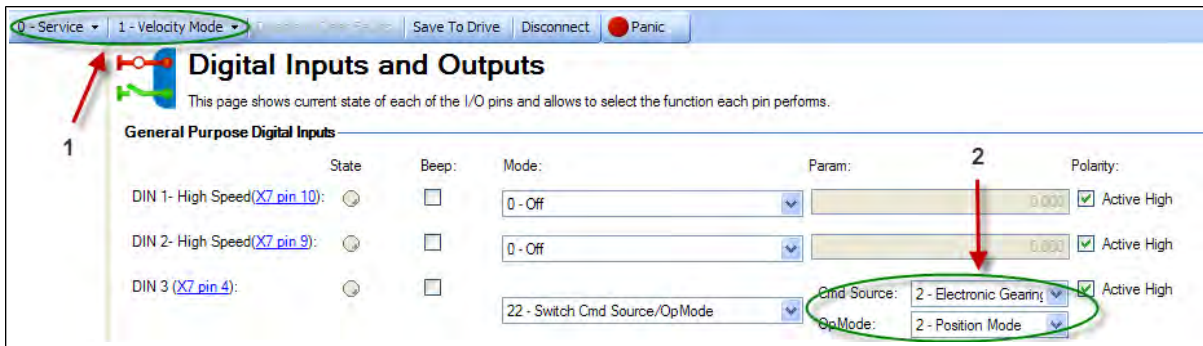
Mode 22: Switch Command Source and Opmode

This mode is used to switch between the present command source/opmode and another command source/opmode setting predetermined by the user upon the level change of a digital input. This mode is valid for all opmodes and command source combinations. Arrow 1 in the screen shot below indicates the present command source/opmode that the drive is set for. This is the mode the drive is in when the digital input is not high. This low state is determined by the original settings for DRV.CMDSOURCE and DRV.OPMODE.

The drive will switch into the command source/opmode setting shown in arrow 2 when the digital input level changes to high. This setting is stored by DINx.PARAM and is edited with the drop down boxes at arrow 2.

NOTE

When the digital input is switched high, DRV.CMDSOURCE and DRV.OPMODE will take the values defined by DINx.PARAM. Do not perform a "drive save" in this state, or the low state and high state settings will become the same.



| DINX.PARAM | Command Source | Opmode |
|------------|----------------------|------------|
| 0 | 0-service | 0-torque |
| 1 | 0-service | 1-velocity |
| 2 | 0-service | 2-position |
| 10 | 1-fieldbus | 0-torque |
| 11 | 1-fieldbus | 1-velocity |
| 12 | 1-fieldbus | 2-position |
| N/A | 2-electronic gearing | 0-torque |
| N/A | 2-electronic gearing | 1-velocity |
| 22 | 2-electronic gearing | 2-position |
| 30 | 3-analog | 0-torque |
| 31 | 3-analog | 1-velocity |
| 32 | 3-analog | 2-position |

If more than one digital input is configured to this mode, and one of them is active, then the command source/op-mode combination configured for that input will be active. If additional inputs become active, the command source/opmode combination configured for the lowest numbered will be the active one.

Example

Assume:

Input 1 is configured for electronic gearing/position.

Input 2 is configured for service/velocity.

Input 3 is configured for fieldbus/position.

The system is in service/torque.

Result:

With no inputs active, the system remains in service/torque.

If input 3 goes active first, then the system will go to fieldbus/position

If input 1 then goes active, then the lowest active input is now 1 so the system will go to electronic gearing/position

If input 2 then goes active, then the lowest active input is still 1 so there is no change.

If input 3 then goes inactive, then the lowest active input is still 1 so there is no change,

If input 1 then goes inactive, then the lowest active input is 2 so the system will go to service/velocity

If input 2 then goes inactive, then there are no active inputs and the system returns to service/torque.

Mode 23: Change algebraic sign of the measured analog input voltage

This mode can either change the algebraic sign of the measured analog input voltage, or zero the value using a digital input. Since the analog input voltage is used to generate command values in `DRV.CMDSOURCE=3` (analog command source), Mode 23 can also be used to change the direction of movement or stop motion using a digital input in `DRV.CMDSOURCE=3`.

This mode is valid for all opmodes and command source 3 (analog).

The value of `DINx.PARAM` defines the value of a multiplication factor for the measured analog voltage. The low-byte of the `DINx.PARAM` value determines the value of this factor, which is multiplied by the measured analog input voltage upon a rising edge on the associated digital input. The high-byte determines the factor upon a falling edge as follows:

$$\text{DINx.PARAM} = 0\text{xFFRR} \text{ (F=Falling edge; R = Rising edge)}$$

The following values are used in this mode:

| Value | Description |
|-------|--|
| 0x00 | Zero the measured analog voltage. |
| 0x01 | Multiply the analog voltage with 1. |
| 0x02 | Multiply the analog voltage with -1. |
| 0x03 | Zero the measured analog voltage plus trigger in addition a software enable command. |
| 0x04 | Multiply the analog voltage with 1 plus trigger in addition a software enable command. |
| 0x05 | Multiply the analog voltage with -1 plus trigger in addition a software enable command. |
| 0x06 | Zero the measured analog voltage plus trigger in addition a software disable command. |
| 0x07 | Multiply the analog voltage with 1 plus trigger in addition a software disable command. |
| 0x08 | Multiply the analog voltage with -1 plus trigger in addition a software disable command. |

Example 1

$$\text{DINx.PARAM} = 513 = 0\text{x0201}$$

The measured analog input voltage is multiplied with a factor of 1 upon a rising edge on the associated digital input.

The measured analog input voltage is multiplied with a factor of -1 upon a falling edge on the associated digital input.

Example 2

$$\text{DINx.PARAM} = 256 = 0\text{x0100}$$

The measured analog input voltage is multiplied with a factor of 0 upon a rising edge on the associated digital input.

The measured analog input voltage is multiplied with a factor of 1 upon a falling edge on the associated digital input.

Example 3

$$\text{DINx.PARAM} = 1540 = 0\text{x0604}$$

The measured analog input voltage is multiplied with a factor of 1 upon a rising edge on the associated digital input. Additionally, the rising edge on the associated input triggers a software enable command, similar to the `DRV.EN` command.

The measured analog input voltage is multiplied with a factor of 0 upon a falling edge on the associated digital input. Additionally, the falling edge on the associated input triggers a software disable command, similar to the `DRV.DIS` command.

10.1.4 Digital Outputs

Digital inputs can be set in different modes based on the desired function. These functions are outlined below.

NOTE

If an output is overloaded (> 100 mA), then the output will turn off (with no indication in Work-Bench) and remain off until one of the following occurs:

- The power supply driving the output is removed.
- The output is turned off from the firmware.
- The 24V supply to the AKD is power cycled.

If the overload condition still exists, the output will not turn on.

Mode 0-User (Default = 0): The output state is decided by the user or fieldbus. This mode is valid for all opmodes and command source combinations.

Mode 1-Mains Ready: The output mode produces a high signal if the drive DC bus voltage is higher than the under voltage error level and lower than the over voltage error level. This mode is valid for all opmodes and command source combinations.

Mode 2-Software Limit: This output turns on when the software limit positions are reached. This output produces a high signal if a software limit is reached by traveling in the direction of that software limit. Software limits are set in the **Limits** view. In the **Limits** view, Position 0 is the position limit for negative travel, while Position 1 is the limit for positive travel.

This mode is valid for all opmodes and command source combinations.

Mode 3-Move Complete: When a motion task has completed its move and the trajectory reaches zero, the move is considered complete and the output will activate.

This mode is valid for opmode 2 (position) and command source 0 (service) only.

Mode 4-In Position: This output mode produces a high signal when the absolute value of the position error is less than the parameter entered in the extra parameter field.

This mode is valid for all opmode 2 (position) and all command sources.

Mode 5-Position Greater than X: When the position is greater than the parameter entered in the extra parameter field, the output will activate.

This mode is valid for all opmodes and command source combinations.

Mode 6-Position Less than X: When the position is less than the parameter entered in the extra parameter field, the output will activate.

This mode is valid for all opmodes and command source combinations.

Mode 7-Warning: This output will activate when the drive experiences a warning, such as positive or negative limit switch input triggered.

This mode is valid for all opmodes and command source combinations.

Mode 8-Enable: If you need an output to indicate that the drive is enabled, use this output mode.

This mode is valid for all opmodes and command source combinations.

Mode 9: Reserved.

Mode 10-Motor Brake: The output mode produces a high signal if a brake is released (this is when the power is applied to the brake and the motor is free to spin). The output mode produces a low if a brake is applied (this is when power is removed from the brake and the brake is set).

This mode is valid for all opmodes and command source combinations.

Mode 11-Drive Faults: The output mode produces a high signal if the drive has a fault.

This mode is valid for all opmodes and command source combinations.

Mode 12-Absolute velocity greater than x: The output mode produces a high signal when the absolute value of the velocity is greater than a variable x. Use DOUTx.PARAM to set x.

This mode is valid for all opmodes and command source combinations.

Mode 13-Absolute velocity less than x: The output mode produces a high signal when the absolute value of the velocity is less than a variable x.

Use DOUTx.PARAM to set x.

This mode is valid for all opmodes and command source combinations.

Mode 14-Homing complete: The output mode produces a high signal when the homing process is completed.

This mode is valid for opmode 2 (position) and command source 0 (service) only.

Mode 15- PLS.STATE bits OR connected: The output mode produces a high signal if at least one of the PLS.STATE bits is high (the PLS is active) and if the corresponding bit in the DOUTx.PARAM parameter also has been set to high. The DOUTx.PARAM command connects the PLS.STATE bits to the digital output itself and thus acts as an enable mask.

This mode is valid for all opmodes and command source combinations.

Example

```
|<- Bit 7 to 0 ->|
```

```
DOUT1.PARAM = 23 = 0b 0 0 0 1 0 1 1 1 (Binary code)
```

The digital output 1 is active when bit 0 or bit 1 or bit 2 or bit 4 of PLS.STATE is high. All other bits within PLS.STATE are not considered by the digital output mode due to the DOUT1.PARAM setting. Do not use decimal places for the DOUTx.PARAM parameter for this particular digital output mode.

Summary of Opmode and Command Source Dependencies

| DINx.MODE | Mode Description | Opmode | Command Source |
|------------|--------------------------------|-----------------|----------------------|
| 0 | Off | all | all |
| 1 | Fault Reset | all | all |
| 2 | Start Motion Task | 2-Position | 0-Service |
| 3 | Motion Task Select Bit | 2-Position | 0-Service |
| 4 | Motion Task Start Selected | 2-Position | 0-Service |
| 5 | Start Home | 2-Position | 0-Service |
| 6 | Start Jog | 1 2-Position | 0-Service |
| 8 | Zero Latch | all | all |
| 9 | Command Buffer | all | all |
| 10 | Control Fault Relay | all | all |
| 11 | Home Reference | 2-Position | 0-Service |
| 13 | Controlled Stop | all | all |
| 15 | Quick Stop | all | 0-Service |
| 16 | Activate Electronic Gearing | 2-Position | 2-Electronic Gearing |
| 17 | Electronic Gear Position Shift | 2-Position | 2-Electronic Gearing |
| 18 | Positive Limit Switch | all | all |
| 19 | Negative Limit Switch | all | all |
| 20 | Brake Release | all | all |
| 21 | Current Limitation | all | all |
| 22 | Switch CmdSource/Op-mode | all | all |
| 23 | Analog In Sign Control | all | 3-Analog |
| DOUTx.MODE | Mode Description | Opmode | Command Source |
| 0 | User- (Default=0) | all | all |

| DINx.MODE | Mode Description | Opmode | Command Source |
|-----------|---------------------------|------------|----------------|
| 1 | Mains Ready | all | all |
| 2 | Software Limit | all | all |
| 3 | Move Complete | 2-Position | 0-Service |
| 4 | Inposition | 2-Position | all |
| 5 | Position > x | all | all |
| 6 | Position < x | all | all |
| 7 | Warning | all | all |
| 8 | Enable | all | all |
| 10 | Motor Brake | all | all |
| 11 | Drive Fault | all | all |
| 12 | Absolute Velocity > x | all | all |
| 13 | Absolute Velocity < x | all | all |
| 14 | Homing Complete | 2-Position | 0-Service |
| 15 | Programmable Limit Switch | all | all |

10.2 Command Buffer

10.2.1 Overview

The Command Buffer input mode (10.1 Digital Inputs and Outputs) allows you to change values for parameters using a digital input.

The drive has four available buffers. A digital input configured for command buffer mode is linked to one command buffer set. This is determined by the user (see arrow 1). In this case, command buffer 1 is used.



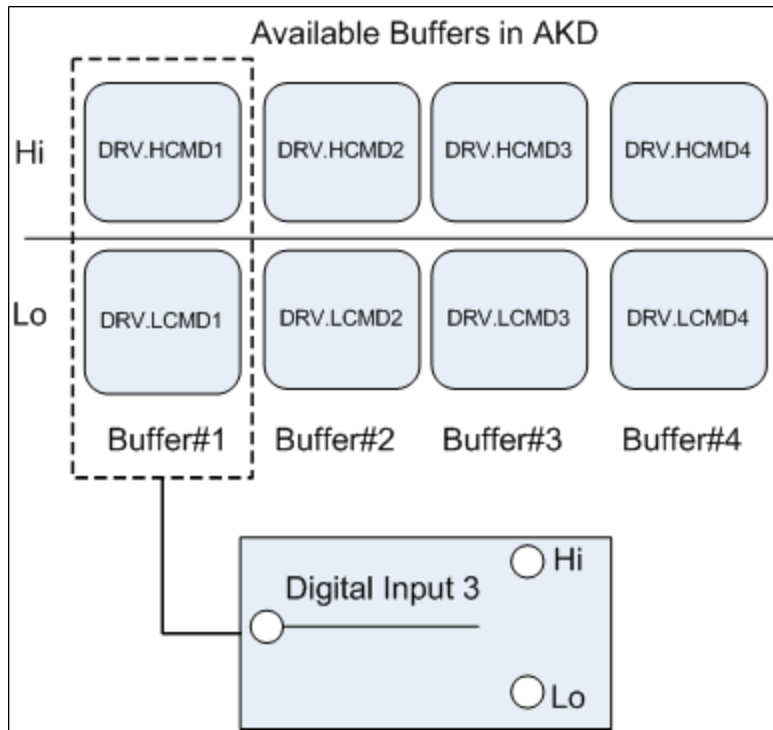
Digital Inputs and Outputs

This page shows current state of each of the I/O pins and allows to select the function each pin performs.

General Purpose Digital Inputs

| | State | Beep: | Mode: | Param: | Polarity: |
|-------------------------------|----------------------------------|--------------------------|--------------------|--|---|
| DIN 1- High Speed(X7 pin 10): | <input type="radio"/> | <input type="checkbox"/> | 0 - Off | 0.000 | <input checked="" type="checkbox"/> Active High |
| DIN 2- High Speed(X7 pin 9): | <input type="radio"/> | <input type="checkbox"/> | 0 - Off | 0.000 | <input checked="" type="checkbox"/> Active High |
| DIN 3 (X7 pin 4): | <input checked="" type="radio"/> | <input type="checkbox"/> | 9 - Command Buffer | Buffer # 1 <input type="button" value="Edit"/> | <input checked="" type="checkbox"/> Active High |
| DIN 4 (X7 pin 3): | <input type="radio"/> | <input type="checkbox"/> | 0 - Off | 0.000 | <input checked="" type="checkbox"/> Active High |
| DIN 5 (X8 pin 6): | <input type="radio"/> | <input type="checkbox"/> | 0 - Off | 0.000 | <input checked="" type="checkbox"/> Active High |
| DIN 6 (X8 pin 5): | <input type="radio"/> | <input type="checkbox"/> | 0 - Off | 0.000 | <input checked="" type="checkbox"/> Active High |
| DIN 7 (X7 pin 2): | <input type="radio"/> | <input type="checkbox"/> | 0 - Off | 1.000 | <input checked="" type="checkbox"/> Active High |

The graphic below explains the architecture of the buffers.



10.2.2 Editing the Command Buffers

By default, the buffers are empty. Each side of the buffer can contain 128 characters maximum (parameter and value included). You can use the **Command Buffer Editor** to enter the sequence of commands to the digital input buffer. You can enter the sequence of commands into Low command buffer (DIN.LCMDx) or High command buffer (DIN.HCMDx) and save these settings to the drive.

To edit the command buffers, open the **Digital I/O** view and select **Digital input mode** as **9 – Command Buffer**.

| General Purpose Digital Inputs | | | | | | |
|--------------------------------|-----------------------|--------------------------|--------------------|--------|-------------------------------------|---|
| | State | Beep: | Mode: | Param: | Polarity: | |
| DIN 1- High Speed(X7 pin 10): | <input type="radio"/> | <input type="checkbox"/> | 9 - Command Buffer | 1 | <input type="button" value="Edit"/> | <input checked="" type="checkbox"/> Active High |

The **Param** box lists the available command buffers. Select desired command buffer number for the DIN. This number sets to the DINx.PARAM keyword. When you mouse over the **Param** box, the Tooltip displays the current content of the High command buffer and Low command buffer in the drive.

To edit the selected command buffer, click **Edit** to display the command buffer editor screen.



The command buffer editor screen has following properties:

| Button or Dialog Box | Description |
|------------------------------|---|
| Command Buffer Number | The identification number of the command buffer (1, 2, 3, 4). |
| High Command Buffer | Adds sequence of commands to the High command buffer parameter. Contents are saved to the keyword DIN.HCMDx. A maximum of 128 characters can be set to drive along with the separator “;”. Commands must be entered line by line and when saving to the drive each commands will be formed into single line separated by “;”. |
| Low Command Buffer | Adds sequence of commands to the Low command buffer parameter. Contents are saved to the keyword DIN.LCMDx. A maximum of 128 characters can be set to drive along with the separator “;”. Commands must be entered line by line and when saving to the drive each commands will be formed into single line separated by “;”. |
| Ok | Saves the sequence of commands to the drive. |
| Close | Closes the screen and returns to Digital I/O view. If contents are not saved to drive before closing the screen, confirmation message “The commands have been modified and not save to drive. Do you want close without saving?” is displayed. |

Commands and parameters are entered on separate lines with a space between the parameter and the value. A semicolon separator is not necessary in the editor, but it is required if the buffers are edited inside the terminal window.

10.2.3 Behavior of the Command Buffer

Digital inputs have either a high or a low state. The contents of the buffer are executed at the rising edge of the state change. The contents of the buffer are also loaded at drive power up according to the starting state of the digital input. When the command buffer is initially configured, the buffer is not executed until the first digital input state change is detected.

Tip: Once you have the buffer configured and tested, put the digital input in the most common state that it will be in at start up. Save the parameters to the drive. This will synchronize the NVRAM with the buffer, so at start up, values will not have to be changed.

10.2.4 Delays for the Buffer

There is a delay command that can be used to delay the execution of a command or parameter change (DRV.CMDDELAY). The value can be from 0 ms to 5000 ms.

The following commands may require a delay before the next command in the buffer can be executed:

DRV.EN (100 ms min)

DRV.DIS (50 ms min)

Example:

```
DRV.EN
DRV.CMDDELAY 100
MT.MOVE
```

The command buffer does not send back warnings when a parameter is invalid or out of range, so make sure the syntax is correct and that the digital input changes during legal drive states for the commands given.

10.3 Digital Inputs (X7/X8)

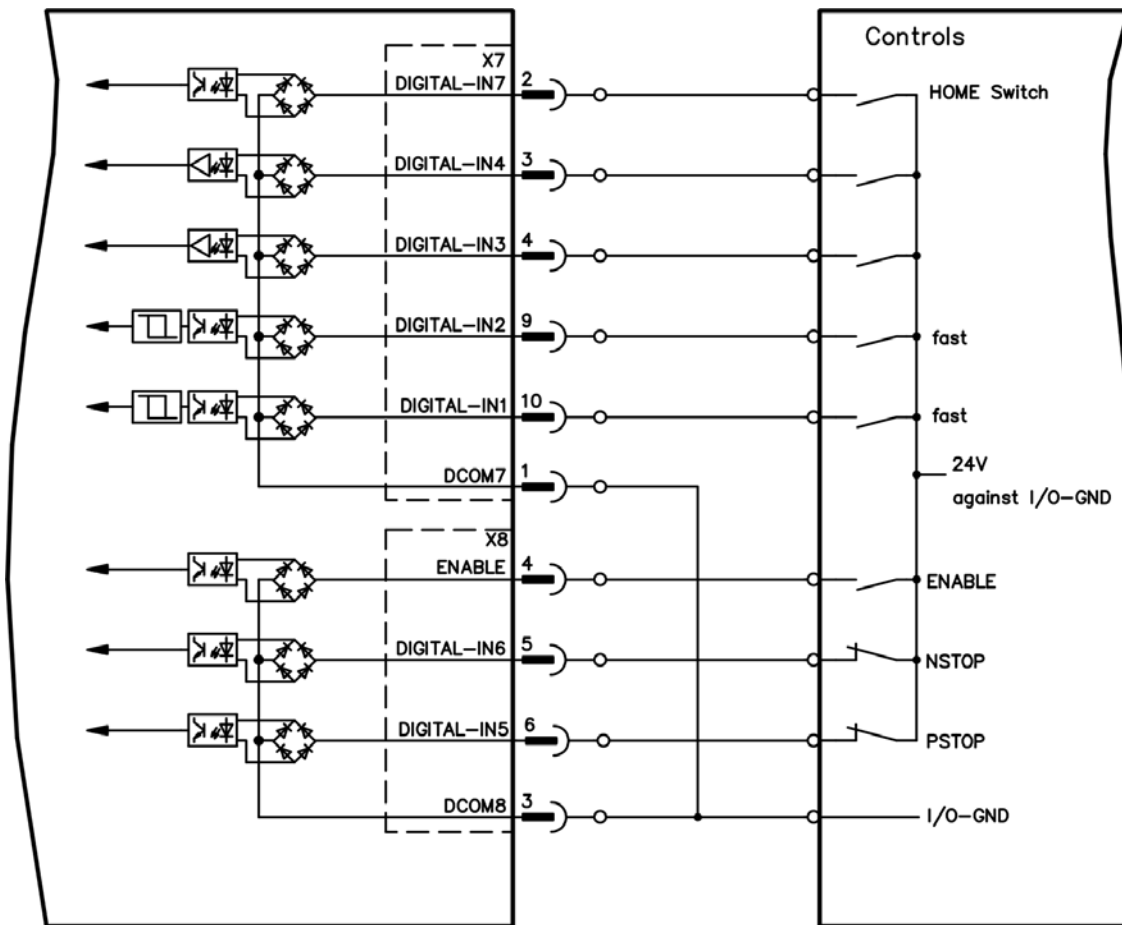
The drive provides 8 digital inputs (=> p. 273). These can be used to initiate pre-programmed functions that are stored in the drive. A list of these pre-programmed functions is included in the WorkBench. Digital Input 8 is not programmable but is fixed to the ENABLE function.

If an input is programmed, it must be saved to the drive.

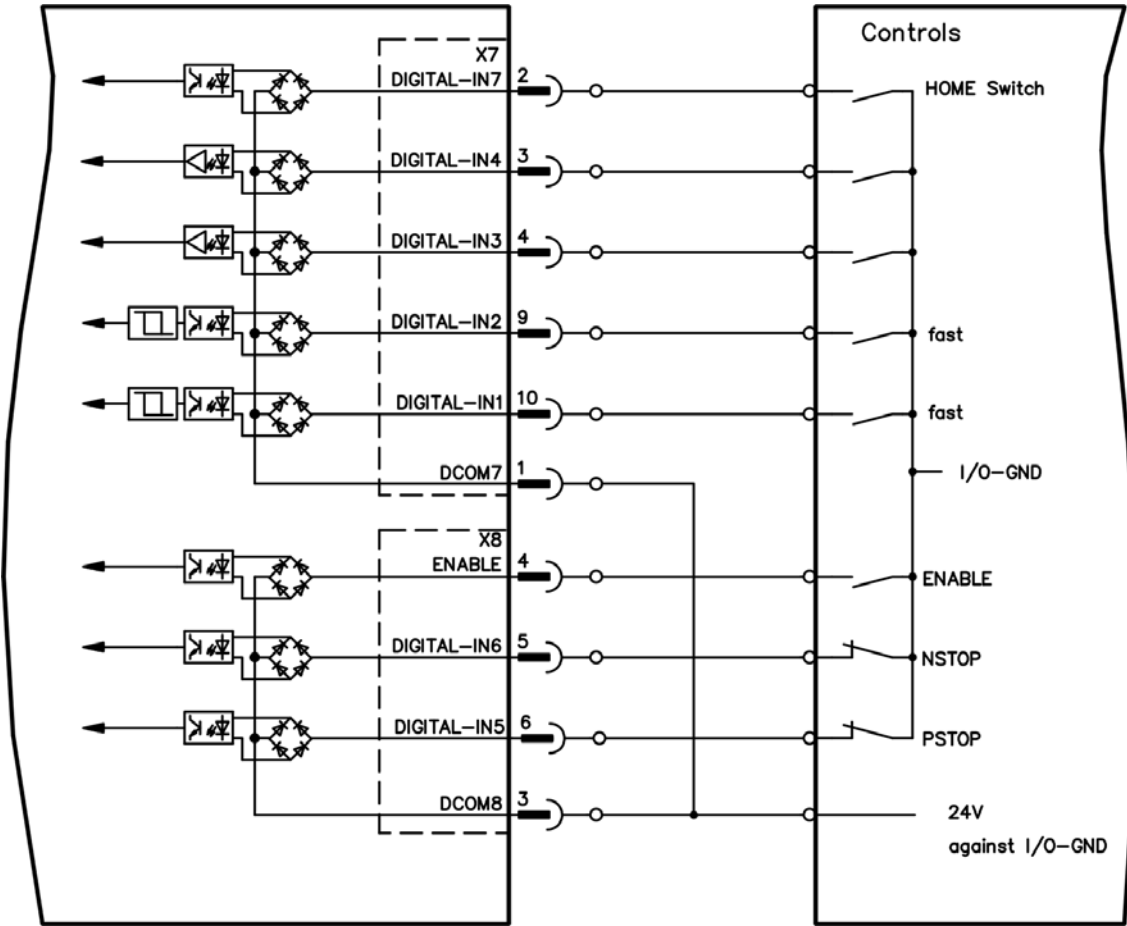
| | |
|-------------|--|
| NOTE | Depending on the selected function the inputs are high or low active. |
|-------------|--|

The inputs can be used with switched +24 V (source type) or switched GND (sink type). See diagrams below for typical examples of digital input wiring.

Digital Input Wiring Diagram (Source type connection, example)



Digital Input Wiring Diagram (Sink type connection, example)



10.3.1 Digital Inputs 1 and 2

These inputs (X7/9 and X7/10) are particularly fast and are therefore suitable for latch functions, for example. They can also be used as 24 V inputs for electronic gearing (=> p. 1)

Technical characteristics

- Floating, reference common line is DCOM7
- Sink or Source type sensors possible
- High: 3.5 to 30 V/2 to 15 mA , Low: -2 to +2 V/<15 mA
- Update rate: Hardware 2 μ s

10.3.2 Digital Inputs 3 to 7

These inputs are programmable with the setup software. By default, all inputs are not programmed (off). For more information refer to the setup software.

Technical characteristics

Choose the function you require in WorkBench.

- Floating, reference common line is DCOM7 or DCOM8
- Sink or Source type sensors possible
- High: 3.5 to 30 V/2 to 15 mA , Low: -2 to +2 V/<15 mA
- Update rate: Software 250 μ s

10.3.3 Digital Input 8 (ENABLE)

Digital Input 8 (terminal X8/4) is set to Enable function.

- Floating, reference common line is DCOM8
- Sink or Source type wiring is possible
- High: 15 to 30 V/2 to 15 mA , Low: -3 to 5 V/<15 mA
- Update rate: direct connection to hardware (FPGA)

The output stage of the drive is enabled by applying the ENABLE signal (Terminal X8/4, active high). Enable is possible only if input STO has a 24 V signal (see page 1). In the disabled state (low signal) the connected motor has no torque.

A software enable by means of the setup software is also required (AND link), although this can also be permanently enabled with WorkBench.

10.4 Analog Input

10.4.1 Analog Input

If the drive command source is set to analog, then the analog input to the drive supplies the current or velocity command to the control loops of the drive. The default analog input screen displays a summary block diagram of the analog input. You can adjust the analog input settings from this view as follows:

| Button or Dialog Box | Description | Parameter |
|-----------------------|--|--|
| Offset | The offset adds a bias to the analog input command. This offset is commonly used to remove any biases that may be present on the analog input signal. | AIN.OFFSET |
| Input Voltage | The value of the analog input after the offset, deadband, and low pass filters. | AIN.VALUE |
| Scale | If the opmode is current mode, then this value is the amount of current that will be commanded for each volt on the analog input. If the opmode is velocity mode, then this value is the velocity that will be commanded for each volt on the analog input. | AIN.ISCALE AIN.VSCALE AIN.PSCALE |
| Torque Command | The current or velocity command that is sent to the control loops. | IL.CMD PL.CMDor VL.CMD |

Click the **More** button to access a detailed view of the analog input. You can adjust additional analog input settings from this view as follows:

| Button or Dialog Box | Description | Parameter |
|------------------------|--|--------------|
| Low Pass Filter | The break point frequency for the low pass filter. | AIN.CUTOFF |
| Deadband | The threshold for the deadband. This parameter is commonly used to reduce noise while the drive is stationary. | AIN.DEADBAND |

Both analog input views provide a link to the encoder emulation output setup; see 8.4.1 Encoder Emulation for more details on this feature.

10.4.2 Related Parameters and Commands

AIN Parameters

10.5 Analog Output

The drive has one analog output. You can either directly control the voltage output or select a different mode to output different signals. Analog output is configured through the X8 connector; for detailed information this connector, see [Connection Diagrams, Analog Output](#).

| Button or Dialog Box | Description | Parameter |
|----------------------------|--|-----------------------------|
| Analog Output Mode | Select which internal signal is output by the analog output. | AOUT.MODE |
| Analog Output Value (user) | Enter the analog output value (when AOUT.MODE = 0, analog output signal is determined by the user) | AOUT.VALUEU |

| Button or Dialog Box | Description | Parameter |
|------------------------------|---|-----------------------------|
| Analog Output Value | Displays the voltage that this analog output generates. | AOUT.VALUE |
| Analog Velocity Scale Factor | Scales the analog output (AOUT.VALUE) for AOUT.MODE = 1, 2, or 3. | AOUT.VSCALE |

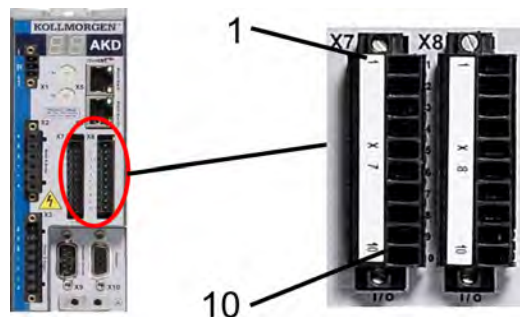
The analog output modes consist of the following:

| AOUTx.MODE | Description |
|------------|---|
| 0 | User variable. The analog output signal is determined by the user (using AOUT.VALUEU). |
| 1 | Actual velocity. The analog signal describes the current velocity value (VL.FB). |
| 2 | Velocity error. The analog signal describes the velocity error value. |
| 3 | Velocity command. The analog signal describes the velocity command value. |
| 4 | Actual current. The analog signal describes the actual current value. |
| 5 | Current command. The analog signal describes the current command value. |
| 6 | Actual position. The analog signal describes the current position value. |
| 7 | Position error. The analog signal describes the position error value. |
| 8 | Triangle wave. The analog signal is a triangle wave (sawtooth pattern). |
| 9 | Debug mode. In this mode the user can define a drive variable to monitor via the analog output (AOUT.VALUEU). |
| 10 | Unfiltered Velocity (VL.FBUNFILTERED) |
| 11 | Filtered Velocity - 10Hz Lowpass (VL.FBFILTER) |

Analog output return

10.6 I/O Connection

All standard digital and analog I/O signals are connected to X7 and X8.



10.6.1 I/O Connectors (X7 and X8)

| Conn. | Pin | Signal | Abbreviation | Function | Specials |
|-------|-----|--------------------|--------------------|--|------------------|
| X7 | 1 | Digital Common X7 | DCOM7 | Common line for X7 pins 2, 3, 4, 9, 10 | - |
| X7 | 2 | Digital Input 7 | DIGITAL-IN 7 | Programmable | - |
| X7 | 3 | Digital Input 4 | DIGITAL-IN 4 | Programmable | - |
| X7 | 4 | Digital Input 3 | DIGITAL-IN 3 | Programmable | - |
| X7 | 5 | Digital Output 2- | DIGITAL-OUT2- | Programmable | - |
| X7 | 6 | Digital Output 2+ | DIGITAL-OUT2+ | Programmable | - |
| X7 | 7 | Digital Output 1- | DIGITAL-OUT1- | Programmable | - |
| X7 | 8 | Digital Output 1+ | DIGITAL-OUT1+ | Programmable | - |
| X7 | 9 | Digital Input 2 | DIGITAL-IN 2 | Programmable | high speed |
| X7 | 10 | Digital Input 1 | DIGITAL-IN 1 | Programmable | high speed |
| X8 | 1 | Fault Relay Output | Fault Relay Output | Fault Relay Output | - |
| X8 | 2 | Fault Relay Output | Fault Relay Output | Fault Relay Output | - |
| X8 | 3 | Digital Common X8 | DCOM8 | Common line for X8 pins 4, 5, 6 | - |
| X8 | 4 | Digital Input 8 | DIGITAL-IN 8 | Output stage enable | not programmable |
| X8 | 5 | Digital Input 6 | DIGITAL-IN 6 | Programmable | - |
| X8 | 6 | Digital Input 5 | DIGITAL-IN 5 | Programmable | - |
| X8 | 7 | Analog Ground | AGND | Analog GND | - |
| X8 | 8 | Analog Output + | Analog-Out | Actual velocity voltage | - |
| X8 | 9 | Analog Input - | Analog-In- | Velocity set point | - |
| X8 | 10 | Analog Input + | Analog-In+ | | - |

Digital common lines for X7 and X8 are not common to each other.

The DCOMx lines should be connected to the 0V of the I/O supply when using sensors of type "Source" with digital inputs.

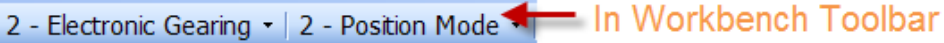
The DCOMx lines should be connected to the 24V of the I/O supply when using sensors of type "Sink" with digital inputs.

10.7 Electronic Gearing

10.7.1 Overview

Electronic gearing is the act of sending a digital pulse position command to the AKD drive. The X9 connector is used for controlling the drive through an A/B type signal, pulse and direction (also called step and direction), or up/down command. A common application for electronic gearing is using servos with a stepper controller or daisy chaining multiple AKD drives from one master drive as slave drives.

In order to command an AKD using electronic gearing, the command source (DRV.CMDSOURCE) must be set to **2-Electronic Gearing** and the opmode (DRV.OPMODE) must be set to **2-Position Mode**.



The input modes of the X9 connector are used to set up the AKD for electronic gearing.

Electronic Gearing ? [Learn](#)

Electronic Gearing allows the drive to follow a position being supplied to the drive.

Resolution:

Input Type:

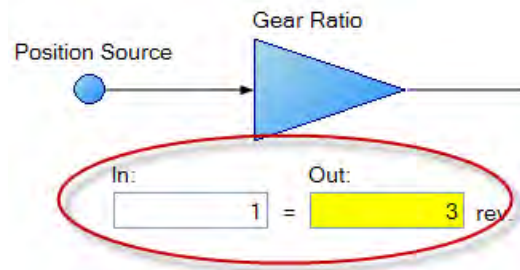
0 - EEO connector is not operative
 0 - EEO connector is not operative
 1 - Output - With once per rev index pulse
 2 - Output - With absolute index pulse
 3 - Input - A/B signals
 4 - Input - Step and direction signals
 5 - Inputs - up-down signals

Position Source → ▶ Gear Ratio → Limits

3,229,110,765

Only input modes 3,4, and 5 pertain to electronic gearing input.

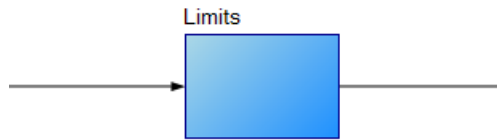
Resolution is the post-quadrature value of the counts/revolution of the input. Additionally, a gear ratio can be applied to affect the output ratio of the motor.



The position command (DRV.HANDWHEEL) reads the EEO value, where 4,294,967,296 is a full revolution of the input, then the value rolls over. Gear ratio does not affect the EEO value. If the output is set to 3 output revs per input rev, there will be 4,294,967,296 counts per 3 revolutions of the motor.

10.7.2 Limits

Electronic Gearing has independent limits, as shown below:



Maximum Velocity: rpm
 Maximum Acceleration: rpm/s
 Maximum Deceleration: rpm/s

These limits (GEAR.ACCMAX, GEAR.DECMAX, GEAR.VELMAX) are applied only during gearing mode and the units are consistent with speed and acceleration of the output motor. All other limits in the drive are active along with gearing limits.

If the master is already moving when entering electronic gearing mode, velocity or position can be matched (GEAR.MODE):

The Gearing type defines how gearing starts if the master is already moving—

- Velocity Matching
 Position Matching

In **Velocity Matching**, the motor will ramp up to the same velocity with no concern over lost steps during the acceleration period.

In **Position Matching**, the motor will match the position command from the switchover point by speeding up to recover the lost steps during the acceleration period.

More information on each type of input mode is found here:

- 3 – AquadB (see Command encoder signal connection)
- 4 – Pulse/Direction input (see Pulse / Direction signal connection)
- 5- Up/Down input (see Up / Down signal connection)

10.7.3 Determining Maximum Cable Length

When you use an external incremental encoder as an input to X9, you must determine the maximum allowable cable length.

NOTE This information is only applicable when using an external encoder as either a secondary feedback input or a gearing command (DRV.EMUEMODE3). Not applicable for any other X9 mode or when using two AKDs in a master/slave system.

The X9 port has a 5V output used to supply power to an external incremental encoder.

The maximum cable length depends on the current draw of the external encoder and the cable type connecting the X9 port. The following example can be used as a guide to calculate the maximum cable length for your application.

X9 port characteristics:

Nominal Supply Voltage: 5 V
 Tolerance: 5%
 Minimum Supply Voltage: 4.75 V
 Maximum current: 0.25 A
 Permitted wire gauge: 20-28 AWB (Typical for D9 connector)

Sample Application Hardware:

Example external encoder: Hengstler RI-36H (RS-422 encoder) used with X9 port.
 Encoder Nominal Supply Voltage: 5V (+/- 10%)

Minimum Supply Voltage: 4.5 V calculated based on tolerance above

Maximum required encoder supply current: 50 mA

Example cable:

Lapp Li2YCY - 24AWG (0.22 mm²)

Loop resistance: 0.186 Ohms/m

Sample Calculations:

Maximum Permissible voltage cable drop = 0.25 V

= (Minimum Supply Voltage from AKD) 4.75 V – (Minimum supply voltage of RI-36H encoder) 4.5V

Maximum permissible resistance of cable run to X9 = 5 Ohms

= (Max voltage cable drop) 0.25V ÷ (Maximum encoder current) 0.05A

Maximum permissible cable length for example application = 26.9 m

= (Max cable resistance) 5 Ohms ÷ 0.186 Ohms/m

10.7.4 Related Parameters:

GEAR Parameters

DRV.CMDSOURCE

DRV.EMUEMODE

DRV.EMUERES

DRV.HANDWHEEL

DRV.OPMODE

This screen allows you view and modify the various drive limits.

| Button or Dialog Box | Description | Parameter |
|----------------------------|---|---------------|
| Current Limits | | |
| Positive Peak Current | The maximum positive current allowed. | IL.DLIMITP |
| Negative Peak Current | The maximum negative current allowed. | IL.DLIMITN |
| Velocity Limits | | |
| Positive Speed Limit | The maximum speed allowed in the positive direction. | VL.LIMITP |
| Negative Speed Limit | The maximum speed allowed in the negative direction. | VL.LIMITN |
| Over Speed Limit | The threshold velocity for an over speed fault. | VL.THRESH |
| Position Limits | | |
| Maximum Following Error | The maximum position error. If the position error PL.ERR is larger than PL.ERRFTHRESH, then the drive generates a fault | PL.ERRFTHRESH |
| Position Limit 0 | The minimum position the drive can reach before generating a negative software position fault. | SWLS.LIMIT0 |
| Position Limit 1 | The maximum position the drive can reach before generating a positive software position fault. | SWLS.LIMIT1 |
| Acceleration Limits | | |
| Acceleration | The acceleration ramp used to profile some types of motion. | DRV.ACC |
| Deceleration | The deceleration ramp used to profile some types of motion. | DRV.DEC |
| Motor Limits | Motor limits are set through the Motor Foldback Screen (see 8.8 Foldback | |

10.7.5 Limits

The limits screen covers most of the basic system limits, including current, velocity, and position.

- **Current Limits:** The current limits are set based on the drive ratings. You can change these limits to be lower than the default values for the drive, however, this may affect the expected performance of your application.
- **Velocity Limits:** The velocity limits are set based on the motor ratings. You can modify these settings above the ratings of the motor if the application requires some overhead, but be aware the motor has mechanical limitations and may be damaged if run above those limits. It is best to leave these at the default ratings for the motor selected.
- **Position Limits:** The position limits can be set based on your specific machine application requirements. The Maximum Position Error can be set to cause a fault when the position error exceeds the value you insert here. Position Limit 0 is tied to the clockwise (positive) motor direction. As the motor reaches the position entered, the motor will stop and display a warning n107. Position Limit 1 is tied to the counter-clockwise (negative) motor direction. As the motor reaches the position entered, the motor will stop and display a warning n108.
- **Acceleration Limits:** This field allows you to raise accelerations to give the system crisp moves. These limits default to a low value, so you may wish to change these after the mechanics and other sections of your system are defined.

10.8 Programmable Limit Switch

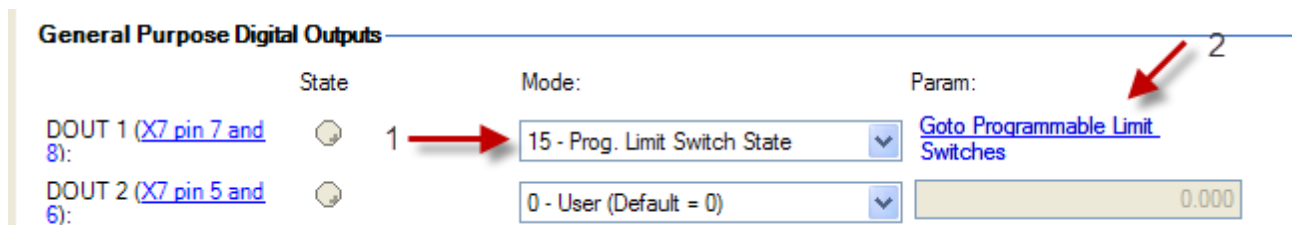
10.8.1 Overview

Programmable Limit Switches (PLSs) are used to turn on and off drive digital outputs based on the drive's position. Multiple positions can be combined to affect the state of an output when PLSs are combined.

10.8.2 Using Programmable Limit Switches

To use PLSs, you must first configure a digital output as follows:

1. Click on the Digital I/O icon in the tree view.
2. Set the output of your choice to mode fifteen (see 1 below). In this example, digital output 1 is used.
3. Now that the digital output mode is set for PLS, you can click on the **Goto Programmable Limit Switch** link (see 2 below) to open the PLS screen (this screen is also shown in the WorkBench tree view).



The PLS screen is used to establish the positions for the output(s) to turn on.

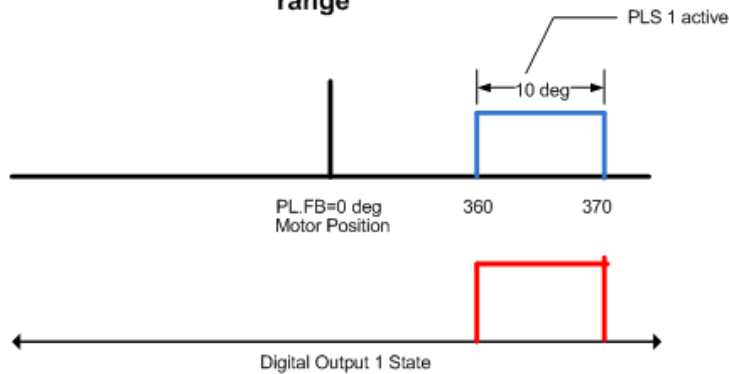
| PLS | Enabled | State | Mode | Position | Units | Width/Time | Reset |
|------|-------------------------------------|-------|--------------|-------------|------------|------------|-------|
| PLS1 | <input checked="" type="checkbox"/> | ⊕ | 0-Continuous | 360.000 deg | 0-Position | 10.000 deg | Reset |
| PLS2 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS3 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS4 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS5 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS6 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS7 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS8 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |

The PLS configuration section of the screen sets the mode and limits of each of the eight PLSs. The PLS is ignored unless it is enabled (see arrow). In the screen example, PLS1 is set for continuous operation in position mode. Every time the 360 degree position (PL.FB) is crossed in either direction, the output will turn on for 10 degrees of motor movement.



The final step is to configure the OR gate for the PLSs on which output is triggered. The gate appears for setup in the screen when a digital output is configured in Mode 15 – Prog Limit Switch State. Since only PLS1 is configured, select PLS 1 (see arrow above)

PLS Example 1 – Turning on a digital output for a position range



Parameters
 DOUT1.MODE = 15- Prog Limit Switch State
 DOUT1.PARAM=1 (PLS1 in OR Gate)
 PLS.EN = 1 = PLS1 enabled
 PLS.P1 = 360 deg
 PLS.WIDTH1 = 10 deg
 PLS.MODE = 0-Continuous
 PLS.UNITS = 0 (position)

To setup an output with multiple turn-on points, configure and enable more PLS's and include them in the OR Gate.

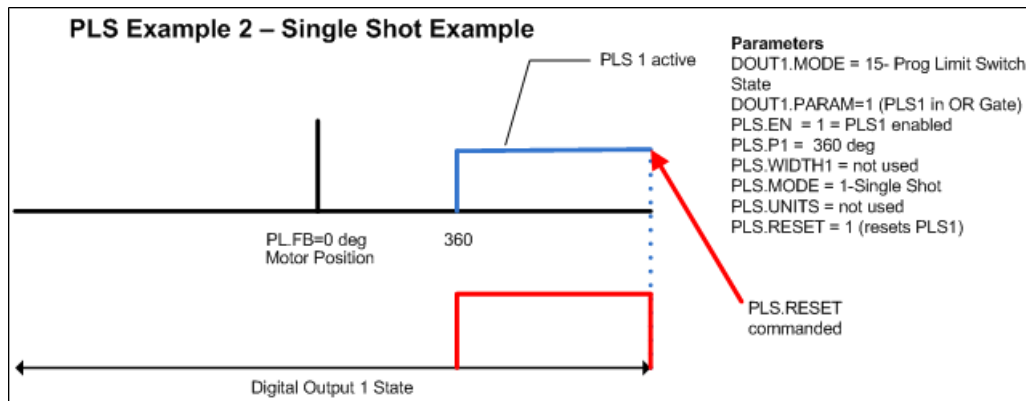
| PLS | Enabled | State | Mode | Position | Units | Width/Time | Reset |
|------|-------------------------------------|-------|--------------|---------------|------------|------------|-------|
| PLS1 | <input checked="" type="checkbox"/> | ⊕ | 0-Continuous | 360.000 deg | 0-Position | 10.000 deg | Reset |
| PLS2 | <input checked="" type="checkbox"/> | ⊕ | 0-Continuous | 1,000.000 deg | 0-Position | 10.000 deg | Reset |
| PLS3 | <input checked="" type="checkbox"/> | ⊕ | 0-Continuous | 2,000.000 deg | 0-Position | 10.000 deg | Reset |
| PLS4 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS5 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS6 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS7 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |
| PLS8 | <input type="checkbox"/> | ⊕ | 0-Continuous | 0.000 deg | 0-Position | 0.000 deg | Reset |

Current Position: Position Feedback: 121.549 deg

10.8.3 Single Shot Mode

Single shot mode is a special mode of PLS. Single shot mode (see 1 below) turns on the output until it is reset (see 2 below). Normal operation of this mode usually depends on a machine controller to reset the PLS using the fieldbus object for PLS.RESET.

Single Shot Example:



10.8.4 Related Parameters and Commands

PLS Parameters and Commands

10.1 Digital Inputs and Outputs

10.9 Enable/Disable

10.9.1 Enable Modes

The AKD offers several options for hardware and software enables, as well as safe torque off (STO) to cover a variety of conditions.

10.9.1.1 Hardware Enable Mode

The AKD has two methods hardware enable methods. These methods are controlled by DRV.HWENMODE. Mode 0 allows for the drive to enable and clear faults on the rising edge of the hardware enable input. Mode 1 will NOT clear any faults on the rising edge of the hardware enable input, allowing you to review any current faults and manually clearing.

10.9.1.2 Software Enable Default

In addition, the Software Enable has two methods for enabling the AKD. These are controlled by DRV.E-NDEFAULT. Default 0 leaves the software in the disabled state upon start-up. Default 1 enables the software upon start-up.

10.9.2 Disable Modes

Use DRV.DISMODE to select the method for stopping the drive.

Mode 0: Immediately disable drive.

With this condition, the drive will immediately disable the power stages and the driven motor will either coast to a stop or in the case of a vertical or overhung load axis, will fall abruptly. If a brake is present, the brake will be applied according to MOTOR.TBRAKEAPP. Using Digital Input mode 13, you can achieve a controlled stop as described in Mode 2.

Mode 1: Dynamic Brake to a stop.

In this situation, the drive will use the dynamic brake feature and stop motion quickly, then disable the power stage. In most cases, the driven motor will stop quickly (pending the joules available and load circumstances). In the case of an overhung or vertical load, the driven motor will attempt to stop, but then will continue to allow the load to fall if no measures have been taken to secure the load.

Note: for Modes 2 and 3, you can access the Controlled stop section to set the values of the controlled stop deceleration rate, the velocity threshold, and time in velocity threshold to deactivate the drive.

Mode 2: Controlled stop, then disable.

In this mode, a controlled stop will take place based on a variety of parameters that you set. First, the driven motor will decelerate at a controlled rate (CS.DEC) until one of two things happens. 1) The motor reaches the velocity limit set (CS.VTHRESH) for a period of time (CS.TO), or 2) The drive emergency time out is reached (DRV.DSTO). Once either of these cases is reached, the power stage will be disabled (and brake applied if present)

Mode 3: Controlled stop, then dynamic brake.

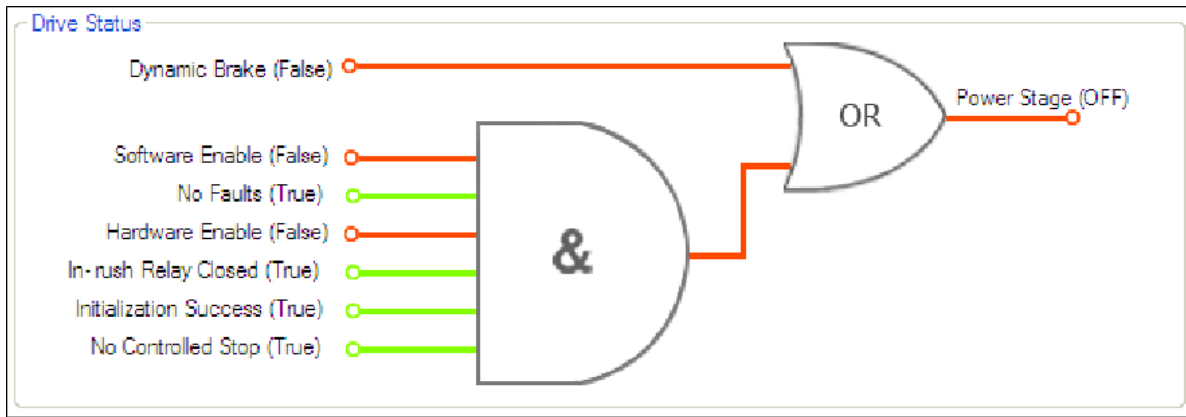
Similar to Mode 2, the motor will decelerate at a controlled rate (CS.DEC) until CS.VTHRESH is reached for a period of time (CS.TO). The drive will then dynamically brake and disable under the same conditions as described in Mode 2.

The Disable Timeout Setting determines the amount of time the drive will follow the disable mode before it deactivates the drive regardless of method chosen and alerts with an Emergency Timeout Fault.

All disable modes operate based on the type of disable command received. Any critical faults, hardware disable, or STO disable immediately turn off the power stage, and the motor will coast or free fall, depending on how the motor is secured.

10.9.3 Drive Status

The **Drive Status** area is displayed below the settings area and includes a graphical representation of drive active status with different sets of inputs. If the input or output is enabled, then it is shown in green; if the input or output is disabled, then it is shown in red. The inputs to the OR and & (AND) gates identify which conditions are true (green) or false (red) and follow the normal logic for & (AND) and OR gates. This diagram is useful in finding which input may be preventing the drive from being enabled. Click **More** to see the details of how the control stop is executed displayed both logically and graphically.



10.9.4 Controlled Stop

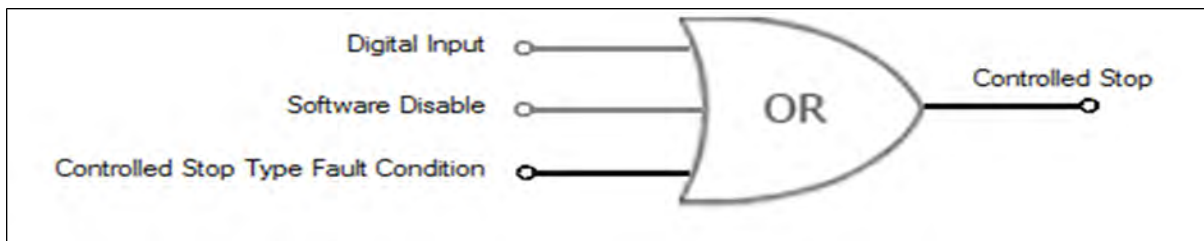
The Controlled Stop area displays values for parameters associated with controlled stop.

| Button or Dialog Box | Description | Parameter |
|-----------------------------------|--|------------|
| Velocity Threshold | Sets the velocity threshold for controlled stop process. | CS.VTHRESH |
| Velocity Threshold Timeout | Sets the velocity threshold, which is the time value for the drive velocity to be within CS.VTHRESH before the drive disables | CS.TO |
| Deceleration | Sets the deceleration value for the controlled stop process. | CS.DEC |
| Control stop Input | Displays the list of digital inputs configured to controlled stop mode separated by ','. When no controlled stop mode is configured, this box displays the message: No CS Input Configured . Configure Input link will opens the Digital I/O screen where you can configure the controlled stop mode. | |

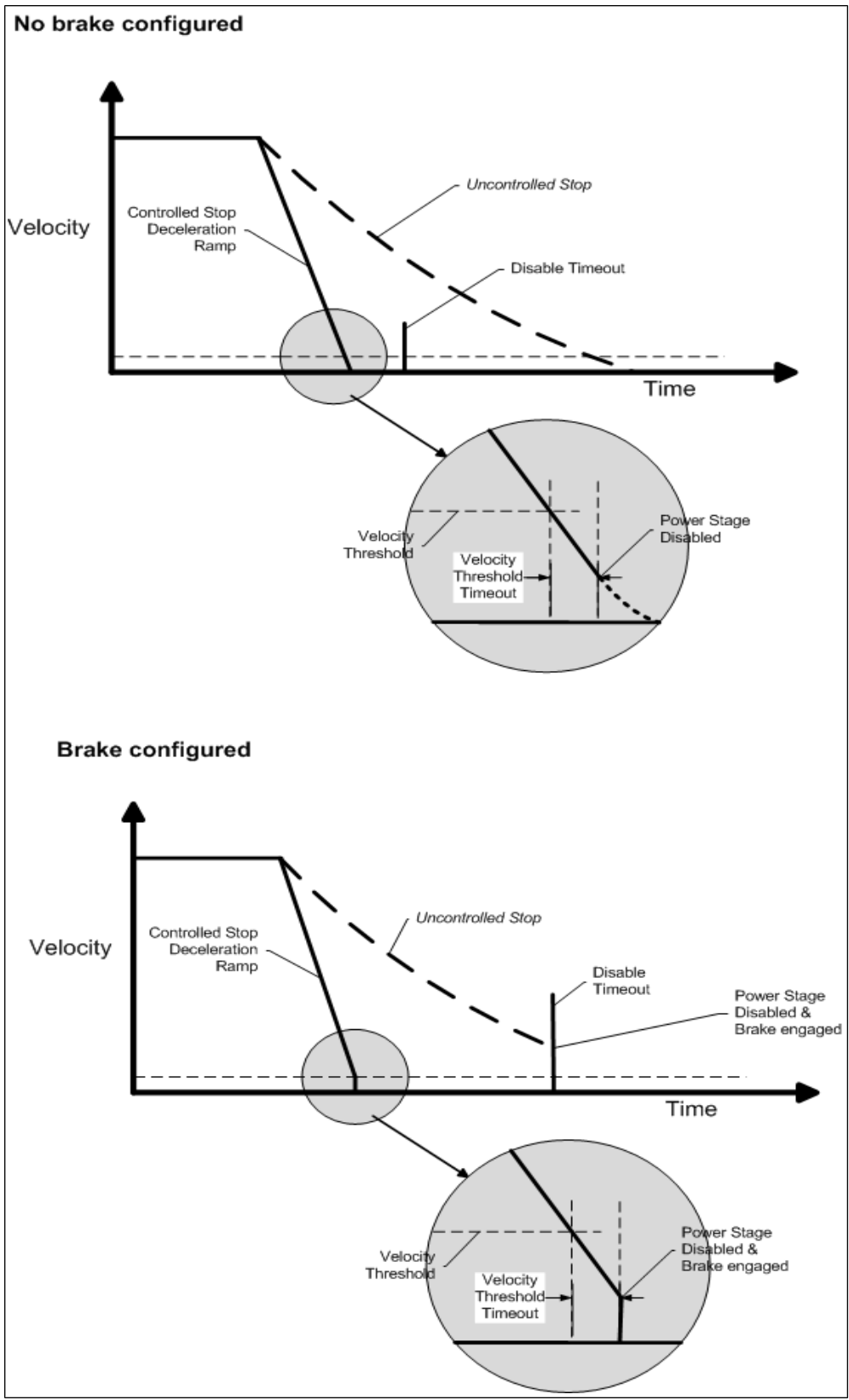
10.9.5 More/Less Button

The **More** button reveals the status diagram for the controlled stop configuration. It also displays the block diagram for the control stop. Two block diagrams available: one for a brake fitted and another for no brake.

Control stop status diagram



Controlled Stop Block Diagram



10.10 Controlled Stop

In a controlled stop, drive motion is brought to a standstill in a controlled manner. The drive commands a zero velocity from the motor. The motor decelerates at the prescribed deceleration value (CS.DEC).

A controlled stop can occur in four ways:

- The user configures a programmable digital input to mode 13 using DINx.MODE. For example, if [DIN1.MODE 13](#) is applied, digital input 1 is set to controlled stop.
- Either a controller or the user (through the WorkBench terminal window) initiates a software disable (DRV.DIS) command.
- CANopen PDO is set to 3442.

| CANopen Property | Value |
|------------------|-----------------|
| Index/Subindex | 3442/0 |
| Data Type | Unsigned 8 |
| Access | W/O |
| PDO mappable | N/A |
| Description | Controlled stop |
| ASCII Object | |

- A fault initiates a controlled stop from the drive. See 17.1 Fault and Warning Messages for the faults which initiate a controlled stop.

The controlled stop mechanism is activated in the following cases:

1. DRV.DISMODE = 2 and user executes DRV.DIS from the terminal or WorkBench disable buttons.

NOTE

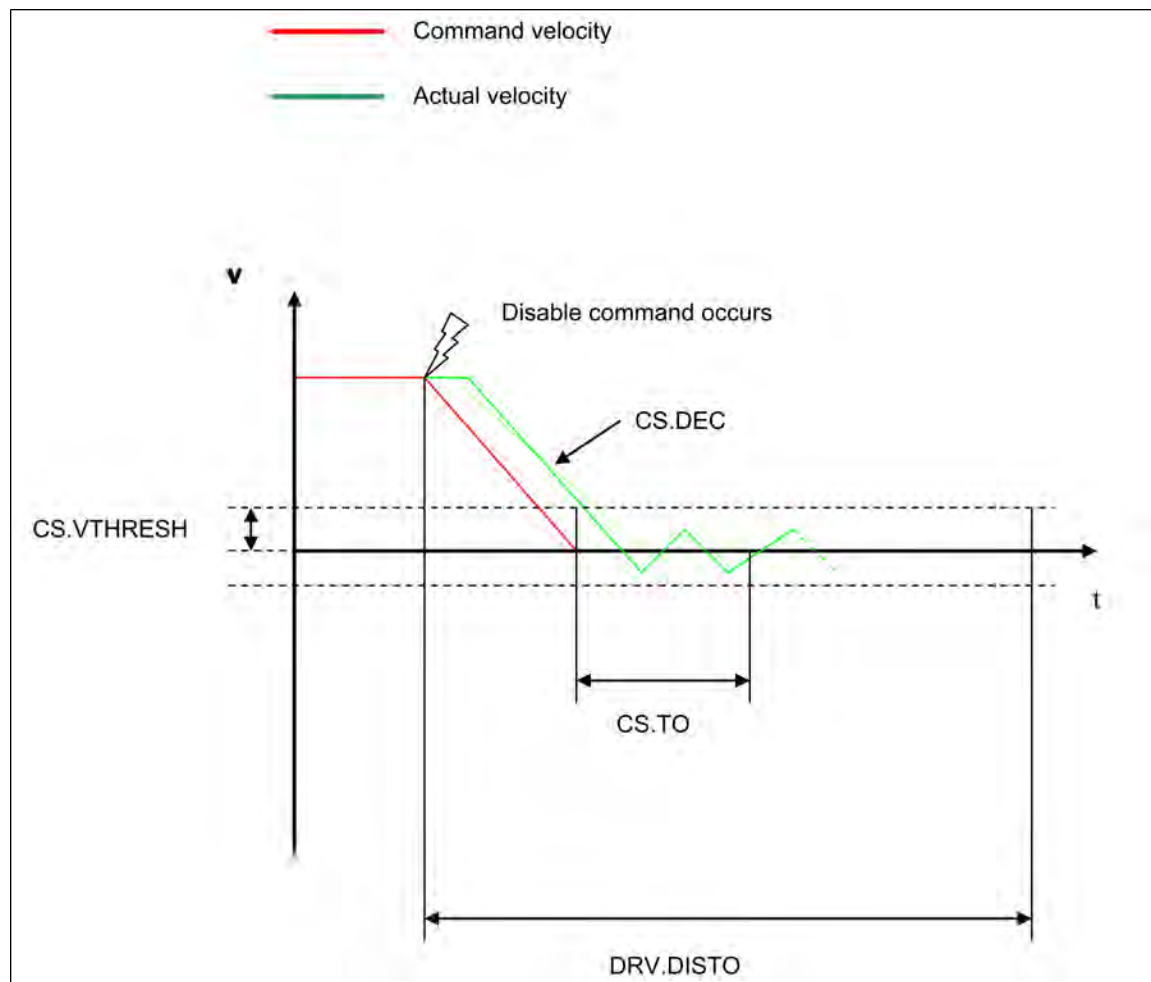
You must disable the drive in order to set DRV.DISMODE.

2. DRV.DISMODE = 2 and user executes DRV.DIS from a fieldbus connected to the drive.
3. A fault happens for which the reaction evolves controlled stop (CS). After the CS is executed, the drive disables.
4. A digital input mode (DINx.MODE) is set to 13. If the digital input state changes (active high or low according to DINx.INV) the CS is executed, then the drive disables.
5. HW limit switch: A digital input is defined as a positive (negative) limit switch (DINx.MODE 18 or 19). When the limit switch is met, the CS mechanism starts running. In this case, the parameter DRV.DISTO is not active.
6. SW limit switch: SWLS defines an active SW limit. When the limit is met, the CS mechanism starts running. In this case, the parameter DRV.DISTO is not active.

Use the drive CS parameters to configure a controlled stop as follows:

1. CS.DEC: Deceleration ramp that is used for disable.
2. CS.VTHRESH: Velocity 0 threshold. The motor shaft is considered as stopped as soon as the actual velocity (filtered through a 10 Hz filter, such as VL.FBFILTER) is within \pm CS.VTHRESH.
3. CS.TO: Velocity 0 time. The actual velocity must be consecutively within $0 \pm$ CS.VTHRESH for the time CS.TO, before the drive completes the CS process. This value is used since the motor can overshoot out of the VELO window depending on the gains, deceleration ramp, motor inertia and so on.
4. DRV.DISTO: Disable time out. This parameter sets an overall and independent running check as to whether or not the drive can achieve the disable state. If the VELO window set in step 3 is too small, it is possible that the drive may never reach the end of the CS process. The DRV.DISTO parameter and functionality addresses this issue by disabling the drive after the DRV.DISTO time elapses, even if the CS process did not end.

Controlled Stop Diagram



When configuring the controlled stop feature, please note the following:

- If the HW limit switch is active and any of the other CS activated, the only difference will be that in this case the DRV.DISTO will limit the time before disabling the drive.
- If the value of DRV.OPMODE of the drive is current mode, the drive will not execute the CS but instead stop immediately.
- Set DRV.DISTO to an appropriate value that will allow the motor to decelerate from any velocity to 0 with DRV.DEC. This value must also allow the motor to afterwards remain within VL.FB for CS.TO consecutively within $0 \pm \text{CS.VTHRESH}$.

The drive issues a fault FF703 in case that the DRV.DISTO counter expires during a controlled stop procedure.

10.10.0.1 Related Parameters and Commands

CS Parameters

CS.STATE: Reads the current state of controlled stop process (0 = controlled stop is not occurring. 1 = controlled stop is occurring).

DIN1.MODE TO DIN7.MODE

DRV.DIS

DRV.DISTO

DRV.DISMODE

10.10.1 Related topics:

10.12 Emergency Stop

10.1 Digital Inputs and Outputs

17.1 Fault and Warning Messages

10.11 Dynamic Braking

Dynamic braking is a method to slow a servo system by dissipating the mechanical energy in a resistor driven by the motor back EMF. The drive has a built in advanced (patent pending) dynamic braking mode which operates fully in hardware. When activated, the drive shorts the motor terminals in phase with the back EMF (q axis) but continues to operate the non-force producing current loop (d-axis) with 0 current. This action forces all of the dynamic braking current toward stopping the motor current and insures the fastest stopping/amp of motor terminal current.

The drive hardware also limits the maximum dynamic braking motor terminal current via the DRV.DBILIMIT parameter to prevent the drive, motor, and customer load from encountering excessive currents/forces. When the current is not being limited, the mechanical energy is dissipated in the motor terminal resistance. When the current is being limited, energy is returned to the drive bus capacitors. When the amount of returned energy raises the bus capacitor voltage enough, the drive activates the regeneration control to start transferring the returned energy to the regen resistor. This resistor could be internal or external to the drive depending on drive model and drive wiring.

Whether and how the drive uses dynamic braking mode depends on the drive disable mode (DRV.DISMODE) setting.

10.11.1 Drive Regeneration

When the servo motor is slowing down at a rate faster than friction and motor losses would slow the motor, then mechanical energy can be returned to the drive. This returned energy initially drives the internal bus voltage upwards. When the returned energy is high enough, the regeneration control transfers the excess returned energy into the regeneration power resistor. If the regeneration control cannot fully handle the power returned (for example, because there is not a regen resistor present or its resistance value is too high), then the bus voltage will continue to rise and a bus over voltage fault will happen and disable the drive completely, which allows the motor to freewheel.

10.11.1.1 AKD-x00306 to AKD-x00606

These units do not have an internal regeneration resistor. In many applications machine friction, motor losses, and the limited bus capacitor energy absorption handles the application. But, depending on the exact application requirements, an external resistor can be connected.

10.11.1.2 AKD-x01206 to AKD-x02406 and AKD-xzzz07

These units have an internal regeneration resistor plus the capability to connect an external resistor if higher power levels are needed.

Related Topics

See section 6.14 *Dynamic Braking* in the *AKD Installation Manual* for detailed information on dynamic braking.

DRV.DISMODE

DRV.DBILIMIT

10.12 Emergency Stop

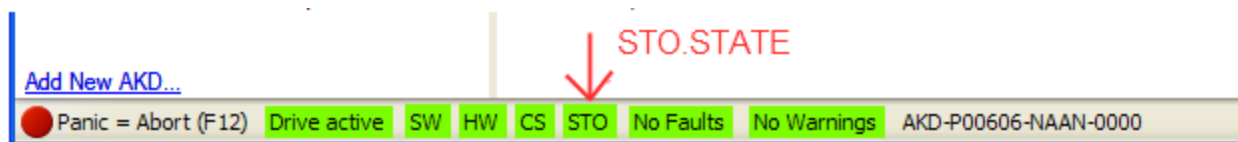
10.13 Safe Torque Off (STO)

Safe torque off (STO) is a restart lock safety feature that prevents a system restart. The STO input provides direct electrical control of the power stage of the AKD drive; it bypasses the processor and will disable the power stage irrespective of software or other hardware signals.

STO is controlled by a digital input on the X1 connector (pin 3) that must have 24V applied to it; otherwise, the drive will not enable. If the STO digital input does not have 24V applied to it and you attempt to enable the drive (using the hardware and software enables) the drive will generate a 602 “Safe Torque Off” fault. If you see this fault you need to apply 24V to the STO input and then clear the fault (DRV.CLRFAULTS) before you can enable the drive. The STO will not generate a fault until you try to enable the drive. You can read the current faults using DRV.FAULTS.

```
-->DRV.FAULTS
602: Safe torque off.
-->
```

The current state of the STO can be read using the STO.STATE parameter (this returns 1 if 24V is being applied to this input). WorkBench also shows you the state of the STO input in the status bar at the bottom of the window.



10.14 Under Voltage Fault Behavior

You can adjust the conditions for an under voltage fault using VBUS.UVMODE in the WorkBench terminal screen:

VBUS.UVMODE = 1 (default)

The drive will not report an under voltage fault unless the drive is enabled and VBUS.VALUE falls below VBUS.UVFTHRESH

VBUS.UVMODE = 0

The drive will report an under voltage condition any time VBUS.VALUE falls below VBUS.UVFTHRESH.

When an under voltage fault occurs, the drive is disabled and issues the following alerts:

- WorkBench alert: 502 Bus Under Voltage
- Drive LED alert: Left LED displays [F], right LED displays [u-V].
- Fault relay output turns on.

11 Using Command Source and Operating Modes

| | |
|--|-----|
| 11.1 Overview..... | 120 |
| 11.2 Using Command Source and Operation Modes..... | 120 |
| 11.3 Current Loop..... | 121 |
| 11.4 Velocity Loop..... | 124 |
| 11.5 Position Loop..... | 126 |

11.1 Overview

Operation modes (opmodes) allow you to set up your drive to communicate directly over the Ethernet input, a specific fieldbus, or an analog or digital control.

There are two basic components to how you will command the drive and how it will behave. The “Service mode” indicates how the drive will be communicated to. The drive has communication options via Ethernet, a variety of fieldbuses, through an analog input, and through electronic gearing, or digital inputs. The second component ties to which loop you will be controlling (torque, velocity, or position).

11.2 Using Command Source and Operation Modes

There are two methods to access these two parameters within WorkBench. The first is by selecting the **Settings** screen from the left hand tree. By clicking on the top level of the **Settings** folder, you access the graphical representation of both the Command source and Operation Mode. The drop down box then allows you to select the desired command type and control loop you wish to activate. Please note that some Command Sources can only be used with certain control loops (as an example, Electronic Gearing can only be used in position loop Operation Mode).

11.2.1 Command Source

The command source sets how you communicate with the drive. Initially, you might be communicating via your PC using the Ethernet connection. Each Command Source is listed below:

11.2.1.1 Service

This is the most common source used when establishing initial communications with the drive to set the system up, and when you may need to “service” the drive. The Service source communicates with your PC via the Ethernet port located on the top of the drive at connector X11.

11.2.1.2 Fieldbus

When using a fieldbus, such as CANOpen or EtherCAT, the drive is set with this command source. For CAN-Open, use the connectors X12 and X13 located on the top of the drive. For EtherCAT, Connectors X5 and X6 located on the front of the drive are used.

11.2.1.3 Electronic Gearing

If the drive will be used to follow the output of an external encoder, following motion using an electronic gear ratio, then you should use this command source. When using Electronic Gearing, the Operation mode must be set in the Position Loop mode. This is also the mode used for step and direction inputs as well.

11.2.1.4 Analog

This command source allows the drive to be controlled from an analog source. Typically a +/- 10 Vdc signal is attached to connector X8 pins 9 and 10. Varying the analog input will then vary torque, velocity, or position based on the Operation mode selected.

11.2.1.5 Operation Mode

The operation mode identifies which servo loop you will be controlling. The drive offers torque, velocity, or position control. From the setting screen page, a graphical representation of the loop is displayed. By clicking these graphical representations, you can access additional loop information such as gains, filters, and other settings.

11.2.2 Related Parameters

DRV.CMDSOURCE

DRV.OPMODEDRV.OPMODE

11.3 Current Loop

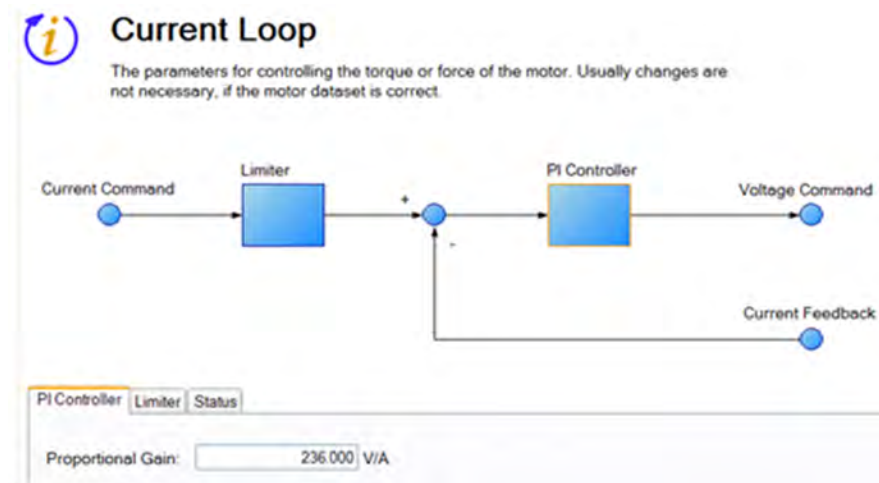
11.3.1 Overview

The current loop is active when the drive operates in current torque (current) mode ($DRV.OPMODE = 0$). The parameters that govern the current loop are shown in the Current Loop view. The various types of tuning for the drive adjust these parameters automatically, so you normally do not need to adjust the current loop parameters in the current loop screen. The Current Loop view includes an active block diagram. If you click on a block in the diagram, the appropriate tab opens below.

A more detailed block diagram for the current loop is included in Block Diagrams

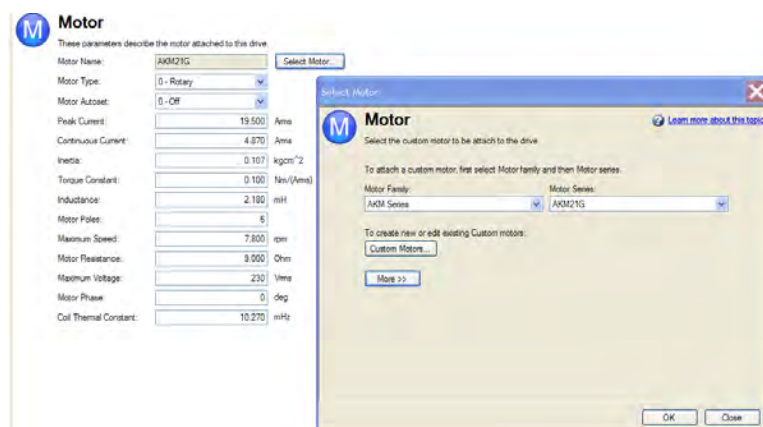
11.3.2 Current Loop Gain

The current loop is tuned based on the inductance of the motor used with the drive. The current loop gain is automatically set so that the idealized current loop crossover frequency is $IL.KP/L$ in rad/sec where L is the motor line-line inductance.



Current loop gain is set automatically by the drive using the following methods:

- **Option A.** When a feedback device is automatically identified by the drive and the motor data is automatically populated (Motor Autaset = 1-On) the proportional gain of the current loop ($il.kp$) is set based on the motor data and is shown as a read-only parameter in the current loop screen.
- **Option B.** When the motor is selected using the motor database or using the custom motor tool, the imported inductance value is used to set the current loop proportional gain.



NOTE

Manual adjustments to the current loop proportional gain parameter are not normally required during the motor tuning procedure. If manual adjustments are made to the current loop proportional gain parameter, repeating the motor setup procedure will overwrite the changes and restore the value to the Kollmorgen calculated value.

Related Parameters

IL Parameters

DRV.OPMODE

11.3.3 Current Loop Gain Scheduling

This feature is needed when the motor inductance is saturated during normal operation. Because the current loop gain is calculated using the motor inductance, if the inductance changes, the current loop is at risk of becoming unstable.

If a motor is being used under high loads, the inductance is probably saturating if one or more of the following occurs:

- Audible chirping occurs.
- Chirping gets louder with higher commanded current.
- Instability occurs under high current loads (at or near MOTOR.IPEAK)

To rectify this problem, you can use gain scheduling to change the current loop gain as a function of the current command (IL.CMD).

11.3.3.1 Using the Gain Scheduling View in WorkBench

To use this feature effectively, you should have either an inductance chart of motor inductance as a function of current, or you must have an idea of how the current loop is performing.

If a chart of motor inductance versus current command is available, it is possible to calculate the required current loop gain values over the range of motor currents.


$$\text{Current Loop Gain} = 2000 * 2\pi * \text{Motor Inductance (H)}$$

Example

Motor Inductance is 3.19 mH, current loop gain would be 40.01

$$\text{Current Loop Gain} = 2000 * 2\pi * 0.00319 = 40.01$$

Once suitable values have been identified, you can enter these values into WorkBench in the **Gain Scheduling** view. The **Import** and **Export** buttons at the bottom of the view allow you to import and export data as .csv files.



Gain Scheduling

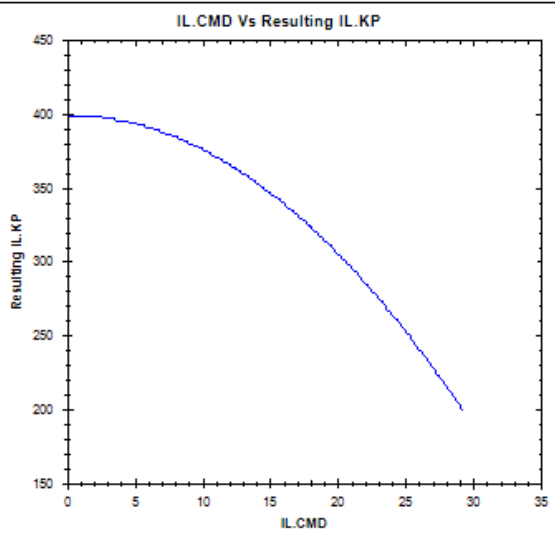
[? Learn more about this topic](#)

Advanced user current loop gain lookup table for special motors that have varying inductance.

Drive I_{Peak} : Arms

Proportional Gain : V/A

| IL.CMD (A) | Table Index | Value (%) | Resulting IL.KP |
|------------|-------------|-----------|-----------------|
| 0.000 | 0 | 100.000 | 400.072 |
| 0.115 | 1 | 99.607 | 398.500 |
| 0.229 | 2 | 99.607 | 398.500 |
| 0.344 | 3 | 99.607 | 398.500 |
| 0.459 | 4 | 99.607 | 398.500 |
| 0.573 | 5 | 99.607 | 398.500 |
| 0.688 | 6 | 99.607 | 398.500 |
| 0.803 | 7 | 99.607 | 398.500 |
| 0.917 | 8 | 99.607 | 398.500 |
| 1.032 | 9 | 99.607 | 398.500 |
| 1.146 | 10 | 99.607 | 398.500 |
| 1.261 | 11 | 99.607 | 398.500 |
| 1.376 | 12 | 99.607 | 398.500 |



By default, the current loop value will be what is defined in IL.KP across the whole range of currents. To change the value of IL.KP over a range of values, simply enter a scaling term: 0 - 100% of the current value.

Example

If the current loop gain entered on IL.KP was 40.124 (shown above) and a current loop gain of 36 was desired, a scaling term of 90% should be entered for the desired current ranges.

$$40.124 * 0.90 = 36.112$$

11.3.3.2 Using the Terminal View for Gains Scheduling

You can also use the Terminal to set up the Gains Scheduling Table. If you use the Terminal, then two parameters are required for each lookup table point: IL.KPLOOKUPINDEX and IL.KPLOOKUPVALUE. IL.KPLOOKUPINDEX specifies the lookup table index (0 – 255), and IL.KPLOOKUPVALUE specifies the scaling term (0 – 100%) to scale IL.KP.

The current to which a lookup index refers can be calculated as follows:

$$\text{IL.CMD Range} = \text{DRV.IPEAK}/157 * \text{IL.KPLOOKUPINDEX}$$

A full list of table values can also be retrieved using IL.KPLOOKUPVALUES, which returns a comma delimited table as follows:

```
-->IL.KPLOOKUPVALUES
```

```
Index Value
```

```
0, 100.000
```

```
1, 100.000
```

```
2, 100.000
```

```
3, 100.000
```

```
4, 100.000
```

5, 100.000
 6, 100.000
 7, 100.000
 8, 100.000
 9, 100.000
 10, 100.000

11.4 Velocity Loop

11.4.1 Overview

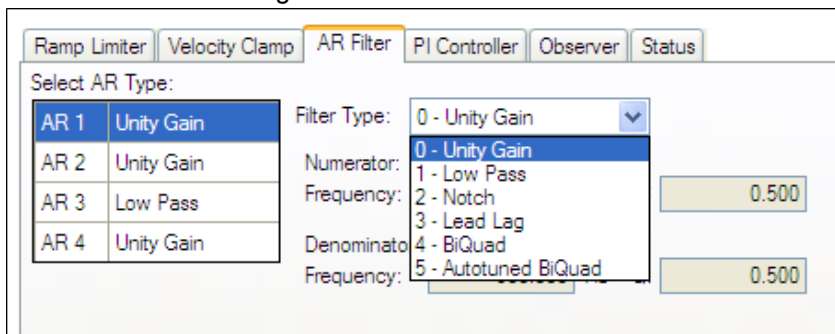
The velocity loop is active when the drive operates in velocity mode (DRV.OPMODE = 1) or position mode (DRV.OPMODE = 2). The parameters that govern the velocity loop are shown in the Velocity Loop view. The various types of tuning for the drive adjust these parameters automatically, so you normally do not need to adjust the velocity loop parameters in the velocity loop screen.

A detailed block diagram for the velocity loop is included in Block Diagrams.

11.4.2 Tabs in the Velocity Loop View

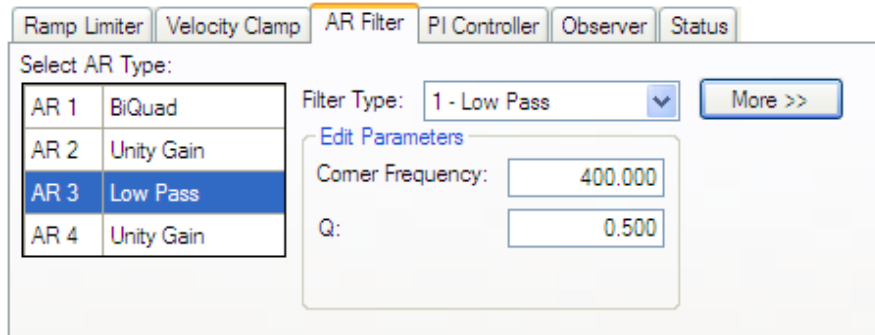
The velocity view includes an active block diagram. If you click on a block in the diagram, the appropriate tab opens below.

- **Ramp limiter** . The ramp limiter consists of the acceleration limits of the drive. These acceleration limits override both motion task and electronic gearing acceleration limits, so they must be set higher than the highest required motion task acceleration or gearing acceleration value. These acceleration and deceleration limits are also shown in the Service Motion view and the Limits view (DRV.ACC and DRV.DEC).
- **Velocity clamp**. The velocity clamp affects the maximum speed of the drive when the command source is service (DRV.CMDSOURCE = 0). This speed limit affects motion commanded in service motion and in motion tasks. These limits are also found in the limit screen on WorkBench. (VL.LIMITP and VL.LIMITN)
- **AR1, AR2, AR3, AR4**: These values are the independent bilinear quadratic (bi-quad) filters inside the drive. AR1 and AR2 are in the forward path and AR3 and AR4 are in the feedback path. These bi-quad filters can each be configured in five different modes.



0–Unity Gain. The filter is off, and it will not affect the loop.

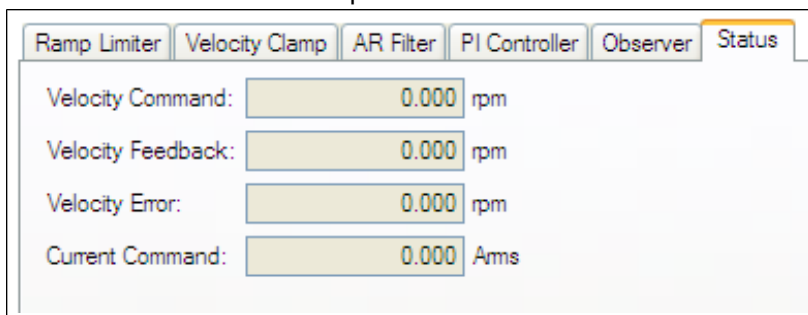
1–Low Pass. In modes 1, 2, and 3, the bi-quad filter is configured for each respective type of filtering. The Edit Parameters field is used to set up the filter. The actual bi-quad filter values are shown to the left:

**2–Notch****3–Lead Lag**

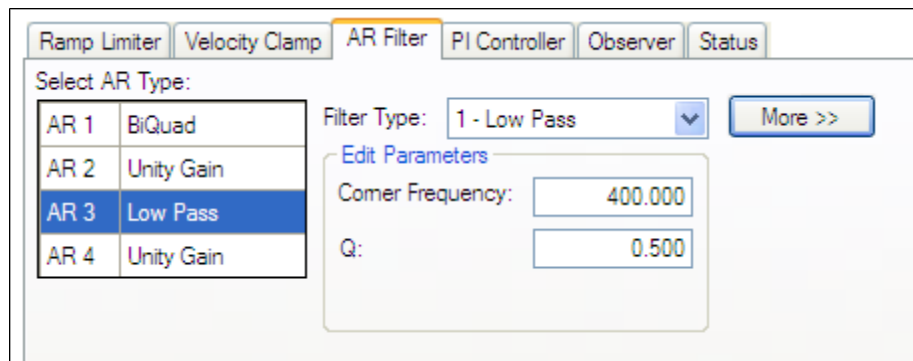
4–Bi-quad . A manually configured Bi-quad filter. This is an advanced tuning function.

5–Autotuned Biquad . When the PST sets a filter after the PST process is complete, the values are input into the Bi-Quad filter and are shown as a read only values.

- **Status.** The status tab shows parameters that are relevant to the velocity loop performance.

**11.4.3 Velocity Loop Default Settings and Changes**

By default, a PI loop with a low-pass filter (AR3) is set in the drive.



The default value for the low-pass filter is 400 Hz. The low-pass filter is important for disturbance rejection, and it also reduces the audible noise of the system.

11.4.3.1 Velocity Loop Changes Based on Slider Tuning

Slider tuning (see 14.1 Slider Tuning) uses the slider control to adjust the proportional gain and integral gain values of the velocity loop based on the desired bandwidth. If you adjust the bandwidth using the slider tuner and then return to the velocity loop screen, you will see different values inside the proportional gain and integral gain fields. No adjustment is made to the filters automatically by using the slider tuner. Only the proportional and integral terms are adjusted.

11.4.3.2 Velocity Loop Changes Based on PST

When the PST (see 14.3 Using the Performance Servo Tuner) is used, changes are made to the proportional gain, integral gain, filters, and other parameters not related to the velocity loop screen directly. The values adjusted are dependent on the drive, motor, load, and the PST settings. The filters that are adjusted by the PST are automatically put into mode **5-Autotuned Bi-quad**.

No adjustments can be made to mode **5- Autotuned BiQuad** filters that are set by the PST. If adjustment to the tuning of the system is desired after the PST process is completed, then these adjustments should be made in the settings of the PST. The PST process can then be repeated.

Related Parameters

VL Parameters

DRV.ACC

DRV.CMDSOURCE

DRV.DEC

DRV.OPMODE

Related Topics

10.7.5 Limits

12.2 Motion Tasks

12.4 Service Motion

10.7 Electronic Gearing

14 Tuning Your System

11.5 Position Loop

11.5.1 Overview

The position loop is active when the drive operates in position mode (DRV.OPMODE = 2). The parameters that govern the position loop are shown in the Position Loop view. The various types of tuning inside AKD adjust these parameters, so you normally do not need to adjust the position loop parameters in the position loop screen.

A detailed block diagram for the position loop is included in Block Diagrams.

11.5.2 Tabs in the Position Loop View

The position loop view includes an active block diagram. If you click on a block in the diagram, the appropriate tab opens below.

- **Gains.** This tab shows the gains for the position loop.
- **Limiter.** The value in the **Maximum Position Error** box (PL.ERRFTHRESH) limits the position error

(PL.ERR) that can be present. When the maximum position error is exceeded, the drive generates fault F439, Following Error. If the maximum position error is set to 0 (default) then the maximum position error is ignored.

The screenshot shows the 'Limiter' tab selected. Below the tab name, there is a label 'Maximum Position Error:' followed by a text input field containing the value '300.000' and the unit 'deg'.

- **Status.** This tab shows the present value of commanded position (PL.CMD), position feedback (PL.FB), position error (PL.ERR), and velocity command (VL.CMD).

11.5.3 Position Loop Default Behavior and Changes

By default, only a proportional gain (PL.KP) is applied in the position loop.

The screenshot shows the 'Gains' tab selected. It contains several parameter fields:

- Proportional Gain: 100.000 (rev/s)/rev
- Integral Gain: 0.000 Hz
- Feed Forward Gain: 0.000
- Integral Input Saturation Level: 360.000 deg
- Integral Output Saturation Level: 360.000 deg

11.5.3.1 Position Loop Changes Based on Slider Tuning

Slider Tuning (see 14.1 Slider Tuning) adjusts the proportional gain of the position loop (along with velocity loop view parameters; see 11.4 Velocity Loop). If you adjust the bandwidth using the slider tuner, then when you return to the position loop screen, you will see a change to the proportional gain only. No adjustment is made to the integral gain or feedforward gain through the slider tuner. The integral saturation levels are not applicable when the integral gain is set to 0. In the **Gains** tab, the boxes for these values may be populated with default values whether or not the integral gain is set to 0.

11.5.3.2 Position Loop Changes Based on PST

When the Performance Servo Tuner (PST, see 14.3 Using the Performance Servo Tuner) is used, changes are made to the position loop proportional gain, integral gain, feedforward gain and other parameters not related to the Position Loop view directly. The values adjusted are dependent on the drive, motor, load, and the PST settings.

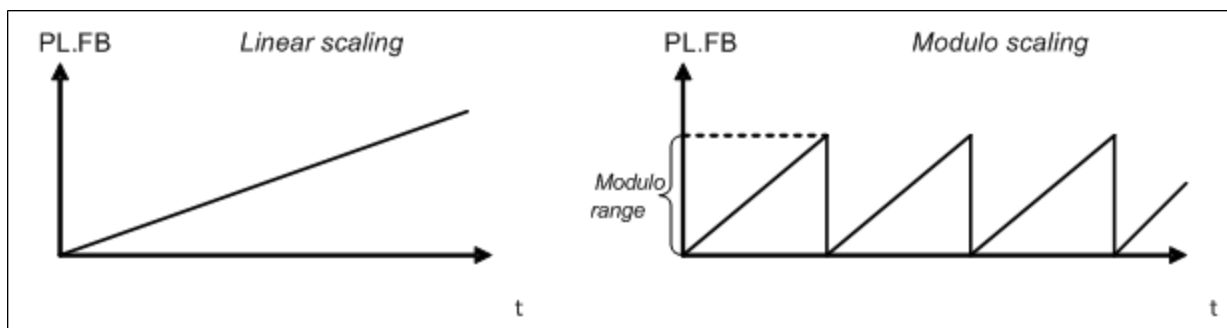
Related Parameters

PL Parameters
 DRV.OPMODE
 VL.CMD

11.5.4 Modulo Position

Modulo position is a function that simplifies rotational applications such as unidirectional rotating assembly tables. When enabled, the modulo axis feature converts several position-based parameters to fit in a defined modulo range. Once this range is defined, a given position value will roll over at the end of the modulo range and return to the beginning of the modulo range. This behavior affects some drive functions, which work with modulo-scaled position variables when the modulo-feature is enabled.

The following figure describes the progress of the actual position value (PL.FB) for linear scaling and modulo scaling when the motor moves continuously in a positive direction:



11.5.4.1 Setting up the modulo axis in WorkBench

You can set up the modulo axis from the **Modulo** view in WorkBench.

Modulo

Modulo is used to set a rollover value for the position scale.

[Learn more about this topic](#)

Modulo Position: 1 - Enabled

Modulo Range Begin/End: deg

Modulo Range Begin/End: deg

Direction For Absolute Motion Task: 1 - Positive

Position Feedback: deg

| Button or Box | Description |
|---|---|
| Modulo Position | Enables or disables modulo (PL.MODPEN) |
| Modulo Range Begin/End | Sets the beginning and end of the modulo range (PL.MODP1, PL.MODP2) |
| Direction for Absolute Motion Task | Sets positive or negative direction (PL.MODPDIR) |
| Position Feedback | Reads and displays the position feedback (PL.FB) |

11.5.4.2 Setting up the modulo axis from the Terminal

You can use the following parameters to configure the modulo-axis feature:

- PL.MODPEN: Enables or disables the modulo-axis feature.
- PL.MODP1: Defines either the beginning or the end of the modulo range, depending on the PL.MODP2 setting.
- PL.MODP2: Defines either the beginning or the end of the modulo range, depending on the PL.MODP1 setting.

11.5.4.3 Parameters affected by the modulo axis

The following parameters are converted into modulo format when the values of these parameters are queried by a user, a fieldbus, or the software oscilloscope.

- PL.FB: The actual position of the drive is converted into modulo scaling.
- PL.CMD: The command position of the drive is converted into modulo scaling.
- CAP0.PLFB: The actual position of the drive, which has been captured by the capture engine 0, is converted into modulo scaling.

- CAP1.PLFB: The actual position of the drive, which has been captured by the capture-engine 1, is converted into modulo scaling.

11.5.4.4 Drive functions affected by modulo axis

Software limit switch

The software limit switches in the drive compare the actual position (PL.FB) with threshold values. Motion is stopped when the actual position exceeds the software limits. Since PL.FB is affected by the modulo-axis feature, the software limit switches monitor the modulo-converted PL.FB value. Software limit switches with thresholds outside of the modulo range never limit motion.

Programmable limit switch

The programmable limit switches compare the actual position (PL.FB) with selectable thresholds and then status flags are set to "true" when the actual position is currently within these position limits. The programmable limit switches monitor the modulo-converted PL.FB value. Programmable limit switches that are set outside the modulo-range never become active.

Digital output Modes 5 and 6

The digital output modes 5 and 6 (position greater than x, position less than x) compare the actual position of the drive with thresholds and activate the associated outputs when PL.FB is lower or higher than the threshold. The digital output mode functionality monitors the modulo-converted PL.FB value. Position thresholds that are set outside the modulo range continuously activate or deactivate the digital output.

Motion tasking to absolute target positions

When modulo is enabled, absolute motion tasks assume that the command is modulo converted. Absolute motion tasks to target positions outside of the modulo range generate a warning, n"Fault and Warning Messages" (=> p. 240) (Motion task target position is out of modulo range).

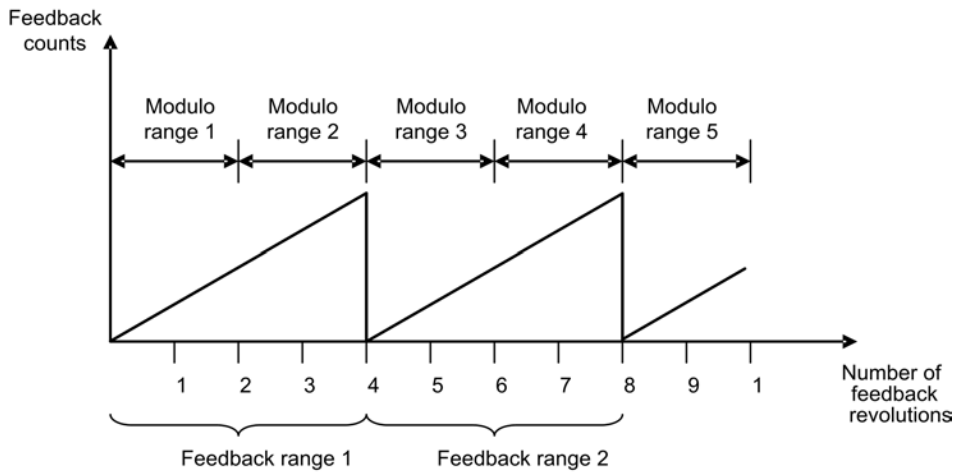
11.5.4.5 Using the modulo position feature with multiturn encoders

A special case exists for the following combinations of events:

- The drive is connected to a multiturn feedback device.
- The modulo axis feature is enabled.
- The selected modulo range does not fit as an integer in the range of the multiturn feedback.
- The application moves further than the total amount of multiturn feedback revolutions. In this case, problems occur because the multiturn feedback position overrun and a modulo range position rollover points do not occur at exactly the same position.

After powerup of the drive, the actual position (PL.FB) will be read from the multiturn feedback device. This position can be considered as a position within the feedback range as described in the figures below.

The following figure illustrates drive behavior when the selected modulo range fits as an integer within the multiturn feedback range. For simplicity, assume that one multiturn feedback range describes four feedback revolutions and the selected modulo range is set to two feedback revolutions.

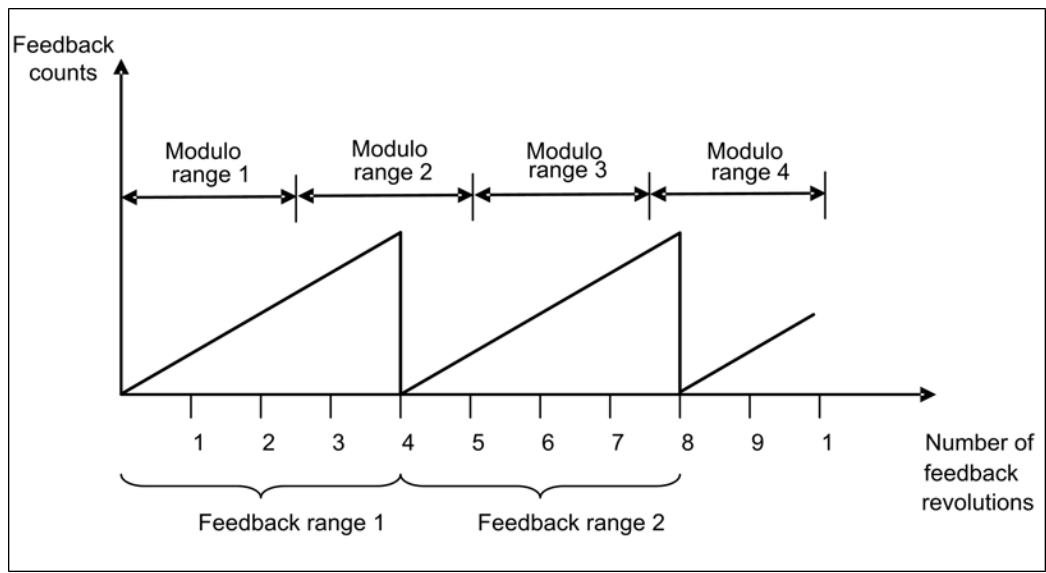


As described in the figure above, the selected modulo range of the drive is repeated exactly at the point where the connected multiturn feedback rolls over (Modulo range 1, 3, 5...). The application can move for several multiturn feedback ranges and the drive can recalculate the modulo position correctly after a power cycle. The positions within the modulo ranges represent the same value in modulo format for each feedback range.

Example

The modulo-converted position, which represents 5 or 9 feedback revolutions, corresponds to the modulo position, which represents 1 feedback revolution.

The next figure illustrates the drive behavior when the selected modulo range does not fit as an integer within the multiturn feedback range. For simplicity, assume that one multiturn feedback range describes four feedback revolutions and the selected modulo range is set to 2.5 feedback revolutions.



As described in the figure above, the selected modulo range is not repeated exactly at the place where the connected multiturn feedback rolls over. The application can move for several multiturn feedback ranges, but the drive cannot calculate the modulo position correctly after a power cycle.

Example

The modulo-converted position, which represents five feedback revolutions, does not correspond to the modulo-position, which represents one feedback revolution.

This page intentionally left blank.

12 Creating Motion

| | |
|---|------------|
| 12.1 Homing | 134 |
| 12.2 Motion Tasks | 144 |
| 12.3 Registration Moves (Motion Task Blending) | 149 |
| 12.4 Service Motion | 152 |
| 12.5 Jog Move | 153 |
| 12.6 Drive Motion Status | 153 |

12.1 Homing

12.1.1 Overview

Homing is used to mechanically move a motor (connected to a mechanism) to a specific location on the machine, referred to as “home”. Motion tasks then use this home as a reference point for movements that must base a known position on this reference point. Motor movement is usually controlled by a variety of limit switches (end of travel) and a home reference switch. Using these reference points with the logic of the drive allows the machine to find and set the home reference point.

12.1.2 Using Homing

The AKD includes a variety of homing methods (set with HOME.MODE) to accommodate your machine needs:

- Home using current position (HOME.MODE 0)
- Find limit input (HOME.MODE 1)
- Find input limit then find zero angle (HOME.MODE 2)
- Find input limit then find index (HOME.MODE 3)
- Find home input (HOME.MODE 4)
- Find home input then find zero angle (HOME.MODE 5)
- Find home input then find index (HOME.MODE 6)
- Find zero angle (HOME.MODE 7)
- Move until position error exceeded (HOME.MODE 8)
- Move until position error exceeded, then find zero angle (HOME.MODE 9)

Each of these homing methods offers a different way to achieve a home reference point based on your particular system mechanics. All homing methods provide the options of adjusting the acceleration, deceleration, and speed for homing moves. In addition, once the homing move is completed, you can either set an offset position or make an offset move as required. Homing modes, guidance for mode selection, and homing examples are included in 12.1.2.2 Selecting and Using Homing Modes

Note: When using any of the methods that use homing switches and limits, please refer to the Input/Output section for proper wiring techniques.

12.1.2.1 Home Default Window

The Home window provides a means to select your homing method and configure the homing settings. This window also provides a simple controls to start homing and confirm homing success.

Home

This page is used to issue a homing command. The home command is used to zero the drives position.

Select the type of homing motion you wish to use:

1 - Find limit input

Settings

Acceleration: rpm/s

Deceleration: rpm/s

Direction:

Distance: deg

Position: deg

Position Lag: deg

Velocity: rpm

Negative Limit Switch: [Configure Inputs](#)

Controls

Found:

Done:

Active:

Error:

Position Feedback: deg

Auto Homing:

Drive is inactive.

[Goto Drive Motion Status](#)

Mode Selection:

Use this box to select the appropriate homing mode. Homing modes are described below in [Selecting and Using Homing Modes](#). The active options in the **Settings** area change depending on the homing mode selected.

Settings:

- **Acceleration:** Sets the acceleration ramp used during the homing procedure.
- **Deceleration:** Sets the deceleration ramp used during the homing procedure.
- **Direction:** Sets the start direction for homing movement.
- **Distance:** Sets a prescribed distance you want the motor to move once the home reference point is found. A zero value (default) corresponds to the axis actively returning to the defined position found during the homing process.
- **Position:** Sets the current position to a prescribed value once the home reference point is found.
- **Position Lag:** Sets the position error threshold, which is used for indicating home reference when using the hardstop modes 8 and 9.
- **Velocity:** Sets the initial velocity used for homing moves.
- **Velocity Factor:** In modes where a limit is reached, and direction is reversed, the velocity factor allows you to reduce the velocity as a percentage of the homing velocity.
- **Positive/Negative Limit Switch/Home Reference/Peak Current:** These fields appear based on the mode selected. For homing to limits and home reference, this field will indicate how the digital inputs are configured as well as providing a link to the digital input page. For Homing to a hard stop, the Peak Current field allows you to set the peak current limit desired during homing.

Controls:

- **Found:** When the home reference is found, then this indicator is green.
- **Done:** When the home move is complete, this indicator is green.
- **Active:** This indicator will be green while the Home move is taking place.
- **Error:** This indicator will be red if something in the homing sequence failed.
- **Position Feedback:** This window reports the current value for PL.PFB.
- **Auto Homing:** Allows the system to auto-home on power up.
- **Start/Stop:** Click this button to start or stop the selected homing method.

12.1.2.2 Selecting and Using Homing Modes**Homing Mode 0: Home Using Current Position**

Using the current position is the most basic homing method. This method simply uses the current position of the motor as the home point reference. Two values allow you to further define homing with this method:

- **Distance:** A value other than zero will cause a movement of the motor the distance entered in counts (or other units based on your units settings). You can use this to establish a home point at some prescribed distance from the initial starting position of the motor. This home will be at the offset distance entered from zero.
- **Position:** You can use this parameter to set the value of the home position other than zero. This allows you to offset your home reference away from zero. PL.FB will be set to the value you enter when the motor reaches the home reference point (based on the method selected).

The distance and position offsets are available and behave similarly on all of the homing types. Either the motor will move an additional distance (distance value) after it finishes the homing method, or will set the position to the amount entered in the position value.

Homing Mode 0 Example

Use the current position as home and have the motor end motion 180 degrees from home:

1. Select Mode 0 from the drop down box.
2. Enter 180 into the **Distance** box.
3. Click **Start**,
4. The motor will move 180 degrees from the start position. The **Position Feedback** box (PL.FB) will show 180 (the motor is now sitting 180 degrees from home).

Homing Mode 1: Find Limit Input

The find limit input mode creates a move to a limit input. This method can be used if you have a positive or negative limit switch available that you want to establish as a home reference point. **NOTE: Limit switches should be set to Active Low (when switch power is lost, no current is flowing, thus drive homes at the point of the loss in power of the limit switch).**

The sequence of this homing mode is as follows:

1. The motor starts to move in the positive or negative direction, depending on the value you set in the **Home** screen (in the **Settings** section, **Direction** box).
2. The motor stops as soon as the hardware limit switch has been detected and then reverses direction.
3. The home position is set when the limit switch is no longer active. The actual and command position of the drive is immediately set to the home position value (HOME.P) and the motor ramps down to zero velocity. The axis is then moved to the position (HOME.P) + distance offset (HOME.DIST).

The values for distance and position can be used as described in homing mode 0.

⚠ CAUTION

When homing to a limit switch, the limit switch must remain in the triggered state while the motor decelerates to zero and begins to reverse. A very low acceleration rate combined with a high approach velocity may overshoot the switch and cause it to become active. This action will cause a homing error fault.

Homing Mode 1 Example

Use the positive end of travel limit as home reference, and then set this position to be -20 degrees.

1. Select Mode 1 from the drop down box and enter 20 into the Position box.
2. Set the direction to positive. When Start is selected, the motor will move until it encounters the positive end of travel switch.
3. As soon as the switch is triggered, the motor will reverse direction until the switch is no longer active.
4. As soon as the switch is no longer active, the position will be set to -20 degrees and the motor will ramp to 0. Depending on the velocity you are homing with, and the settings of the acceleration/deceleration ramps, the position feedback will be close to the position you entered.

Homing Mode 2: Find Input Limit then Find Zero Angle

Similar to the Find Input Limit method, the find input limit then find zero angle¹ mode follows the same steps, but upon completion of the move, it continues to move to find the zero angle reference of the motor.

NOTE: Limit switches should be set to Active Low (when switch power is lost, no current is flowing, thus drive homes at the point of the loss in power of the limit switch).

The specific steps are as follows:

1. The motor starts to move according to the direction (HOME.DIR) setting.
2. The motor stops as soon as the hardware limit switch has been detected and changes direction of movement.
3. The home position has been found as soon as the hardware limit switch is no longer active. The actual and the command position of the drive will immediately be set to the HOME.P value plus distance to the mechanical zero angle of the feedback device according to the current direction.
4. The motor moves to the home position (HOME.P) plus the distance move offset is applied (if present), which is located at the mechanical zero-angle of the feedback.

The values for distance and position can be used as described in home mode 0.

⚠ CAUTION

When homing to a limit switch, the limit switch must remain in the triggered state while the motor decelerates to zero and begins to reverse. A very low acceleration rate combined with a high approach velocity may overshoot the switch and cause it to become active. This action will cause a homing error fault.

Homing Mode 2 Example

Use the positive end of travel limit as home reference and then move to the zero angle of the motor

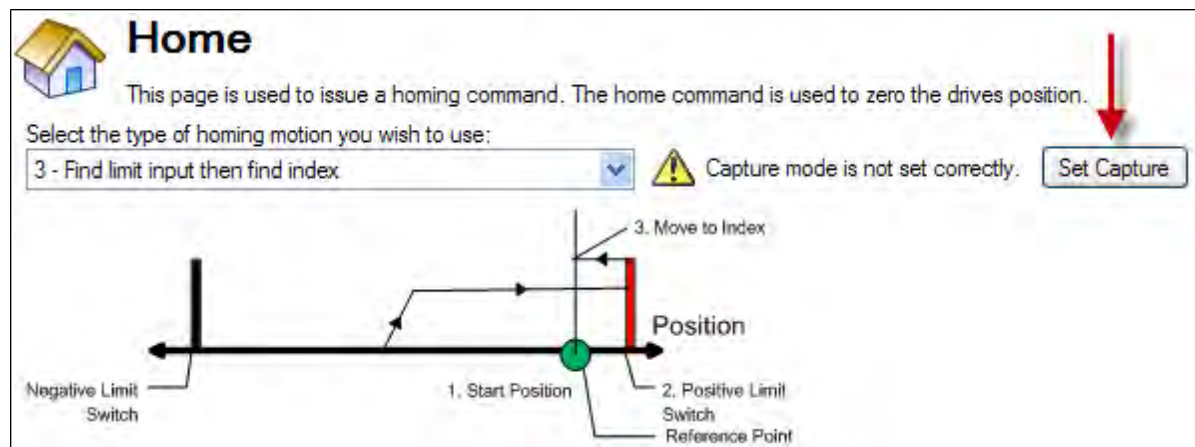
1. Use the positive end of travel limit as home reference and then move to the zero angle of the motor.
2. Select Mode 2 from the drop down box.
3. Set the Direction to Positive.
4. When Start is selected, the motor will move until it encounters the positive end of travel switch.

¹Mechanical zero angle of the feedback = 0 degree.

- As soon as the switch is triggered, the motor will reverse direction and move to the zero angle of the motor.

Homing Mode 3: Find Input Limit then Find Index

Similar to the Find Input Limit method, this follows the same steps, but upon completion of the move, it continues to move to find the index pulse of the motor. This method can only be used with feedback devices that have an index pulse such as incremental encoders and analog sine encoders with an index channel (Feedback selection 10, 11, 20, 21). This method requires that Capture Mode is turned on in the home screen. With Mode 3 selected, a **Set Capture** button appears (see arrow below). Click **Set Capture** to set the Position Capture mechanism correctly for proper homing with an index pulse.



Once homing is triggered, the homing routine is performed as follows:

- The motor starts to move according to the HOME.DIR setting.
- The motor stops as soon as the hardware limit switch has been detected and changes direction of movement.
- The motor ramps down to a reduced velocity as soon as the hardware limit switch is no longer active (please refer also to HOME.FEEDRATE). The drive is searching for the index-signal during this time. The home-position has been found as soon as the index-signal has been detected by the drive.
- The actual and the command position of the drive will be set to the HOME.P value as soon as the index pulse is found. The drive then ramps down to velocity 0. The axis is then moved to the position (home.p) + distance offset (home.dist).

CAUTION

When homing to a limit switch, the limit switch must remain in the triggered state while the motor decelerates to zero and begins to reverse. A very low acceleration rate combined with a high approach velocity may overshoot the switch and cause it to become active. This action will cause a homing error fault.

Homing Mode 3 Example

Use the positive end of travel limit as home reference, and then move to the index reference of the motor feedback device at 50% of the original home velocity.

- Select Mode 3 from the drop down box.
- Set the Direction to Positive.
- In the **Home** screen click **Set Capture**.
- Set the velocity factor to 50%.

5. When Start is selected, the motor will move until it encounters the positive end of travel switch. As soon as the switch is triggered, the motor will reverse direction, decelerate to a reduced velocity based on the Velocity Factor value, and move until the motor encounters the index pules of the feedback device.

Homing Mode 4: Find Home Input

Homing Mode 4 establishes the home reference based on an external home switch connected to a drive digital input (DINx.MODE - 11 Home Reference).

The sequence of this homing mode is as follows:

1. The motor starts to move according to the direction (HOME.DIR) setting.
2. The home position has been found as soon as the home-switch becomes active while traveling in the selected direction of motion (HOME.DIR). The actual and command position of the drive will immediately be set to the position (HOME.P) value and the motor ramps down to velocity 0. The axis is then moved to the position (home.p) + distance offset (home.dist).

The hardware limit switches are monitored during the homing procedure. The drive behaves as follows in case that a hardware limit switch is active before the home-switch has been activated:

- a. The motor changes the direction until the home switch is crossed.
- b. The motor ramps down to zero velocity and reverses direction again after crossing the home-switch.
- c. The home-switch will now be activated according to the direction (HOME.DIR) setting and when the home-position has been found. The actual and the command position of the drive will immediately be set to the position (HOME.P) value and the motor ramps down to zero velocity. The axis is then moved to the position (home.p) + distance offset (home.dist).

Homing Mode 4 Example

Move in the negative direction towards the home reference point and then move 180 degrees from the reference point

1. Select Mode 4 from the drop down box.
2. Set the **Direction** to **Negative** and enter 180 for distance.
3. Click Start.
4. The motor moves until it encounters the home reference switch. As soon as the switch is triggered, the motor moves an increment of 180 degrees as desired.

Homing Mode 5: Find Home Input then Find Zero Angle

The sequence of this homing mode is as follows:

1. The motor starts to move according to the Direction (HOME.DIR) setting.
2. The home position has been found as soon as the home-switch becomes active while traveling in the selected direction of motion (HOME.DIR). The actual and the command position of the drive will immediately be set to the Position (HOME.P) value plus the distance to the mechanical zero angle of the feedback device according to the current direction.
3. The motor moves to the home Position (HOME.P) value plus the distance move offset is applied (if present), which is located at the mechanical zero-angle of the feedback.

The hardware limit switches are monitored during the homing procedure. The drive behaves as follows in case that a hardware limit switch is active before the home-switch has been activated:

- a. The motor changes the direction until the home switch is crossed.
- b. The motor ramps down to zero velocity and changes afterwards the direction again after crossing the home-switch.
- c. The home-switch will now be activated according to the HOME.DIR setting and when the home-position has been found. The actual and the command position of the drive will immediately be set to the position (HOME.P) value plus distance to the mechanical zero angle of the feedback device according to the current direction.

- d. The motor moves to the home Position (HOME.P) value plus the distance move offset is applied (if present), which is located at the mechanical zero-angle of the motor feedback.

Homing Mode 5 Example

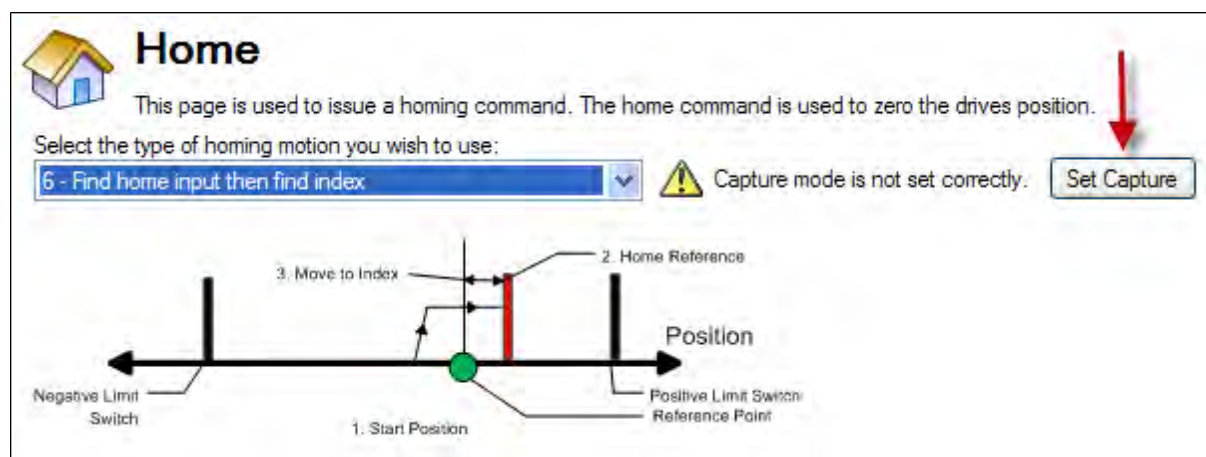
Move in the positive direction towards the home reference point and then move 60 degrees from the zero angle location.

1. Select Mode 5 from the drop down box.
2. Set the Direction to Positive and enter 60 for distance.
3. When Start is selected, the motor will move until it encounters the Home reference switch. As soon as the switch is triggered, the motor will move to the zero angle location plus an additional 60 degrees as desired.

Homing Mode 6: Find Home Input then Find Index

Similar to the Home input method, this follows the same logic as the other homing methods, first completing the home to input method, and finding the index pulse of the motor feedback.

This homing mode starts motion until a digital input, which is assigned to act as a home-switch, has been activated. The motor moves afterwards with a reduced velocity (HOME.FEEDRATE) until the index signal has been detected by the drive. **NOTE: This method requires that Capture Mode is turned on. This is done in the home screen. With Mode 6 selected, a “Set Capture” button will appear (see arrow below). Pressing the button sets the Position Capture mechanism correctly for proper homing with an index pulse..**



The home-switch must be activated according to the setting of the HOME.DIR setting.

The sequence of this homing mode is as follows:

1. The motor starts to move according to the HOME.DIR command.
2. The motor decelerates to a reduced velocity according to the HOME.FEEDRATE setting as soon as the home-switch becomes active during a motion in direction of the HOME.DIR setting.
3. The actual- and the command position of the drive will immediately be set to the HOME.P value as soon as the index-signal has been detected. The motor decelerates until velocity 0 has been reached.

The hardware limit switches are monitored during the whole homing procedure. The drive behaves as follows in case that a hardware limit switch is active before the home-switch has been activated:

- a. The motor changes the direction until the home-switch is crossed.
- b. The motor ramps down to zero velocity and changes direction again after crossing the home-switch.
- c. The home-switch will now be activated according to the HOME.DIR command. The motor decelerates to a reduced velocity according to the HOME.FEEDRATE setting as soon as the home-switch becomes active.
- d. The actual and the command position of the Drive will immediately be set to the HOME.P value as soon as the index-signal has been detected. The motor decelerates until zero velocity has been reached. The axis is then moved to the position (HOME.P) + distance offset (HOME.DIST).

Homing Mode 7: Find Zero Angle

The sequence of this homing mode is as follows:

1. The home value is immediately found by the drive and the actual and the command position of the drive will immediately be set to the Position (HOME.P) value plus the distance to the mechanical zero angle of the feedback device according to the current direction.
2. The motor moves to the home Position (HOME.P) value, which is located at the mechanical zero-angle of the feedback.

Homing Mode 7 Example

Move in the positive direction towards the zero angle location.

1. Select Mode 7 from the drop down box.
2. Set the direction to Positive.
3. When Start is selected, the motor will move to the zero angle location.

Homing mode 8: Move Until Position Error Exceeded

This method is also referred to as move to hard or mechanical stop. The AKD has several options related to this method as well. For this basic method, the motor will move until it encounters a hard stop, causing the position error to exceed a specific threshold you set. Once the threshold is exceeded, the motion stops and home reference is established. You can use the Distance or Position as described initially in this section. One caution; make sure you choose the direction appropriately to move off of the stop if you use the distance offset.

The sequence in this homing mode is as follows:

1. As this home move is initiated, the motor will move according to the Direction (HOME.DIR) value until the position error exceeds the Position Lag (HOME.PERRTHRESH) value.
2. The motor is now at home position (HOME.P) value.

Homing Mode 8 Example

Move in the positive direction towards a hard stop and limit the current to 1 amp. Allow 30 degrees of error before considered at the hard stop.

1. Select Mode 8 from the drop down box.
2. Set the Direction to Positive, Position Lag to 30 degrees and Peak Current to 1.
3. When Start is selected, the motor will move to the hard stop with a peak current of 1 amp.
4. As the position error exceeds the 30 degrees, the home Position (HOME.P) is set.

Homing Mode 9: Move Until Position Error Exceeded then Find Zero Angle

The sequence in this homing mode is as follows:

1. As this home move is initiated, the motor will move according to the Direction (HOME.DIR) value until the position error exceeds the Position Lag (HOME.PERRTHRESH) value.
2. The home value is immediately found by the drive and the actual and the command position of the drive will be set to the Position (HOME.P) value plus the distance to the mechanical zero angle of the feedback device according to the current direction.
3. The motor moves to the home Position (HOME.P) plus the distance move offset is applied (if present) value, which is located at the mechanical zero-angle of the feedback.

Homing Mode 9 Example

Move in the positive direction towards a hard stop and limit the current to 1 amp. Allow 30 degrees of error before considered at the hard stop. Then move to the zero angle of the motor and designate this as the 180 point.

1. Select Mode 9 from the drop down box.
2. Set the Direction to Positive, Position to 180, Position Lag to 30 degrees and Peak Current to 1.
3. When Start is selected, the motor will move to the hard stop with a peak current of 1 amp. As the position error exceeds the 30 degrees, the home Position (HOME.P) is set and the Position will be set to 180.

12.1.2.3 Homing Mode 10: Move Until Position Error Exceeded then Find Index

This method is similar to HOME.MODE 8, but looks for the index pulse after it encounters the hard stop. For this method, the motor will move until it encounters a hard stop, causing the position error to exceed a specific threshold that you set. Once the threshold is exceeded, the motion will reverse and look for an index pulse.

This method can only be used with feedback devices that have an index pulse such as incremental encoders and analog sine encoders with an Index channel (Feedback selection 10, 11, 20, 21). This method requires that Capture Mode is turned on in the home screen. With Mode 10 selected, a **Set Capture** button appears (see arrow below). Click **Set Capture** to set the position capture mechanism correctly for proper homing with an index pulse.

1. Motor advances into the mechanical stop and then reverses direction.
2. The motor is searching for the index pulse during this time.
3. If the motor finds the Index pulse, then the Home-position is found.
4. The actual and the command position of the drive will be set to the HOME.P value as soon as the index signal is found. The drive then ramps down to velocity 0.
5. If another mechanical stop is found before the Index signal, then the homing sequence will fail and the system needs to be reviewed for proper wiring.

Homing Mode 10 Example: Move Until Position Error Exceeded then Find Index.

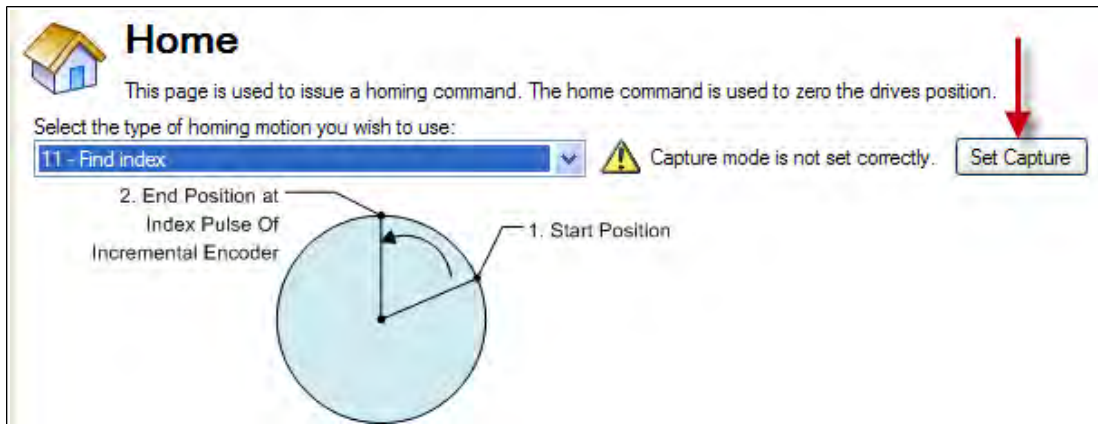
1. Select mode 10 from the drop down box.
2. Set the direction to positive.
3. In the home screen click **Set Capture**.
4. Set position lag and peak current values based on your application requirements.

5. When start is selected, the motor moves in the positive direction until a hard stop is encountered.
6. The motor reverses and moves until it encounters the index reference and then it stops.
7. If another hard stop is encountered before the index reference, home fails.

Unlike Homing Mode 3, the Home position is set as soon as the index pulse is found, irrespective of the direction of motion.

Homing Mode 11: Find Index Signal

This method can only be used with feedback devices that have an index pulse such as incremental encoders and analog sine encoders with an Index channel (Feedback selection 10, 11, 20, 21). This method requires that Capture Mode is turned on in the home screen. With Mode 11 selected, a **Set Capture** button appears (see arrow below). Click **Set Capture** to set the position capture mechanism correctly for proper homing with an index pulse.



Once homing is triggered, the homing routine is performed as follows:

1. The motor starts to move according to the HOME.DIR setting.
2. The Motor is searching for the index pulse during this time.
3. If the Motor finds the Index pulse, then the Home-position is found.
4. The actual and the command position of the Drive will be set to the HOME.P value as soon as the index-signal is found. The Drive then ramps down to velocity 0.
5. If the Limit switch is active before Index signal, then the Motor changes the direction and then repeats steps 3 and 4.

Homing Mode 11 Example: Find Index Signal.

1. Select Mode 11 from the drop down box.
2. Set the Direction to Positive.
3. In the home screen press the "set capture" button.
4. When Start is selected, the motor will move until it encounters the index reference and then it stops.
5. If Limit switch is encountered before Index reference, the Motor changes direction and searches for Index signal in the opposite direction.

Unlike Homing Mode 3, the Home position is set as soon as the index pulse is found, irrespective of the direction of motion.

Homing Mode 12: Homing to a home-switch, including mechanical stop detection

This homing mode starts a motion until a digital input, which is assigned to act as a home switch, has been activated. The home switch must be activated according to the setting of the HOME.DIR setting. The home position is found as soon as the home-switch was activated during a motion in direction of the HOME.DIR setting.

The sequence of this homing mode is as follows:

1. The motor starts to move according to the HOME.DIR setting.
2. The home position has been found as soon as the home-switch becomes active during a motion in direction of the HOME.DIR setting. The actual- and the command position of the Drive will immediately be set to the HOME.P value and the motor ramps down to velocity 0.

This homing mode is similar to the homing mode 4, but checks if the motor hits a mechanical stop instead of the hardware limit switches. A mechanical stop is detected as soon as the absolute value of the position error (PL.ERR) is larger than the position error threshold (HOME.PERRTHRESH) setting. The current command value is limited to the HOME.IPEAK value during the homing process. The motor behaves as follows when a mechanical stop has been detected before the home switch was found:

1. The motor changes the direction until the home switch is crossed.
2. The motor ramps down to velocity 0 and changes afterwards the direction again after crossing the home switch.
3. The home-switch will now be activated according to the HOME.DIR setting and the home-position has been found. The actual and the command position of the drive is immediately set to the HOME.P value and the motor ramps down to velocity 0.

12.1.3 Using Homing: Advanced

The various homing methods in the AKD offer several options for setting up your home reference. When using any of the methods that use homing switches and limits, please refer to the Input/Output section for proper wiring techniques.

12.1.3.1 Related Parameters and Commands

HOME Parameters

PL.FB

"CAP0.MODE, CAP1.MODE" (=> p. 326): Sets index capture method

12.1.3.2 Related Topics

10.1 Digital Inputs and Outputs

DIN Parameters

DOUT Parameters

12.2 Motion Tasks

12.2.1 Overview

The AKD offers several options for executing moves, which are called "motion tasks" in WorkBench. You can directly link an input to a single motion task, assign inputs as a BCD pointer and execute, command over the Ethernet connection, or automatically execute a motion task as the result of the completion of another move event. Using WorkBench, you can also set up a single motion task or a sequence of motion tasks through the Motion Task screen, which is accessed from the Settings view. The Motion Task view allows you to modify a variety of parameters for a given move, including move profile, move type, velocity, distance, and acceleration limits.

12.2.2 Motion Task Input Table

Motion tasks can be easily entered and manipulated using the Motion Task Table. With this table, you can enter specific motion tasks and edit tasks, as well as insert and delete tasks, much like an Excel spreadsheet. The data table you build remains in WorkBench until you load the tasks into the drive. Once the tasks are loaded, you have access to the graphical representation of the moves (as in previous versions of WorkBench).

Motion Tasks
 Motion Tasks allow you to specify one or a sequence of predefined moves that can be started by a digital input or a terminal command.

Start Motion Task Running: Idle

| | Position [counts16bit] | Velocity [rpm] | Acceleration [rpm/s] | Deceleration [rpm/s] | Profile | Type | Following Task |
|----|------------------------|----------------|----------------------|----------------------|---------|------|----------------|
| 0 | | | | | | | |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |
| 12 | | | | | | | |
| 13 | | | | | | | |
| 14 | | | | | | | |
| 15 | | | | | | | |
| 16 | | | | | | | |
| 17 | | | | | | | |
| 18 | | | | | | | |
| 19 | | | | | | | |
| 20 | | | | | | | |

Save Motion Tasks To Drive | Reload Tasks From Drive | Import From File | Export To File

Enter values in each column for each task that you require. Once you enter an item, that field will change color indicating it has been edited but not yet loaded to the drive. Continue entering your tasks row by row. If you find that you skipped a task, you can easily insert it before or after a row that you highlight, and right-click to open the menu options. You can also select a task and using the right click access the copy and paste commands. For delays and motion blends, double click on the line you want to edit and the delay and blend options will be available to adjust. You must load any changes or modifications to the drive before they will be available for use.

12.2.3 Using Motion Tasks

Use the Motion Task view to create and execute new motion tasks within WorkBench. As you add motion tasks, the new tasks appear as branches. You can use the root Motion Task view to view all tasks at once and execute individual tasks. When you select the Motion Task view, the Motion Task table is opened, as shown below.

Motion Tasks
 Motion Tasks allow you to specify one or a sequence of predefined moves that can be started by a digital input or a terminal command.

Start Motion Task Running: Idle

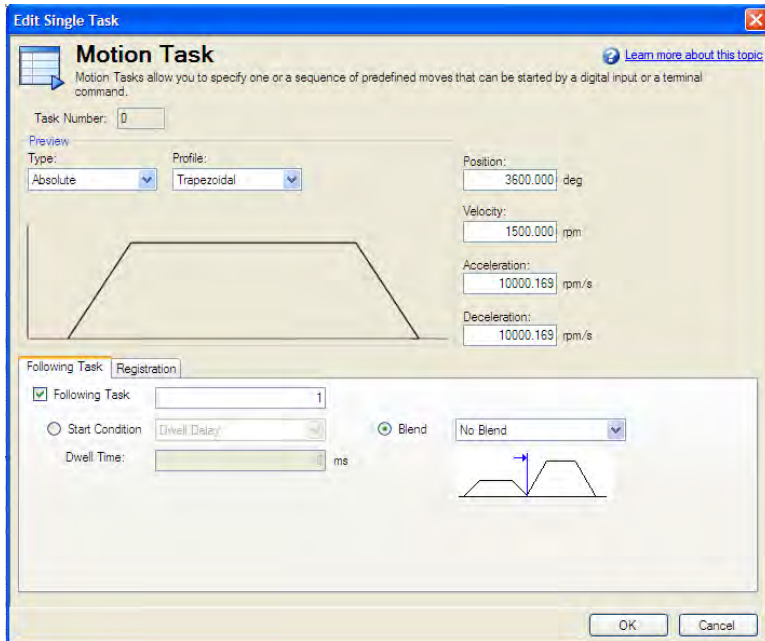
| | Position [deg] | Velocity [rpm] | Acceleration [rpm/s] | Deceleration [rpm/s] | Profile | Type | Following Task |
|---|----------------|----------------|----------------------|----------------------|-------------|-----------------------|----------------|
| 0 | 360.000 | 1500.000 | 10000.169 | 10000.169 | Trapezoidal | Absolute | 1 |
| 1 | -360.000 | 500.000 | 10000.169 | 10000.169 | Trapezoidal | Relative to Comman... | 2 |
| 2 | 0.000 | 2000.000 | 200000.032 | 200000.032 | Trapezoidal | Absolute | 0 |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |

From this view, you can perform the following actions:

| Button or Dialog Box | Description |
|-----------------------------------|---|
| Start | By highlighting an existing motion task, you can use this button to initiate that task (and subsequent tasks it may be linked to). In order to start any Motion Task, the motor must have been homed. |
| More/Less | This expands the Task table to include additional information regarding each task. |
| Save Motion Tasks To Drive | Saves the motion tasks in the table to the drive. This field is highlighted in pink if any values in the table have changed and have not been saved to the drive |

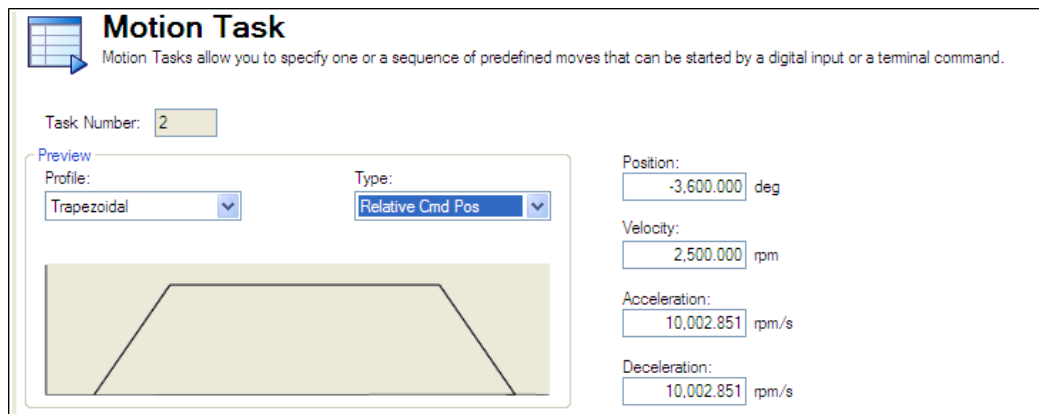
| Button or Dialog Box | Description |
|--------------------------------|--|
| Reload Tasks From Drive | Uploads motion tasks table with current values from the drive. This action deletes any values listed in the table. |
| Import From file | Allows import of a .xml file to the task table. |
| Export To File | Exports to a .xml file the current list of tasks in the table. |

All of the task parameters can be added to the Motion Tasks Table when **More** is selected. You can also view individual tasks in the Motion Task (Edit Single Task) screen by double-clicking on the task row.



Once in the edit screen, you can adjust move type, position command, velocity and accelerations as well as sequencing options. The editable fields include:

- **Type:** Sets the type of move, absolute or incremental type moves.

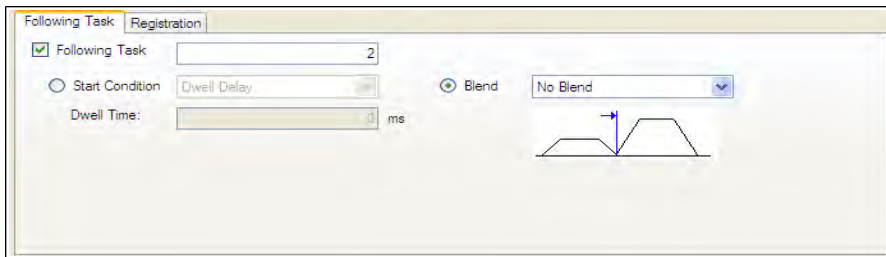


- **Profile :** This area sets the basic shape of the move. The basic move is trapezoidal, but S-curve (1:1) and custom profiles are also available using a "Profile Table".
- **Position:** This is the command position (PL.CMD), based on the type of move selected.
- **Velocity:** Sets the peak or traverse velocity depending on the move parameters
- **Acceleration:** Sets the profile acceleration ramp (this can not be set higher then the drive acceleration limit setting - DRV.ACC)

- **Deceleration:** Sets the profile deceleration ramp (this can not be set higher than the drive deceleration limit setting - DRV.DEC)

In addition to these profile settings, you can set additional parameters on how your next move will be executed. These parameters include:

- **Following Task:** By checking this box, you can indicate which task you want to follow the task that you are editing.



- **Start Condition:** This box allows for different start conditions to initiate the following task. Currently, the only option is a dwell delay. The dwell delay time can be entered in the dwell time box.
- **Blend:** It is possible to blend moves together by using the blend option. This option allows either a blend to velocity, or blend to acceleration.

Once you have entered the data, you can click on the Motion Task root directory to bring the table up to execute your moves. For more advanced motion tasks, specific constraints can be added. Currently, you can have a task that is either an interrupt task, or non-interruptible.

Advanced: Constraints

- **Interrupt:** This constraint can be used to interrupt a task that has not been finished. In other words, this constraint only will initiate the interrupt task motion if another task with no constraints is underway. This selection is a good option for registration tasks when you only want the move to happen if the current task has not finished.
- **Non-Interruptible:** When this constraint is selected, then the task can not be interrupted by another motion task or task call from inputs.

12.2.3.1 Motion Profiles

Trapezoidal moves include a fixed ramped acceleration (set by MT.ACC), a traverse period at velocity (set by MT.VEL), and a fixed ramped deceleration (set by MT.DEC). The distance traveled during the move is determined by the type of move (absolute or incremental). Position is set using MT.P. Note that for short moves, the traverse velocity may not ever be reached.

12.2.3.2 Motion Types

Motion tasks can use the following motion types:

- Absolute
- Relative to command position (PL.CMD)
- Relative to previous target position
- Relative to feedback position

Motion types define how the target position will be calculated. These can be either incremental (relative) or absolute. An incremental move is a move that increments a specific distance. The AKD allows the user to increment based on two different beginning positions as described below. Absolute moves move to a specific position based the actual position from an established zero or home point.

Absolute motion task

An absolute move type will move to the actual position indicated by MT.P. A home move would have been established to provide the reference for actual positions on the machine. In this case, the target position = $PL.CMD=MT.P$

Example:

You want to move to a position that is 68 degrees from home reference.

On the Motion Task edit page, Select the Absolute move type, and then enter 68 into the position block (units should be set in degrees). Enter the traverse speed and adjust the acceleration and deceleration parameters as needed. Once you move off the edit screen you can select this task and initiate a start. Regardless of your current position, the motor will now rotate to the absolute position of 68 degrees as referenced from the home position.

Motion task relative to command position (PL.CMD)

This type is simply an incremental move. The target position is based on the current position represented by PL.CMD plus the increment you want to move. Specifically, Target Position = $PL.CMD + MT.P$.

Example:

Your current motor position at the time you activate the motion task is 38 degrees. You want to move an increment of 30 degrees.

On the Motion Task edit page, select the **Relative to Command Position** move type, and then enter 30 into the position block (units should be set in degrees). Enter the traverse speed and adjust the acceleration and deceleration parameters as needed. Once you move off the edit screen you can select this task and initiate a start. The motor will move 30 degrees from the current position. After the motion task, the motor will now be at 68 degrees ($38 + 30 = 68$).

Motion task relative to previous target position

Using the last target position as the start point, this profile will move the increment chosen from that old position. This motion type is recommended in situations where a previous task may have been interrupted, or you want to eliminate any accumulated error. The target position will look at the previous target position and then add the increment you enter for this task. Specifically, Target Position = Previous Target Position + MT.P.

Example:

You initiated a motion task to increment 360 degrees, but that task was interrupted and the motor was stopped at 175 degrees. You use this method and increment another 360 degrees, the motor will complete the motion at 720 degrees (basically, it finished the first move to 360, and then made the additional distance requested of 360 degrees).

The task would be set up similar to the Relative command position example above. To view this in action, set up 2 tasks, once Relative Command Position and the second, Relative Old Target Position. Use low velocities so you can stop motion before the move is completed. Start the first move and then stop before it finishes. Then select the second move. The motor will stop at the desired end position if the stop had not occurred. Try this again, but do not use the Relative Old Target Position and you will see the difference.

12.2.4 Using Motion Tasks: Advanced

12.2.4.1 Joining multiple tasks

Tasks can be joined in sequence using the task setup screen. Dwells can be added to allow the following tasks to wait a specific amount of time before starting. Tasks can also be blended to provide for smoother transitions

between multiple moves. AKD allows blending with either acceleration or velocity.

12.2.4.2 Start Conditions

AKD currently provides only one alternatives to start a following task at this time (more are planned):

- **Dwell Delay.** Starts the following task after the dwell time entered (MT.FTIME)

12.2.4.3 Blending

As an alternative to dwells, the AKD can blend motion tasks to provide smooth transitions between multiple tasks. This feature extends the life of your machine life by minimizing mechanical wear due to hard move transitions (jerks).

- **No blend.** No blend, move starts upon completion of the previous task
- **Blend into Acceleration.** Blends the acceleration of the current task into the following task. As the target position of the first task is reached, the acceleration blends into the second task. This prevents the motor from decelerating to zero before starting the second move. This only works where both accelerations are driving the motor in the same direction.
- **Blend into Velocity.** Blends the velocity of the current task into the following task. In this method, the target position is reached as the velocity of the second move is reached. The blending begins prior to reaching the target position, and is completed at the target position of the first move, and the traverse velocity of the second move. This only works when both velocities are in the same direction.

Related Parameters and Commands

MT Parameters and Commands

DRV.MOTIONSTAT: Bit 0 monitors if motion tasking is active.

AIN Parameters

AOUT Parameters

DIN Parameters

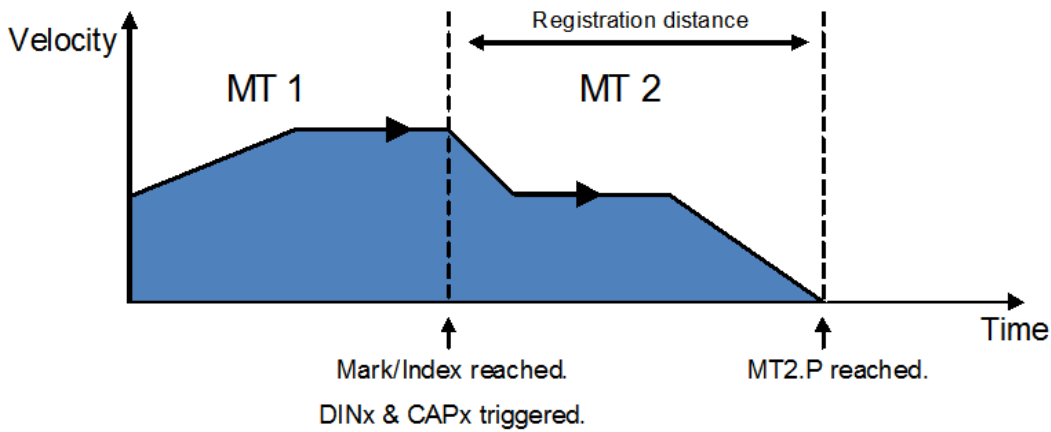
DOUT Parameters

HOME Parameters

12.3 Registration Moves (Motion Task Blending)

Registration moves are also known as "indexing on the fly" or "motion task blending". In a registration move, a digital input interrupts a running motion task and starts a new one. The start position of the new motion task is latched at the time the digital input is activated. The target position of the new motion task is calculated based on the latched position value, for a very accurate target position.

Typical applications for using the registration move are feed-to-length applications, which must guarantee proper positioning with respect to special mark or index. If this mark is reached, an external trigger signal aborts the current move and starts the registration move.



12.3.1 Configuring Registration Moves in WorkBench

You can configure registration moves from the Motion Task editor:

Following Task
Registration

Constraints: None

[Goto Digital I/O](#) [Goto Position Capture](#)

DIN Triggering This MT

DIN1 (mode 2)

Position Capture

CAP0 - DIN 1 - Rising Edge

Configure

DINx: ▼

CAPx: ▼ Apply

| Box | Description |
|-------------------------------------|---|
| Constraints | Configures bits 13 and 14 from the MT.CNTL keyword |
| DIN Triggering (read-only) | Displays the list of all digital inputs that are configured with mode 2 pointing on this motion task, or all digital inputs that have mode 4 configured. These are potentially all digital inputs that can trigger this motion task. This field is read-only, indicating current configuration. |
| Position Capture (read-only) | Displays the list of capture engine that are configured with mode 4. These are potentially all registered position that can be used for this MT. |
| Configure (write-only) | Configures a digital input and the capture engine to be use as registration source for this MT. |

12.3.2 Configuring Registration Moves from the Terminal View

You can also configure registration moves from the terminal view using drive parameters. To configure a registration move, you must configure three sets of parameters.

| | | |
|---|--|--|
| DINx MODE = 2 or 4 PARAM = z | CAPy MODE = 4 TRIGGER = x-1 | MT (NUM = z) CNTL = 5* ACC = user def. V = user def. P = user def. |
|---|--|--|

*other options are possible (Bit 13 & 14); see 12.3 Registration Moves (Motion Task Blending) and the MT.CNTL parameter description.

- Digital input (DINx) : Configure DINx to mode 2 or 4 so that input x will trigger the registration move.
- Capture (CAPx): Either capture engine 0 or 1 may be used to trigger a registration move. Set CAPx.MODE to 4, where "x" indicates the capture engine to be used. CAPx.TRIGGER must be set to x-1, where x is the DIN used above. All other capture parameters can be set as desired (see capture parameter descriptions).
- Motion tasking (MT): A registration move requires the standard motion task parameters (ACC, DEC, V, P). It also requires that the motion control word be set as follows:

0x0005 - standard registration move (this move can be interrupted and can start regardless of previous velocity)

0x2005 - non-interruptible registration move (move must complete before any other motion task can start)

0x4005 - interruptible registration move but will not start if velocity is 0

0x6005 - non-interruptible registration move that will not start if velocity is 0

After the motion task is configured MT.SET but be used to complete the setup of the motion task.

Example

Mark triggers digital input 2.

Capture machine 0 is used.

Motion task 3 is the registration move.

Motion task 3 is only activated if a preceding motion task is active.

Motion task 3 is configured with 1,000 rpm/s² acceleration and deceleration , target velocity of 10 rpm and a relative position of 50,000 counts.

Commands:

```
DIN2.MODE 2
DIN2.PARAM 3
CAP0.MODE 4
CAP0.TRIGGER 1
MT.NUM 3
MT.CNTL 16389
MT.P 50000
MT.ACC 1000
MT.DEC 1000
MT.V 10
```

```
MT.SET
```

The registration move is now active.

12.4 Service Motion

Service Motion allows you to set up simple motion. It is typically used during initial set up to exercise the system. It can be used to help troubleshoot the system, execute tuning, make repetitive moves for verification of mechanical set up, or for other general simple motion needs. There are multiple ways to set up motion depending on the desired result in torque, velocity or position modes. In all modes, you can run a momentary pulse, set up reversing motion, or initiate continuous motion. The chart below identifies the commands available on the **Service Motion** view:

| Button or Dialog Box | Description |
|----------------------------------|--|
| Pulse | Alternates between the commanded current or velocity and zero velocity. You can specify the time period between the commanded value and the return to zero. Setting a time to zero will generate a continuous command. |
| Reversing | Alternates between the two commanded values. You can specify the time the command will be held in each state. |
| Continuous | Runs the commanded current or velocity continuously. |
| Velocity 1/Velocity 2 | Sets the two different velocities that are commanded |
| Time 1/Time 2 | Sets the time for which the different commands are generated. Setting a time to zero generates a continuous command. |
| Acceleration/Deceleration | Used to set acceleration and deceleration values when in velocity and position modes. |
| Start/Stop | Starts and stops the motion. |
| Position Feedback | Displays the present position of the motor. |
| Velocity Feedback | Displays the present velocity of the motor. |

The **Drive Motion Status** view indicates when the drive is performing service motion.

Related Topics

Drive Motion Status

Related Parameters

SM.I1
 SM.I2
 SM.I2
 SM.MOVE
 SM.T1
 SM.T2
 SM.V1
 SM.V2

12.5 Jog Move

This screen verifies that the servo system is able to command motion. Verify the settings for velocity, acceleration, and deceleration. Adjust these settings if necessary. Click **Jog** and continuous motion will occur until you click **Stop**. If motion does not occur, check for warnings, faults, or prompts from WorkBench.

If erratic motion and or vibration occur when motion is commanded, open the wizard **Tuning** screen and reduce the desired bandwidth significantly. If erratic motion continues, exit the **Setup Wizard** and use the **Performance Servo Tuner** and **Service Motion** screens in the settings tree for more advanced setup of the servo system.

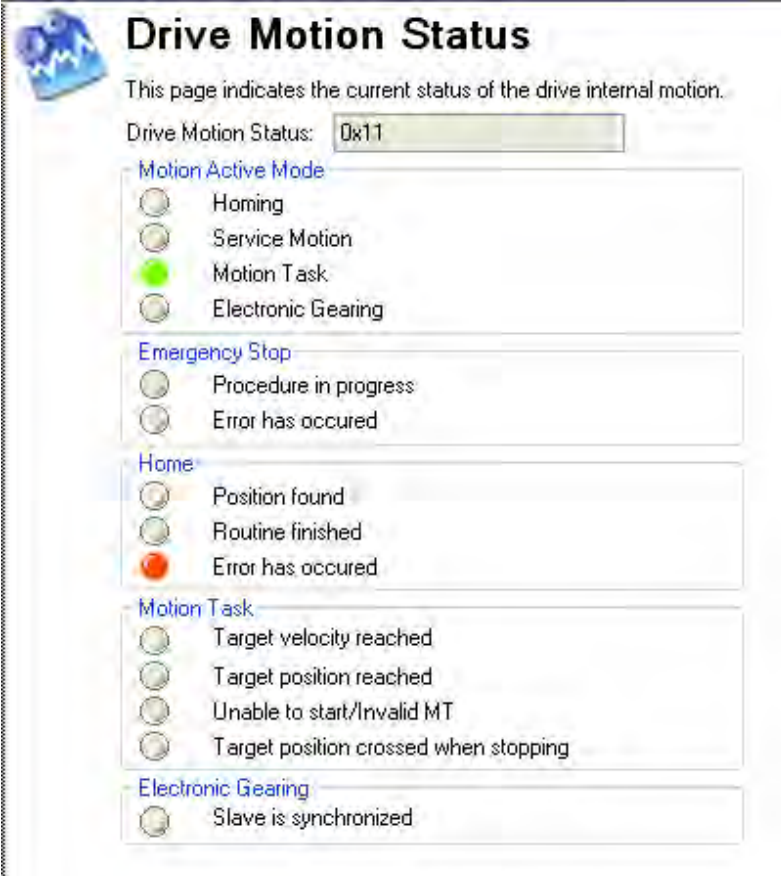
NOTE

In order to tune the system the drive must be in service-type command source and either velocity or position opmode. If the drive is in torque opmode, then a popup screen will allow you to switch into velocity mode.

12.6 Drive Motion Status

Drive Motion Status allows you to view the current status of the drive internal motion. An LED indicates the status of various possible motion states (as read by the DRV.MOTIONSTAT parameter).

The **Drive Motion Status** box displays the DRV.MOTIONSTAT output in hexadecimal form. The boxes below **Drive Motion Status** indicate the drive state. When drive motion is activated, a green LED is shown. When error occurs, a red LED is shown as indicated below:



Drive Motion Status

This page indicates the current status of the drive internal motion.

Drive Motion Status:

Motion Active Mode

- Homing
- Service Motion
- Motion Task
- Electronic Gearing

Emergency Stop

- Procedure in progress
- Error has occurred

Home

- Position found
- Routine finished
- Error has occurred

Motion Task

- Target velocity reached
- Target position reached
- Unable to start/Invalid MT
- Target position crossed when stopping

Electronic Gearing

- Slave is synchronized

Related Topics

Homing

Service Motion

Motion Tasks

Electronic Gearing

Emergency Stop

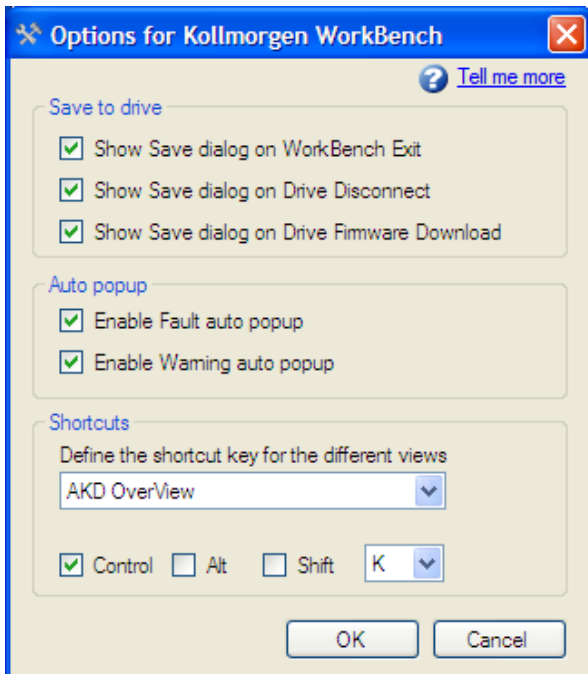
DRV.MOTIONSTAT

13 Saving Your Drive Configuration

| | |
|-------------------------------------|-----|
| 13.1 Save Options..... | 156 |
| 13.2 Save On Exit..... | 156 |
| 13.3 Save On Disconnect..... | 157 |
| 13.4 Save On Firmware Download..... | 158 |

13.1 Save Options

WorkBench offers several options for saving your drive configuration. From the WorkBench menu, if you select **Tools** and then **Options**, the following window appears:

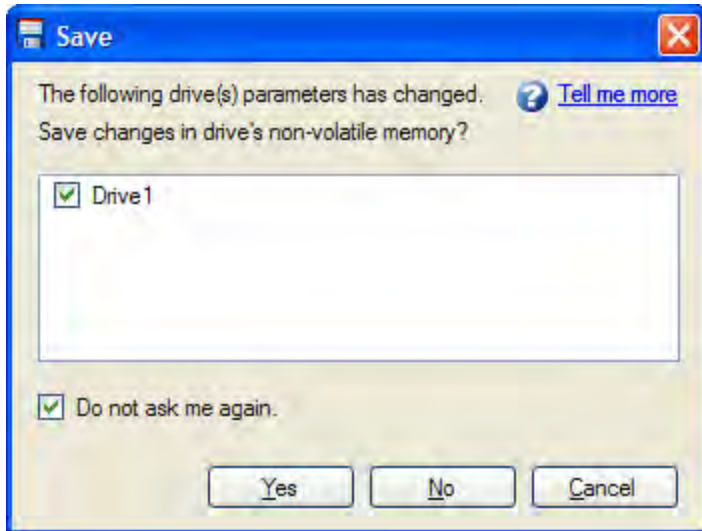


The **Save** options are applied as follows:

| Button or Dialog Box | Description |
|---|---|
| Show Save dialog on WorkBench exit | If this box is checked, then every time you exit WorkBench a dialog will ask you if you wish to save the drive parameters to nonvolatile memory. If this box is not checked, the dialog will not be shown. |
| Show Save dialog on Drive Disconnect | If this box is checked, then every time you disconnect from a drive a dialog will ask you if you wish to save the drive's parameters to nonvolatile memory. If this box is not checked the dialog will not be shown. |
| Show Save dialog on Drive Firmware Download | If this box is checked and any parameter is changed, then every time you download a firmware to a drive, a dialog will ask you if you wish to save the drive's parameters to nonvolatile memory. If this box is not checked the dialog will not be shown |

13.2 Save On Exit

When you exit WorkBench while you are connected to a drive, you may see this dialog box:



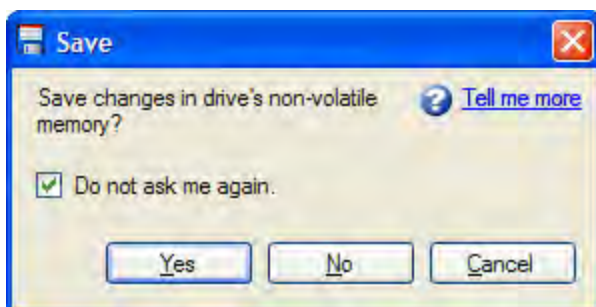
As you are working with a drive all the changes you make are stored in the volatile memory within the drive. If you power cycle the drive or loose power to the drive then any changes you have made to the drive would be lost. You can save the drive parameters to nonvolatile memory at any time and these saved parameters will be restored when the drive next powers up.

While you are connected to the drive WorkBench monitors all the changes you make to the drives parameters. An asterisk in the navigation tree shows if a parameter has been changed. If you do not change any drive parameters this dialog will not be shown.

| Button or Dialog Box | Description |
|----------------------------|--|
| Yes | Saves the parameters to the nonvolatile memory within the selected drives and then exits WorkBench. |
| No | WorkBench will exit. No drive parameters will not be saved to nonvolatile memory. |
| Cancel | This will stop the exit command and WorkBench will remain open. |
| Do not ask me again | If you check this WorkBench will not show this dialog again. There is an option in the Options dialogue to restore this setting. |

13.3 Save On Disconnect

When you disconnect from the drive, you may see this dialog box:



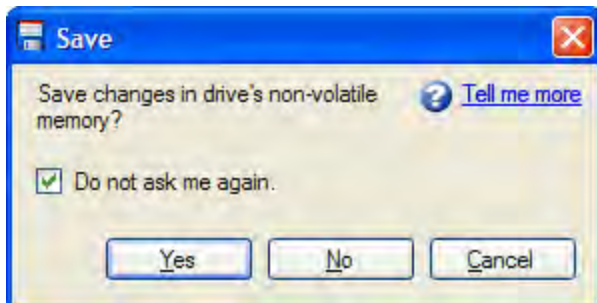
As you work with a drive, all the changes you make are stored in the volatile memory within the drive. If you restart the drive or lose power to the drive, then any changes you made to the drive would be lost. You can save the drive parameters to nonvolatile memory at any time, and these saved parameters will be restored when the drive next powers up.

While you are connected to the drive, WorkBench monitors all the changes you make to the drive parameters. An asterisk in the navigation tree shows if a parameter has been changed. If you do not change any drive parameters, this dialog box will not be shown.

| Button or Dialog Box | Description |
|----------------------------|---|
| Yes | Saves the parameters to the nonvolatile memory within the drive and then finish disconnecting. |
| No | Disconnects the drive. The drive parameters are not saved to nonvolatile memory. |
| Cancel | Stops the disconnect command. You remain connected to the drive. |
| Do not ask me again | Does not display this dialog box again. If you check this box, WorkBench does not show this dialog again. The Options dialog includes a command to restore this setting. |

13.4 Save On Firmware Download

When you download firmware to the drive, you may see this dialog box:



As you work with a drive, all the changes you make are stored in the volatile memory within the drive. If you restart the drive or lose power to the drive, then any changes you made to the drive are lost. To save these changes, you can save the drive parameters to nonvolatile memory at any time. If you save the changes to the nonvolatile memory, these saved changes will be restored when the drive next powers up.

While you are connected to the drive, WorkBench monitors all of the changes that you make to the drive parameters. An asterisk in the navigation tree indicates that a parameter has been changed. If you do not change any drive parameters, this dialog box will not be shown.

| Button or Dialog Box | Description |
|----------------------------|--|
| Yes | Saves the parameters to the nonvolatile memory within the drive and then opens dialog box for the user to select the firmware file for downloading. |
| No | The drive parameters are not saved to nonvolatile memory. It opens dialog box for the user to select the firmware file for downloading. |
| Cancel | Stops the download command. |
| Do not ask me again | Does not display this dialog box again. If you check this box, WorkBench does not show this dialog again. The Options dialog includes a command to restore this setting. |

14 Tuning Your System

| | |
|---|-----|
| 14.1 Slider Tuning..... | 160 |
| 14.2 Load Inertia..... | 160 |
| 14.3 Using the Performance Servo Tuner..... | 160 |
| 14.4 Tuning Guide..... | 189 |

Most servo systems require some level of tuning (setting up the desired response of the system usually with the load attached). This can be done through several methods available within AKD WorkBench.

Slider Tuning - Slider tuning offers a very simplistic approach to tuning and can get you underway quickly. With this method, just the proportional and integral gains are adjusted based on the desired bandwidth you wish to achieve. You can take into account your load inertia if it is known. None of the Bi-quad filters are impacted by the slider tuning.

Performance Servo Tuner - This is an easy way to get a more sophisticated tuning accomplished. The details of how the PST works are included in the advanced section of this subject. However, the PST approach is a simple one-button solution that lets the system configure all of the tuning parameters for you. The PST is a robust solution to get your system tuned and ready to operate in a wide variety of mechanical configurations and loads.

Manual Tuning - Certain applications may require manual tuning where you set the gains and filters based on the specific performance you are looking for. You may also need to "tweak" the tuning that was set up in either the slider or the PST section to optimize your application performance.

14.1 Slider Tuning

This view allows you to vary the tuning of your drive using the slider.

14.1.0.1 Gentle, Medium, and Stiff

These buttons select three of the most common bandwidths:

- Gentle works in all but the most challenging situations.
- Medium is the default and works in most situations.
- Stiff works for unloaded motors.

14.1.0.2 The Slider

As you drag the slider to the right, the stiffness increases. In many situations, you cannot drag the slider fully to the right side because the system will become unstable.

14.1.0.3 Inertia Ratio

If you know the inertia ratio of your load, then entering it can improve the performance of your system. If you do not know the inertia of your load then WorkBench will assume a ratio of 1:1, which will give good performance in many configurations. The inertia ratio is the ratio of your load with respect to the inertia of your motor.

14.2 Load Inertia

14.3 Using the Performance Servo Tuner

14.3.1 Overview

The Performance Servo Tuner (PST) tunes your system quickly and easily. The advanced technology in the PST achieves high performance and stability for both simple and complicated loads. The PST can work in a "one button" mode that requires no decisions from the user. It can also be set up in specific modes to control how the PST operates for specific requirements. Finally, the PST collects frequency response data (a Bode plot) that can be used for advanced analysis.

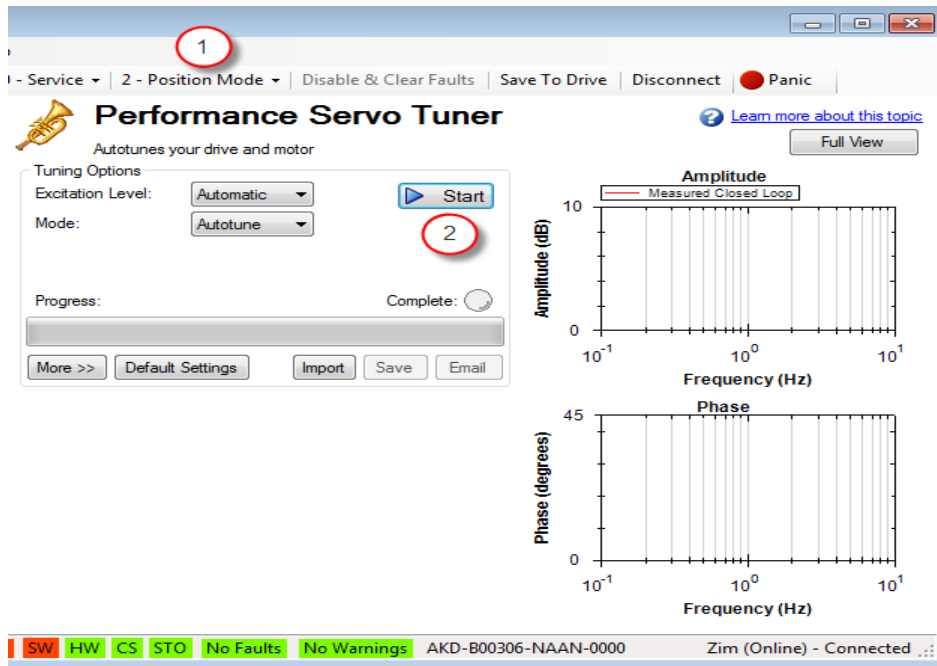
NOTE

The PST will not work reliably on a vertical axis since the motor may rest against an end stop; in this case, the inertia will not always be correctly identified.

14.3.2 Using the PST

The PST defaults to a “one button” mode, in which the PST is completely automatic after pushing the start button. In the **Settings** select your desired operation mode, navigate to the **Performance Servo Tuner** view, and then tune your system as follows:

1. Select whether you would like the drive to be tuned in **1-Velocity** or **2-Position** Mode. If the drive is in torque mode, the PST will tune in position mode by default. This is set by using the tool bar and changing the mode to either velocity or position. The drive must be disabled in order to change the operation mode.
2. Click **Start**.



The PST will then perform several tests and display results as shown below. The progress bar (1) shows the relative progress of the PST, so you can estimate when the tuning will be finished. When the tuning is complete, the green **Complete** LED (2) illuminates, and a Bode plot (3) is displayed showing the frequency response of the tuned system.

Performance Servo Tuner
Autotunes your drive and motor

Tuning Options
Excitation Level:
Mode:

Progress: Complete 1 Complete:

Autotuner | Measurement | Plots | Cursors

Current Amplitude: 0.095 Measurement:
Velocity Amplitude: 3.063 Injection Point:
Smoothing %: 1.000 Excitation Type:
FFT Points: 4096 Velocity Max: 1000.00 (RPM)
Number Points: 60000 Excite Gap: 2

Amplitude 3
Amplitude (dB) vs Frequency (Hz) plot showing Measured Closed Loop response.

Phase
Phase (degrees) vs Frequency (Hz) plot showing Measured Closed Loop response.

14.3.2.1 Saving and Emailing Bode Plots

To save screenshots and raw data of a Bode plot, click on either **Save** (1), or **Email** (2).

Tuning Options
Excitation Level:
Mode:

Progress: Complete Complete:

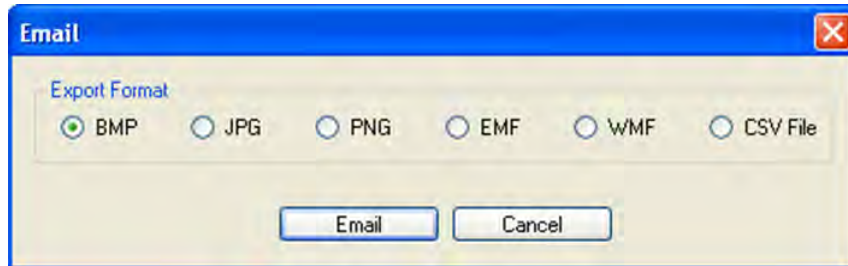
1 2

Clicking **Save** opens a prompt to select how to save the screenshot or data. Selecting BMP, JPG, PNG, EMF, or WMF saves the Bode plot as an image. Choosing CSV saves the raw data that is currently plotted as a comma delimited file. Click **Save** to save the file to your hard drive in the desired format.

Save Image/CSV

Export Format
 BMP JPG PNG EMF WMF CSV File

Clicking **Email** opens a similar prompt. Select the file format in which you wish to save the image or raw data, and an email will be created for you with the file automatically attached for your convenience. Click **Email** to create the email with the selected file attachment.



14.3.2.2 Importing a Frequency Response

If you have previously saved a frequency response measurement into a CSV file, it can be imported for later viewing. Click the **Import** button, and browse to your saved CSV file. You can import while in offline mode for convenience. Importing a frequency response is useful for developers off-site to analyze a machine tool.

14.3.3 Measurement Options

By default, the PST determines the excitation level automatically and autotunes the drive and motor.

The PST also allows you to enter a manual excitation level or to take only Bode measurements (without autotuning the system).

14.3.3.1 Using Manual Excitation Levels

By default, the PST is set to use the automatic excitation level. To obtain the automatic excitation level, the PST runs a friction test at the beginning to determine how much excitation is needed to break friction and get an accurate measurement.

To change this excitation level, click on the **Excitation Level** drop down box (1), and select **Manual**. Then enter a new **Current Amplitude** (2) in amps.

Note: If the **Injection Point** is set to **Current**, then the **Current Amplitude** box will be enabled to enter an excitation level; if the **Injection Point** is set to **Velocity**, the **Velocity Amplitude** box will be enabled to enter an excitation level.



Performance Servo Tuner

Autotunes your drive and motor

[Learn more about this topic](#)

Full View

Tuning Options

Excitation Level: Manual 1 Start

Mode: Autotune

Progress: Complete Complete:

Less << Default Settings Import Save Email

Autotuner Measure 2 Plots Cursors

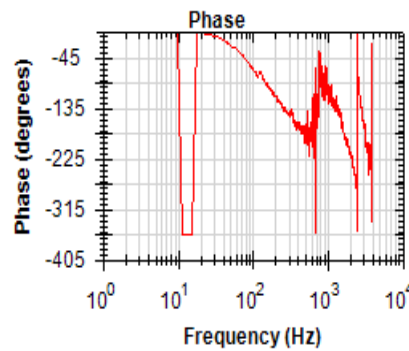
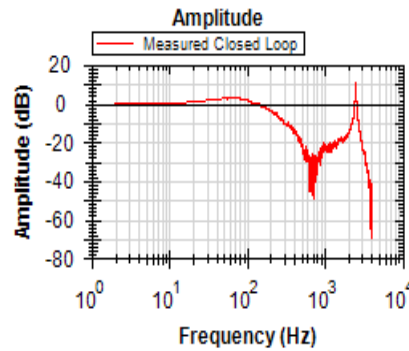
Current Amplitude: 0.095 Measurement: Closed Loop

Velocity Amplitude: 3.063 Injection Point: Current

Smoothing %: 1.000 Excitation Type: PRB

FFT Points: 4096 Velocity Max: 1000.00 (RPM)

Number Points: 60000 Excite Gap: 2



SW HW CS STO No Faults No Warnings AKD-B00306-NAAN-0000 Zim (Onl... ..)

14.3.3.2 Taking a Bode Measurement without the PST

You may wish to take only the frequency response of a system, rather than using the PST. To take a frequency response measurement without the PST, click on the **Mode** dropdown box (1) and select **Bode Plot**, then click **Start** (2).

Tuning Options

Excitation Level: Manual Start

Mode: Bode Plot 1 2

Progress: Complete Complete:

Less << Default Settings Import Save Email

14.3.4 Using the Performance Servo Tuner: Advanced

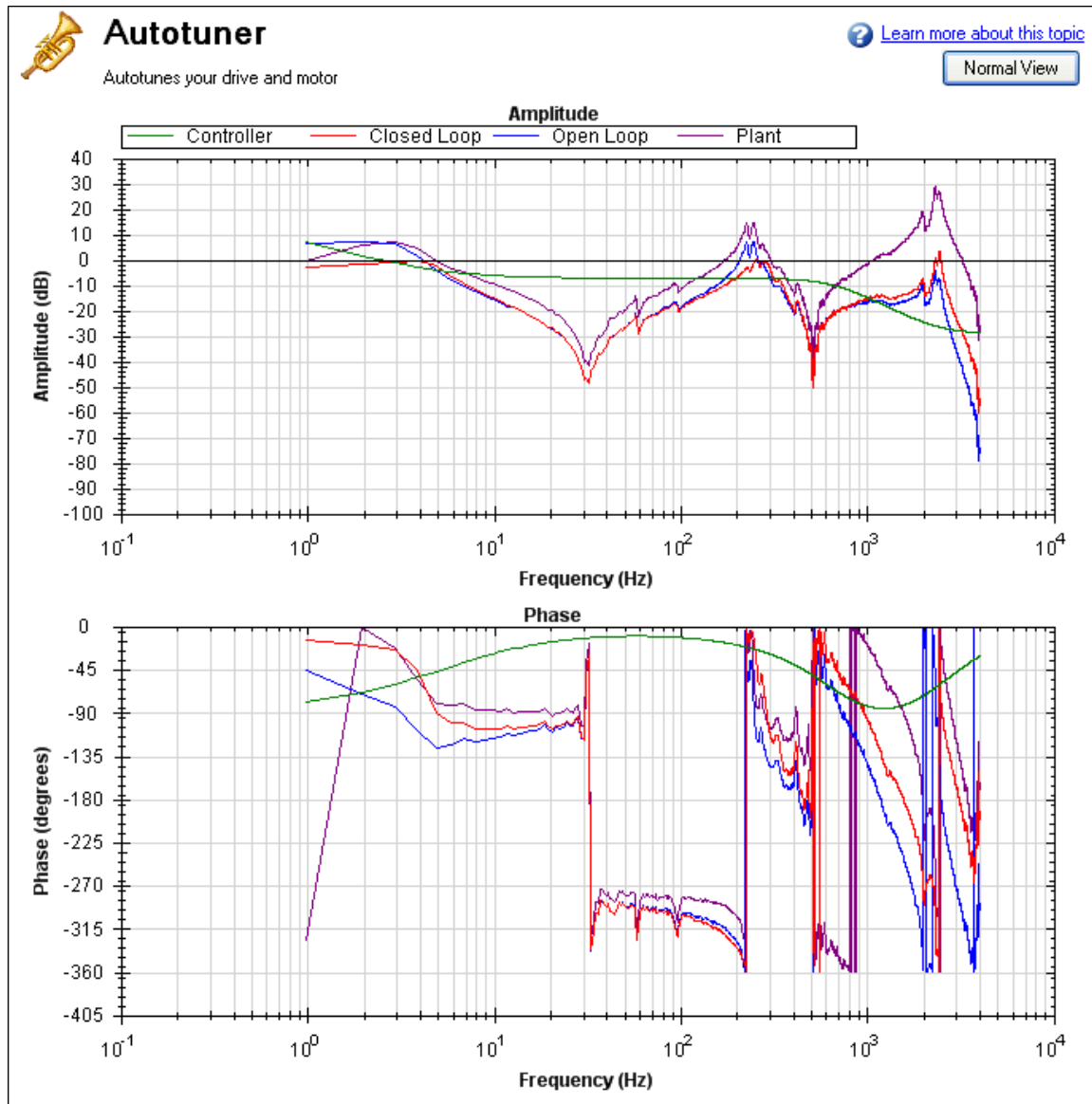
The Performance Servo Tuner (PST) can be set up to use specific modes or limits in tuning to provide tuning in ways you can control, while still taking advantage of the PST's ability to make decisions quickly and effectively for you.

To use the advanced modes of the PST, click the **More** button to display the additional features for advanced autotuning:

14.3.4.1 Typical Cases for Advanced PST Use

Tuning Systems with Low-Frequency Resonances

Systems with low-frequency resonances are challenging because low frequency data is difficult to measure. While the PST can tune these systems, you can expect lower system performance. If your system has a first anti-resonance of 30 Hz (pictured below), you can expect approximately 15 Hz (half the frequency of the first anti-resonance) of closed loop bandwidth.

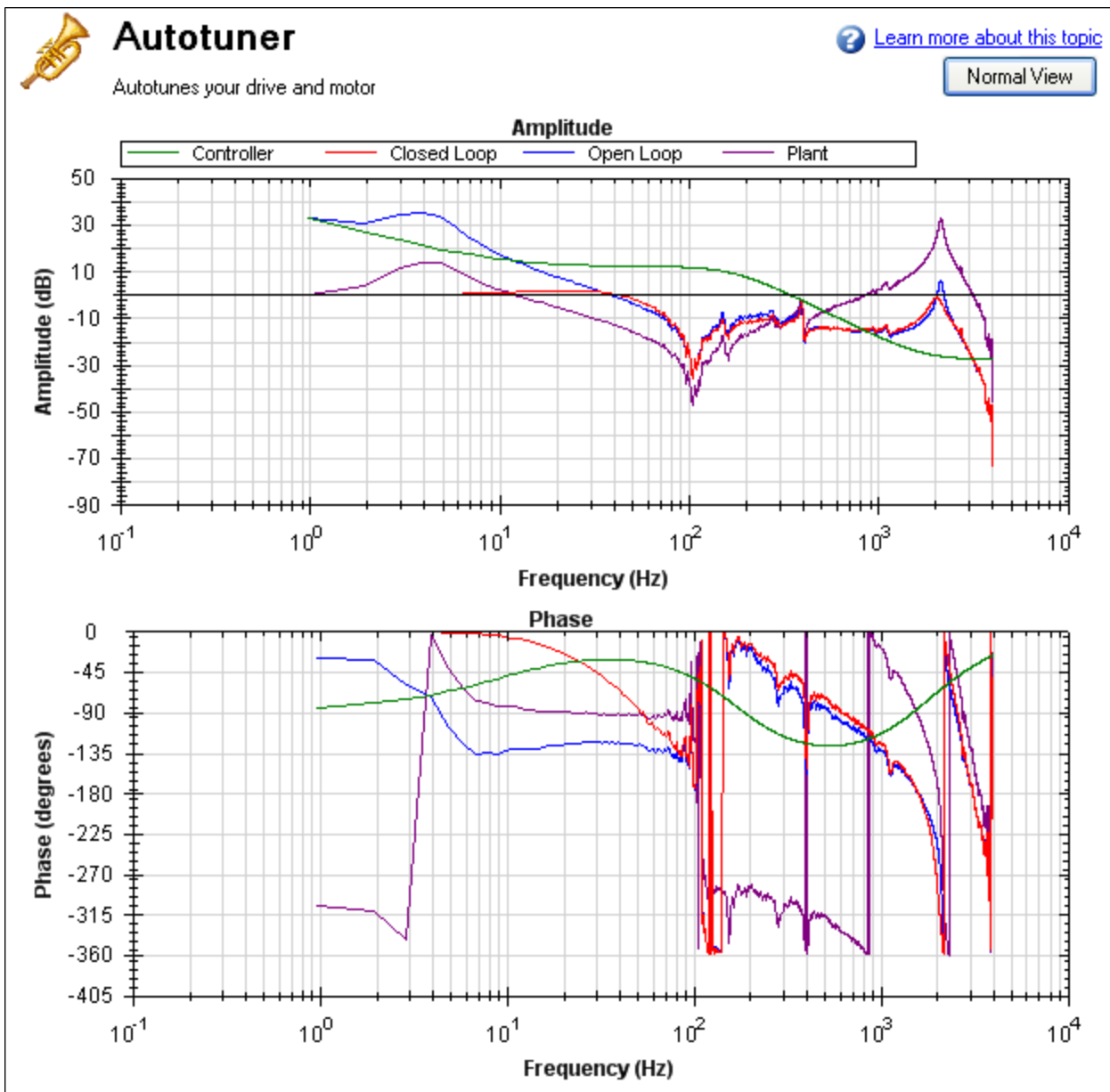


In addition, in order to accurately measure the low frequency resonances, the fast Fourier transform (FFT) resolution must be sufficiently fine to accurately measure the low-frequency resonance. A good place to start is to have an FFT resolution of 1/10 of the frequency of the lowest anti-node. In the case shown above, an anti-resonance of 30 Hz is present, so the resolution should be approximately 3 Hz FFT resolution. The PST can function with the resonance if it is accurately measured, as shown below. To adjust the FFT resolution, adjust **FFT Points** in the **Recording Options** tab as needed.

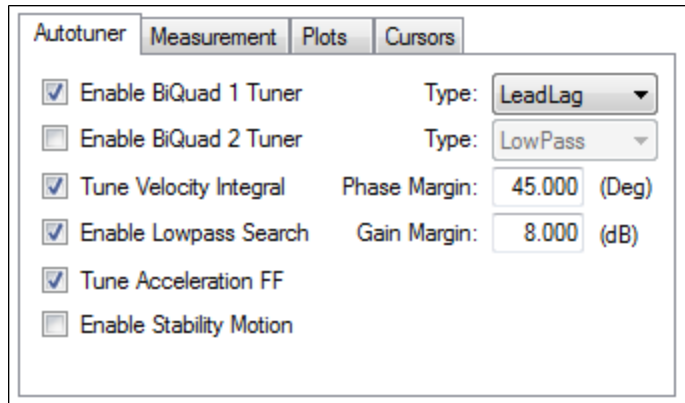
| Autotuner | Measurement | Plots | Cursors |
|---------------------------|-----------------------------|-------|---------|
| Current Amplitude: 0.095 | Measurement: Closed Loop | | |
| Velocity Amplitude: 3.063 | Injection Point: Current | | |
| Smoothing %: 1.000 | Excitation Type: PRB | | |
| FFT Points: 4096 | Velocity Max: 1000.00 (RPM) | | |
| Number Points: 60000 | Excite Gap: 2 | | |

Tuning Systems with High-Frequency Resonances

Some systems have resonances at very high frequencies (greater than 1 kHz). When the resonance is this large, it can prove a challenge in tuning, because these systems generate high noise levels that are often audible. An example of a large resonance is shown below. This example is from a steel flywheel mounted to an AKM 22E motor. The source of the resonance is the spring mass relationship between the motor rotor, shaft, and flywheel.



One way to resolve this problem is to use a low-pass filter in the feedback path. To use this filter, simply check the **Enable Lowpass Search** in the PST, which is the default behavior.

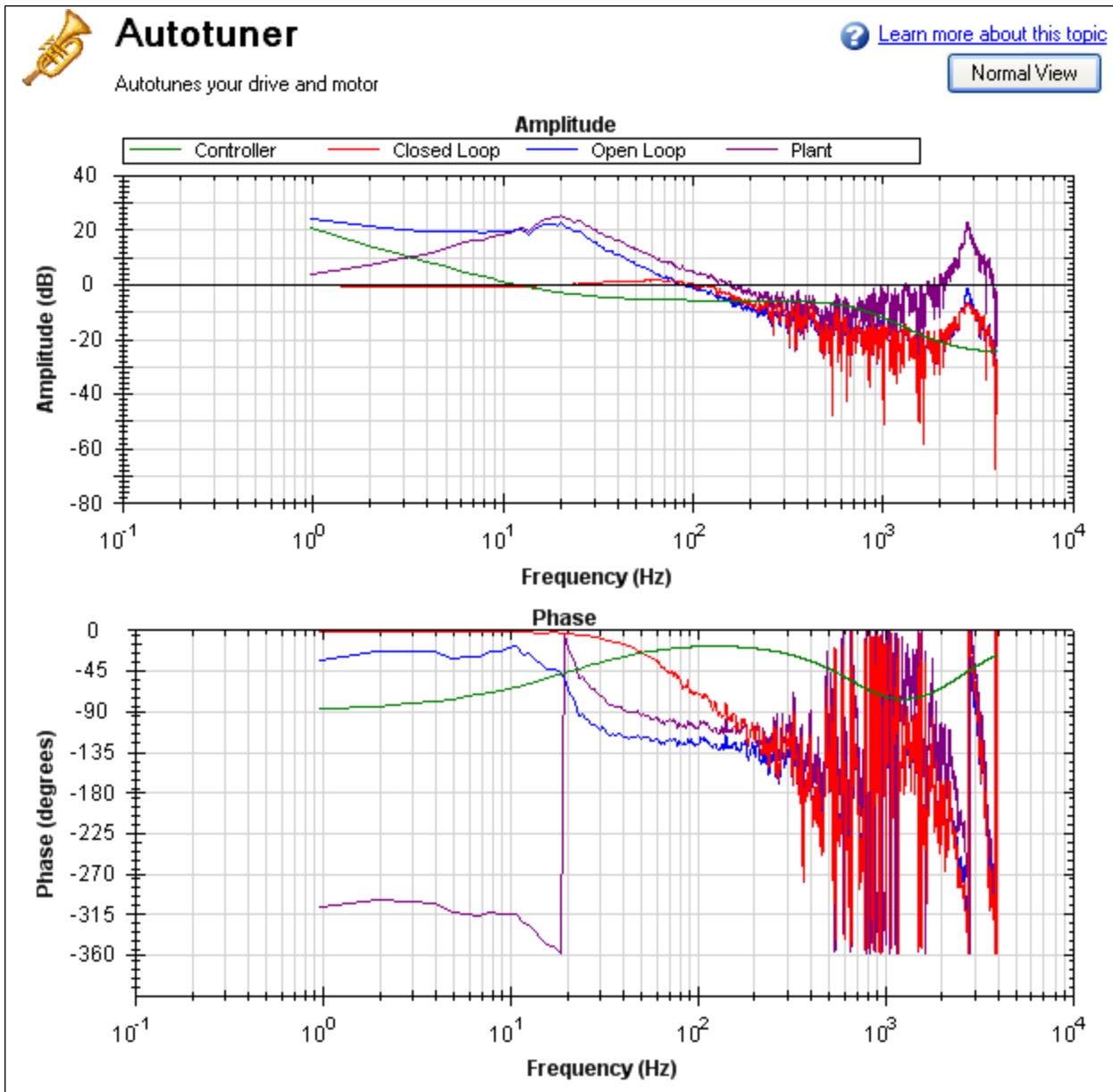


The screenshot shows the 'Autotuner' configuration window with the following settings:

| Option | Type | Value | Unit |
|--|---------------|--------|-------|
| <input checked="" type="checkbox"/> Enable BiQuad 1 Tuner | LeadLag | | |
| <input type="checkbox"/> Enable BiQuad 2 Tuner | LowPass | | |
| <input checked="" type="checkbox"/> Tune Velocity Integral | Phase Margin: | 45.000 | (Deg) |
| <input checked="" type="checkbox"/> Enable Lowpass Search | Gain Margin: | 8.000 | (dB) |
| <input checked="" type="checkbox"/> Tune Acceleration FF | | | |
| <input type="checkbox"/> Enable Stability Motion | | | |

Tuning systems with noisy frequency responses

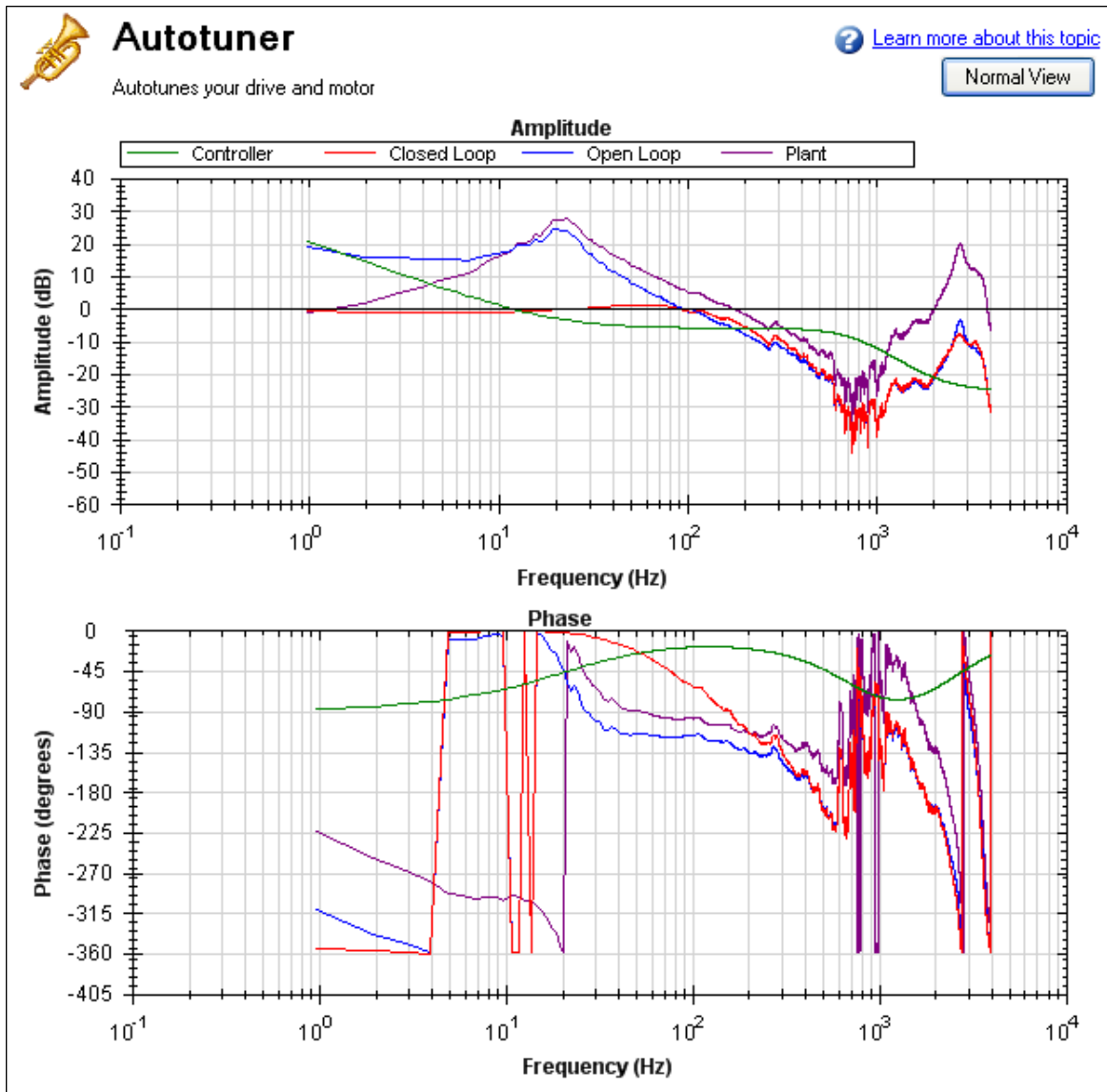
When using a motor with a low-resolution incremental encoder or resolver, the high frequency response may be noisy. Below is a Bode plot created after autotuning of an incremental encoder with 8,192 counts per revolution.



To make the Bode plot easier to read, increase the smoothing factor (1) in the advanced **Measurement Options**.

| Autotuner | | Measurement | Plots | Cursors |
|---------------------|-------|------------------|---------------|---------|
| Current Amplitude: | 0.095 | Measurement: | Closed Loop | |
| Velocity Amplitude: | 3.063 | Injection Point: | Current | |
| Smoothing %: | 4.000 | Excitation Type: | PRB | |
| FFT Points: | 4096 | Velocity Max: | 1000.00 (RPM) | |
| Number Points: | 60000 | Excite Gap: | 2 | |

After increasing the smoothing percentage, the Bode plot traces become cleaner and easier to read:



14.3.4.2 PST Options

When you click **More** in the PST view, the following options are displayed:

Autotuner **Measurement** Plots Cursors

Enable BiQuad 1 Tuner Type: **LeadLag**

Enable BiQuad 2 Tuner Type: **LowPass**

Tune Velocity Integral Phase Margin: **45.000** (Deg)

Enable Lowpass Search Gain Margin: **8.000** (dB)

Tune Acceleration FF

Enable Stability Motion

Enable BiQuad 1 Tuner

Check this box to use the first anti-resonance filter in the forward path (AR1). You can specify the type of filter to use in the **Type** box to the right of **Enable BiQuad 1 Tuner**.

Enable BiQuad 2 Tuner

Check this box to use the second anti-resonance filter in the forward path (AR2). You can specify the type of filter to use in the **Type** box to the right of **Enable BiQuad 2 Tuner**. Enabling this option may significantly slow your computer during this operation.

Biquad Type

For Biquad 1 and 2, you can choose what type of filter to implement. The four options are:

1. **LeadLag**: The LeadLag filter is the default, and will work for most servo systems.
2. **Lowpass**: A Lowpass filter requires the least amount of processing time. The PST will place the lowpass to get the maximum bandwidth possible.
3. **Resonator**: The Resonator filter is like a Notch filter with tunable bandwidth and notch depth. The Resonator takes longer to calculate than the LeadLag filter.
4. **Custom**: The Custom filter takes the longest to calculate and does not restrict the PST to a filter shape. This filter type provides excellent results, but may significantly slow your computer while the filter is calculated.

Tune Acceleration FF

This box turns on and off the acceleration feedforward tuner. If this box is checked, the PST will measure the inertia attached to the motor shaft, and using this measurement, will calculate an appropriate acceleration feedforward and write it to the drive (IL.KACFF)

Enable Stability Motion

When this checkbox is checked, after the PST has completed, the PST will command a short move in the clockwise direction, then back to its origin and monitor the motor's parameters to determine if the tuning is stable. If an instability is detected, the drive will generate Fault F133: Instability during Autotune.

Phase and Gain Margins

The PST always ensures that the tuning satisfies stability criteria that can be adjusted in units of phase margin (in degrees) and gain margin (in dB). The PST uses default values for phase and gain margin, but you can adjust these values to ensure higher stability or to allow the PST to be more aggressive by using lower gain and phase margins.

Tune Velocity Integral

Check this box to tune VL.KI (velocity loop integral gain). If this box is unchecked, the PST will set VL.KI to zero.

Enable Low Pass Search

Check this to tune a fourth-order low pass filter in the feedback path (AR 3 and 4). If this box is unchecked, the PST will not modify the anti-resonance filters in the feedback path.

14.3.4.3 Measurement Options

The PST screen also provides options for measurements:

| Autotuner | Measurement | Plots | Cursors |
|---------------------|-------------|------------------|---------------|
| Current Amplitude: | 0.095 | Measurement: | Closed Loop ▾ |
| Velocity Amplitude: | 3.063 | Injection Point: | Current ▾ |
| Smoothing %: | 1.000 | Excitation Type: | PRB ▾ |
| FFT Points: | 4096 ▾ | Velocity Max: | 1000.00 (RPM) |
| Number Points: | 60000 | Excite Gap: | 2 |

Current Amplitude

This box sets the amplitude of the current used to excite the system during a current injection mode excitation. This amplitude applies to all excitation types when the **Injection Point** is set to **Current**. The **Current Amplitude** box is disabled if the **Injection Point** is set to anything else.

Velocity Amplitude

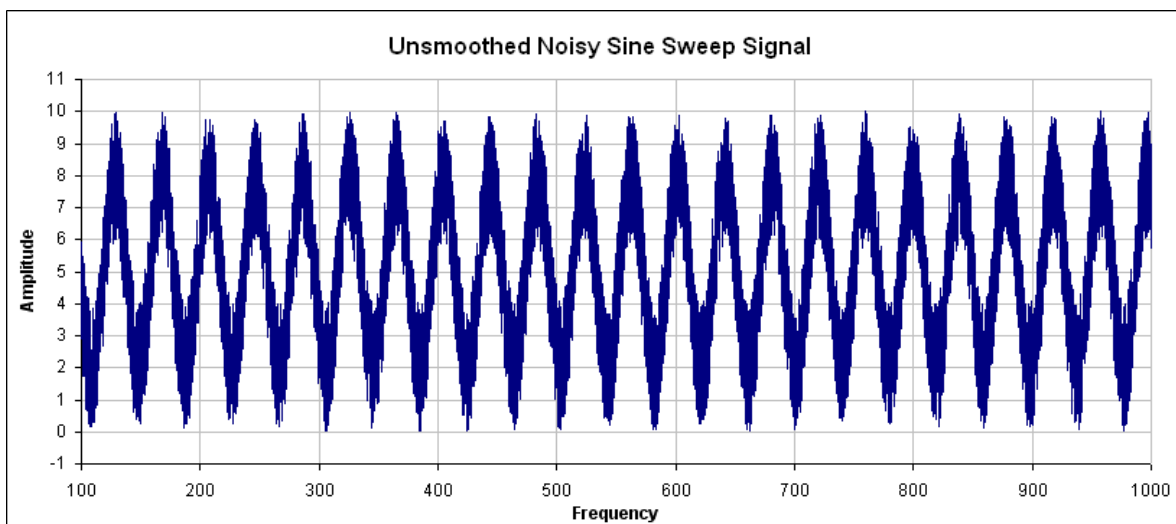
This box sets the amplitude of the velocity used to excite the system during a velocity injection mode excitation. This amplitude applies to all excitation types when the **Injection Point** is set to **Velocity**. The **Velocity Amplitude** box is disabled if the **Injection Point** is set to anything else.

Smooth %

This value applies a moving average smoothing filter to the frequency response gathered during autotuning. This process reduces noise in the frequency response that can occur when making short frequency response measurements, using low resolution encoders, conducting low amplitude frequency response tests, or for other reasons. The smoothing filter iterates through each frequency on the FFT plot. For each frequency, all frequencies within the **Smooth %** range will have their magnitudes averaged.

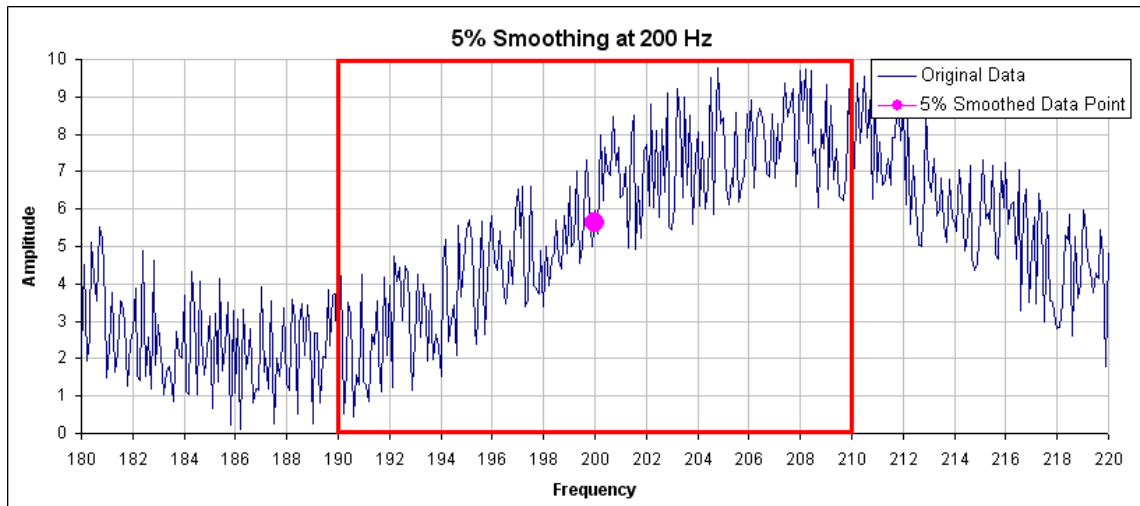
For example, if you smooth a Bode plot with 5% smoothing, at 100 Hz, it will average all the values between 95 Hz and 105 Hz; when the filter gets to 1000 Hz, the filter will average all the values between 950 Hz and 1050 Hz.

As an example, assume a noisy sine sweep signal and use a 5% smoothing factor. Below is a noisy signal with a range of 100 Hz to 1000 Hz.



In this example, examining how the smoothing filter affects a single point shows how the smoothing filter works on a full plot. If you zoom in on 200 Hz +/- 5%, this gives a range of 190 Hz – 210 Hz. The smoothing filter averages

this range of values and puts the average right on 200 Hz. The figure below shows the zoomed data around 200 Hz and the averaged value of all frequencies $\pm 5\%$ (the red box illustrates the range of frequencies being smoothed).



In the PST, the smoothing filter will do this analysis for every frequency point on the Bode plot. If the data is too noisy, then you can increase the smoothing percentage to smooth the noise out and see the underlying data patterns. A comparison of a system with 0.1% smoothing and 8% smoothing is shown below.

0.1% smoothing

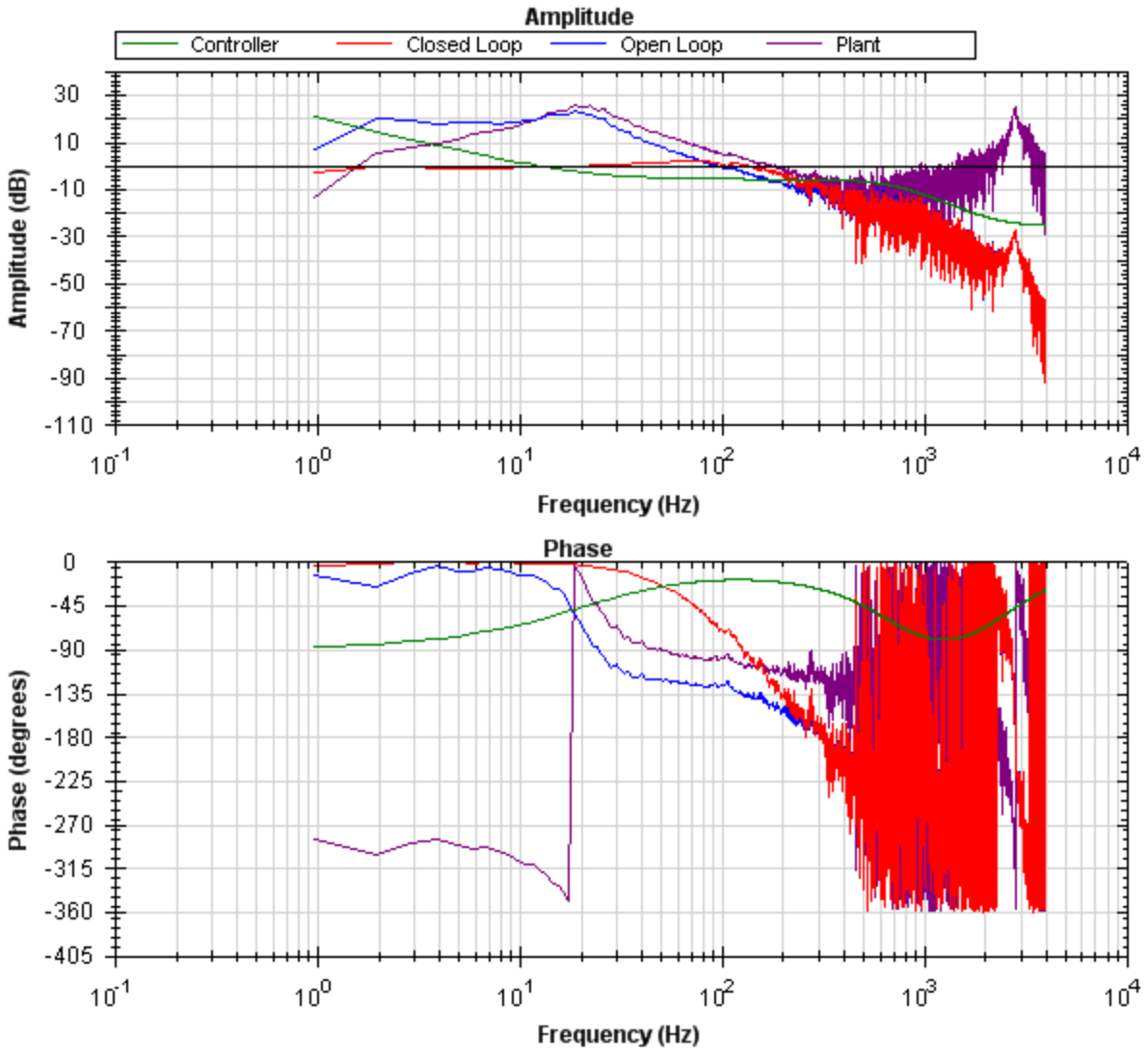


Autotuner

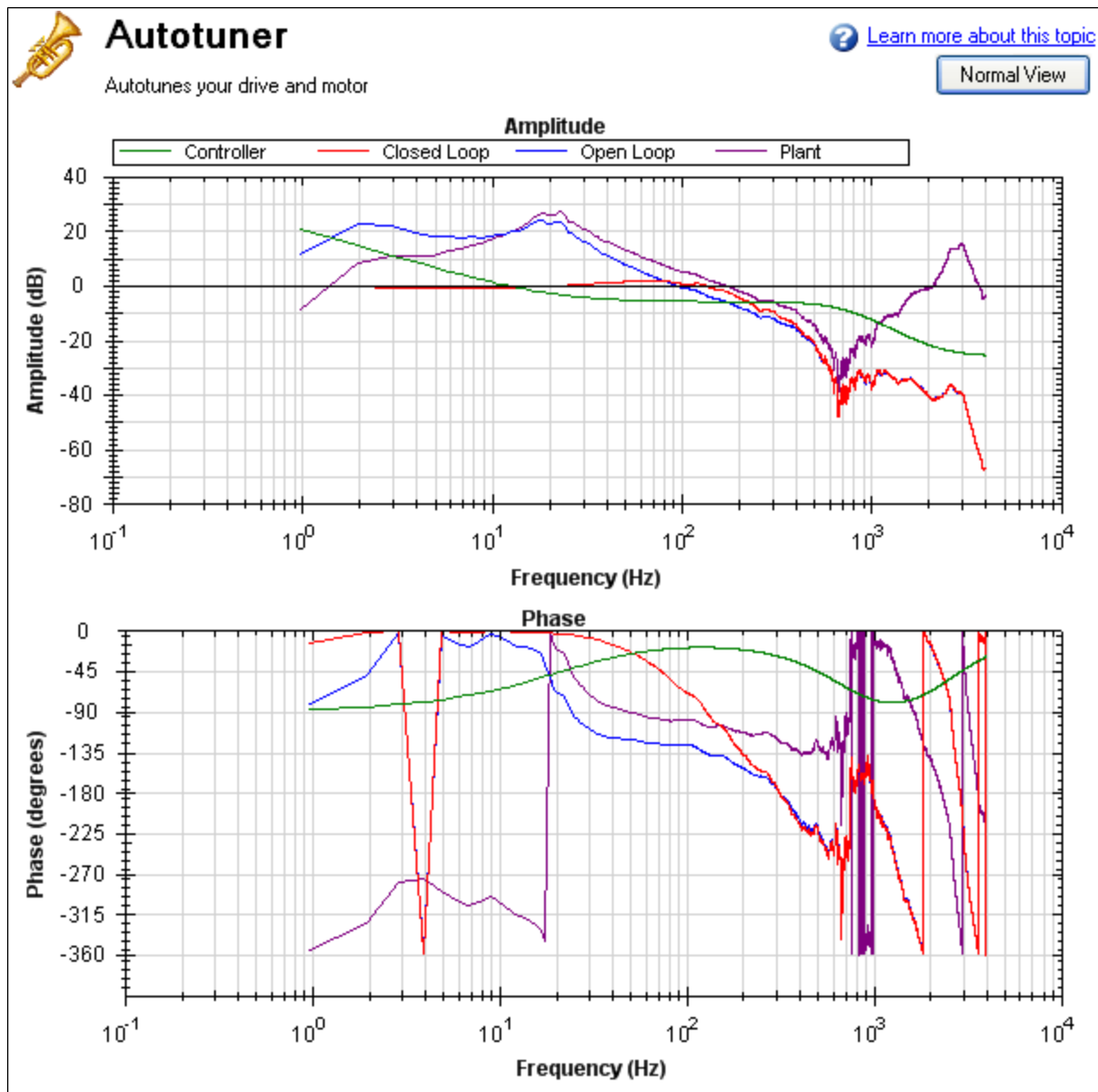
Autotunes your drive and motor

[Learn more about this topic](#)

Normal View



8% Smoothing



Note: Smoothing decreases the peaks of resonances; if smoothing is too high, a resonance may be completely hidden. If the PST cannot identify a resonance due to high smoothing, the system may become unstable.

Measurement

This box sets the measurement type used during a measurement. The PST functions only if **Plant** measurement is selected; autotune does not function in other measurement modes.

- **Closed Loop** directly measures the closed loop frequency response of the servo.
- **Plant** directly measures the plant, including drive, motor, and mechanics coupled to the motor.
- **Controller** directly measures the controller response, which includes the tuning in the velocity and position loops, and anti-resonance filters 1 & 2.

Injection Point

The **Injection Point** box sets the source location of the excitation used during autotuning. **Current** mode uses a torque disturbance at the torque output. During current injection point measurements, the excitation will use the **Current Amplitude** value to set the size of the excitation.

Velocity mode uses a velocity command to excite the system. During velocity injection point measurements, the excitation will use the **Velocity Amplitude** value to set the size of the excitation.

Excitation Type

Excitation Type box allows you to choose the type of excitation. Noise, pseudo random binary (PRB), and sine are the options available.

- **Noise** uses a pseudo random noise signal to excite the system. The signal varies between +/- current or velocity amplitude (depending on injection point). The signal contains a frequency spectrum that goes from a lower limit equal to:

$$16,000/(\text{Excite Gap} * \text{Number Points}) \text{ Hz}$$

to a higher limit equal to:

$$(16,000/\text{Excite Gap}) \text{ Hz}$$

The richness of the frequency spectrum comes from variance in the amplitude of the noise signal.

- **PRB** uses a pseudo random binary signal to excite the system. The signal is either + or – current or velocity amplitude (depending the injection point). The signal contains a frequency spectrum that goes from a lower limit equal to the larger of:

$$(16,000/(2^{\text{BODE.PRBDDEPTH}} * \text{Excite Gap})) \text{ or } 16,000/(\text{Excite Gap} * \text{Number Points}) \text{ Hz}$$

to a higher limit equal to:

$$(16,000/\text{Excite Gap}) \text{ Hz}$$

BODE.PRBDDEPTH is set to 19 by the PST. The richness of the frequency spectrum comes from variance in the phase of the signal, not the amplitude.

- **Sine** requires that you specify the start frequency, end frequency, and frequency step size. The sine sweep takes significantly longer than a noise or PRB measurement, but is often cleaner. Be careful when selecting a step size: too large of a step size may miss important resonances, and too small of a step size increases measurement time.

FFT Points

The **FFT Points** box is only visible and applicable in noise and PRB measurements. **FFT Points** sets the resolution of the FFT's measurement. The frequency resolution is equal to

$$16,000/(\text{Excite Gap} * \text{FFT Points})$$

By increasing **FFT Points**, the resolution becomes finer, but noise in the frequency response increases.

Excite Gap

The **Excite Gap** box is only visible and applicable in noise and PRB measurements. This box sets how frequently the test excitation is updated. The excite gap minimum value is 1; this value is normally set to 2 for autotuning. The excite rate is 16,000/gap. You can limit high frequency excitation by increasing the **Excite Gap** value.

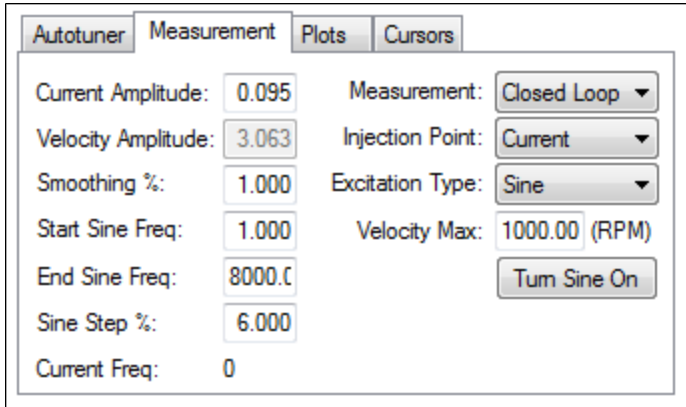
Number Points

The **Number Points** box is only visible and applicable in noise and PRB measurements. This box sets the length of recording while measuring the frequency response of the system. The measurement length is:

$$\text{Number Points} * \text{Excite Gap} / 16,000 \text{ seconds}$$

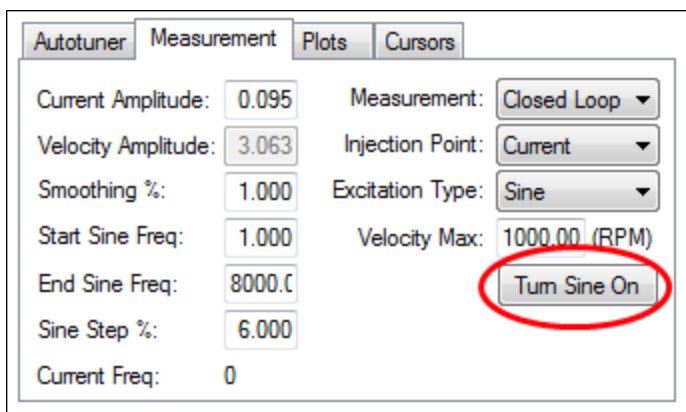
Velocity Max

The **Velocity Max** box allows the user to specify the maximum velocity the motor should be able to move while performing excitation. This box is not in effect for normal drive operation; it is only visible during the PST excitation phases. This value is implemented as soon as the PST begins, and as soon as the PST is finished, the previous overspeed threshold (VL.THRESH) is restored.



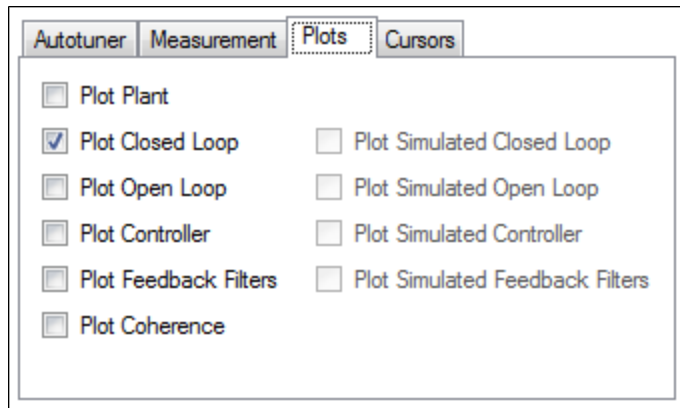
If **Excitation Type** box is set to **Sine**, different configuration options become available.

- **Start Sine Freq:** The Sine sweep test will begin at this frequency. The start frequency must be greater than zero and less than the end sine frequency. **Start Sine Freq** is only visible and applicable to Sine measurements.
- **End Sine Freq:** The Sine sweep test will end at this frequency. The end frequency must be less than or equal to 8,000, and more than the sine start frequency. **End Sine Freq** is only visible and applicable in Sine measurements
- **Sine Step %:** This box sets the sine step size. The sine sweep is discrete, not continuous. Each frequency is a multiple of the previous. For example, if the first frequency was 1 and the step size was 6%, the second frequency would be $1 * 1.06 = 1.06$ Hz, the third frequency would be $1.06 * 1.06 = 1.12$ Hz. This continues until the current frequency exceeds the **End Sine Frequency** value. **Sine Step %** is only visible and applicable in Sine measurements
- **Current Freq:** This field displays the current frequency of the sine sweep. **Current Freq** is only visible and applicable in Sine measurements
- **Turn Sine On:** This button allows the user to excite the system at a single sine frequency. When this button is pressed, it grays out boxes that do not apply. You may change the sine frequency and amplitude. To stop the sine excitation, click **Turn Sine Off**. **Turn Sine On** is only visible and applicable in Sine measurements.



NOTE When the sine excitation is used on low resolution encoders, high frequency excitation may cause less than 1 count of encoder movement. If this occurs, no movement is detected on the motor for that excitation frequency. If this occurs, a data point for that frequency will not be plotted, as this results in a calculation of 0dB for gain and -infinity for phase.

14.3.4.4 Plot Options



By default, only the measured closed loop plot is selected. You can control which of these responses are displayed on the Bode plot by checking or unchecking the **Plot Plant**, **Plot Closed Loop**, **Plot Open Loop**, **Plot Controller**, and **Plot Coherence** checkboxes shown. The options **Plot Simulated Closed Loop**, **Plot Simulated Open Loop**, **Plot Simulated Controller**, and **Plot Simulated Feedback Filters** are only available in Bode plot mode, not PST mode.

Coherence

The coherence option is only available for noise and PRB measurements; it is not available for Sine excitation measurements.

Coherence is an indicator of how accurate your data is. For example, 0 dB (1 in linear numbers) means you have perfect coherence. Another way to think of this concept is that for one unit of input, you get one unit of output. Coherence is calculated as follows:

$$Coherence = \frac{(P_{xy} \times P_{xy}^*)^2}{(P_{xx} \times P_{xx}^*) + (P_{yy} \times P_{yy}^*)}$$

where:

P_{xx} = Power Spectral Density of Input signal

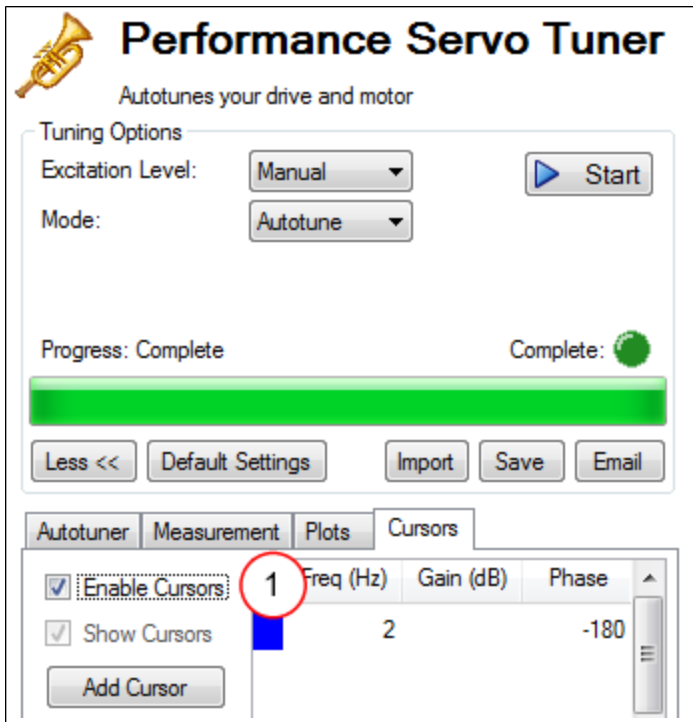
P_{yy} = Power Spectral Density of Output signal

P_{xy} = Cross Spectral Density of Input and Output

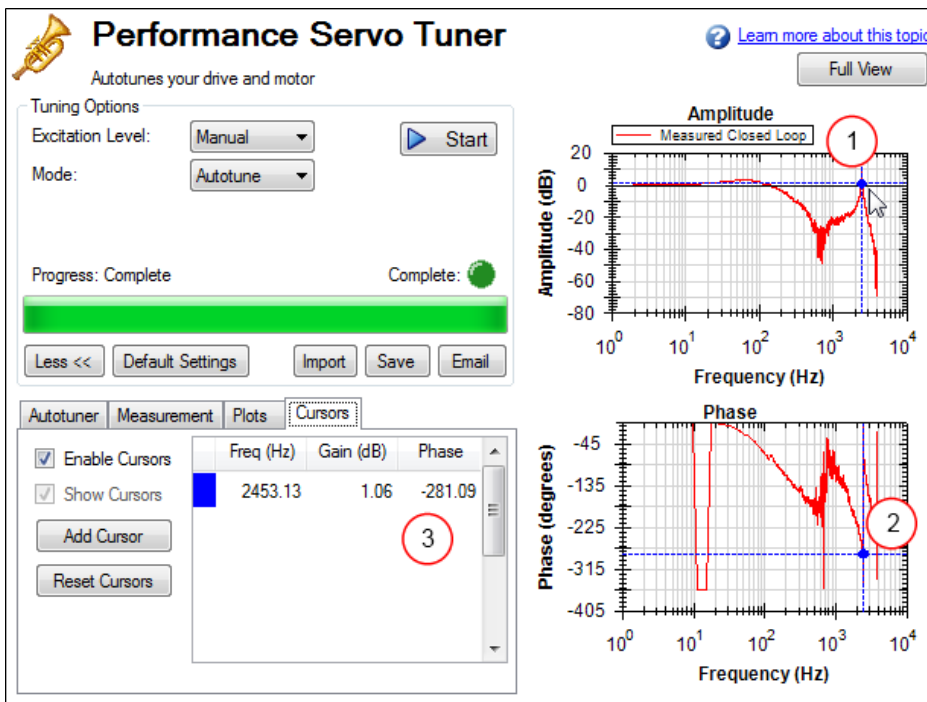
* designates complex conjugate

14.3.4.5 Cursors

Enabling cursors allows you to note specific points of interest on the Bode plot and create a table of reference points in the summary table. To enable cursors, click the **Enable Cursors** checkbox (1).



To move the cursor, move your mouse over the cursor in either the Amplitude (1), or Phase (2) plots, click and hold the left mouse button, and drag the cursor to a new location. Notice as you drag the mouse, the Frequency, Gain and Phase change in the summary window (3).



To add more cursors, click **Add Cursor**; you can add 10 cursors to the Bode Plot. When selecting a cursor, the cursor closest to the mouse will be selected. While dragging the cursor, the cursor will snap to the closest trace on the plot.

When cursors are enabled, zoom functions on the graph are disabled. To re-enable zooming, un-check **Enable Cursors**.

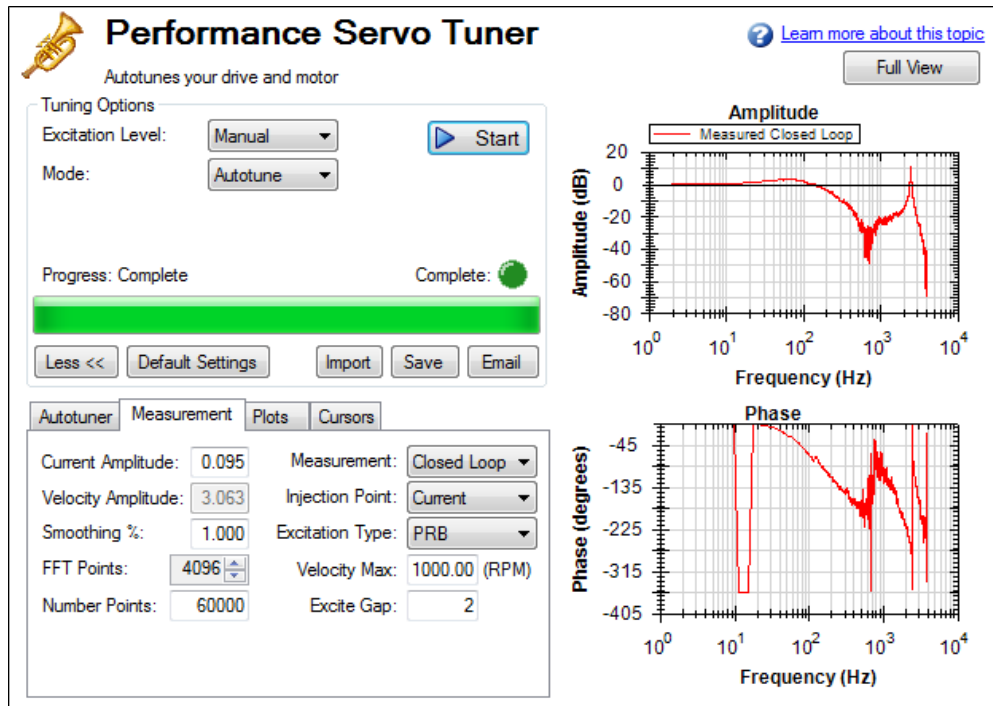
The dotted crosshair lines are only drawn for the active cursor selected; to remove all cursors from the screen, but retain their position, uncheck **Show Cursors** . To reset all cursors, click **Reset Cursors**.

Note: If a CSV file is saved or emailed after placing a cursor on the Bode plot, a cursor summary is included in the CSV raw data.

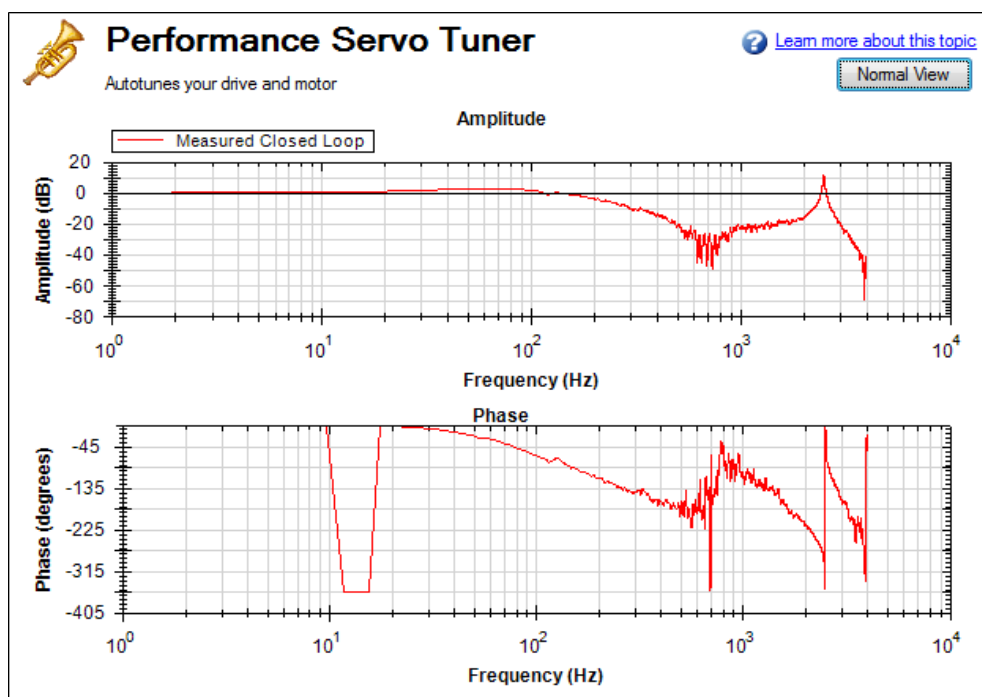
14.3.4.6 Resizing Bode Plots

In the PST view, the **Full View** and **Normal View** button (1) in the upper right of the window allows you to see the Bode Plot in greater or less detail. When viewing the Bode Plot in full view, the PST settings are hidden behind the Bode Plot. To access the PST settings, click the **Normal View** button in the upper right of the window.

Simple measurement normal view



Simple measurement full view



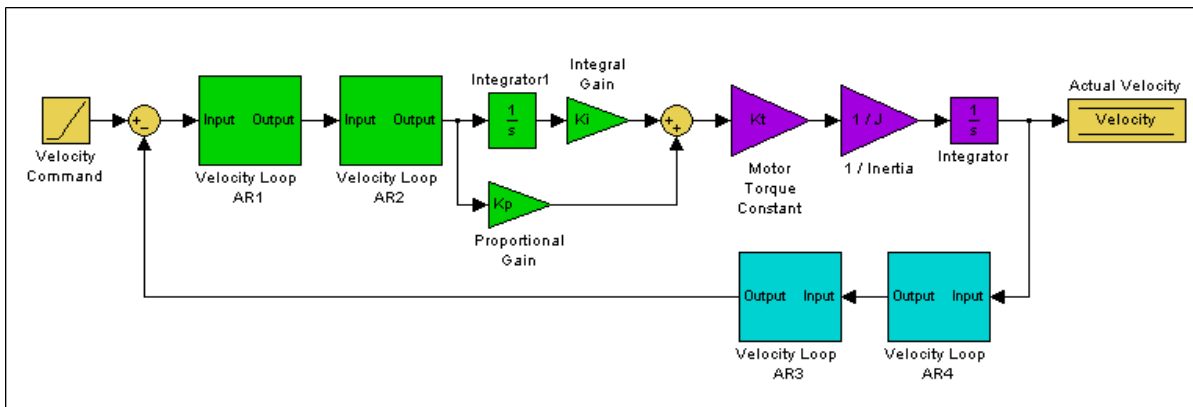
Reading and Understanding the Bode Plot

You can operate the PST without understanding how to read a Bode plot; however, understanding Bode plots will help you to use more advanced tuning techniques, which are covered more in depth in the Advanced Tuning For Velocity and Position Loops documentation.

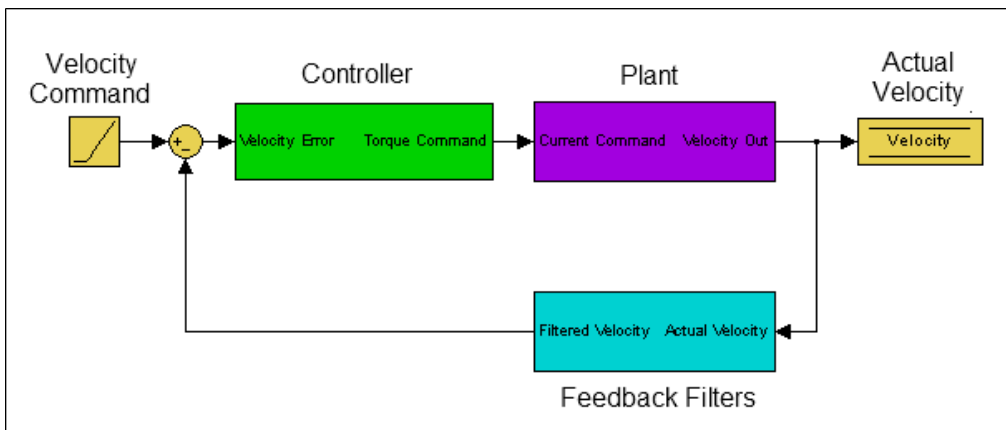
Four Bode plot traces are displayed by default:

1. Controller (green): This trace represents the frequency response of the tuning in the velocity loop and position Loop, this trace also includes anti-resonance filter 1 and 2 (also referred to as [C]).
2. Closed loop (red): This trace shows the frequency response of $G/(1 + G * H)$ where $G = C * P$, and H is the frequency response of anti-resonance filters 3 and 4.
3. Open loop (purple): This trace shows the frequency response of $G * H$, where $G = C * P$, and H is the frequency response of anti-resonance filters 3 and 4.
4. Plant: This trace shows the frequency response of the mechanics of the drive and motor (also referred to as [P])

The diagram of the velocity loop on the drive below explains the frequency response that each of these traces represents: "Tuning Guide" (=> p. 189)



These blocks can be grouped into Controller, Plant, and Feedback sections:



All of the green blocks have been grouped together to create the Controller [C]. The Controller is the portion of the control loop containing all velocity and position loop tuning, including the forward path filters.

All of the purple blocks have been combined to make the Plant [P]. The plant represents the mechanical and electrical properties of the motor, drive and any mechanical bodies attached to the motor.

The two feedback filters have been combined into one block. This value is never measured directly; however it contributes to both the Open Loop [G] and Closed Loop [T] frequency responses.

The definition of the Open Loop [G] frequency response is:

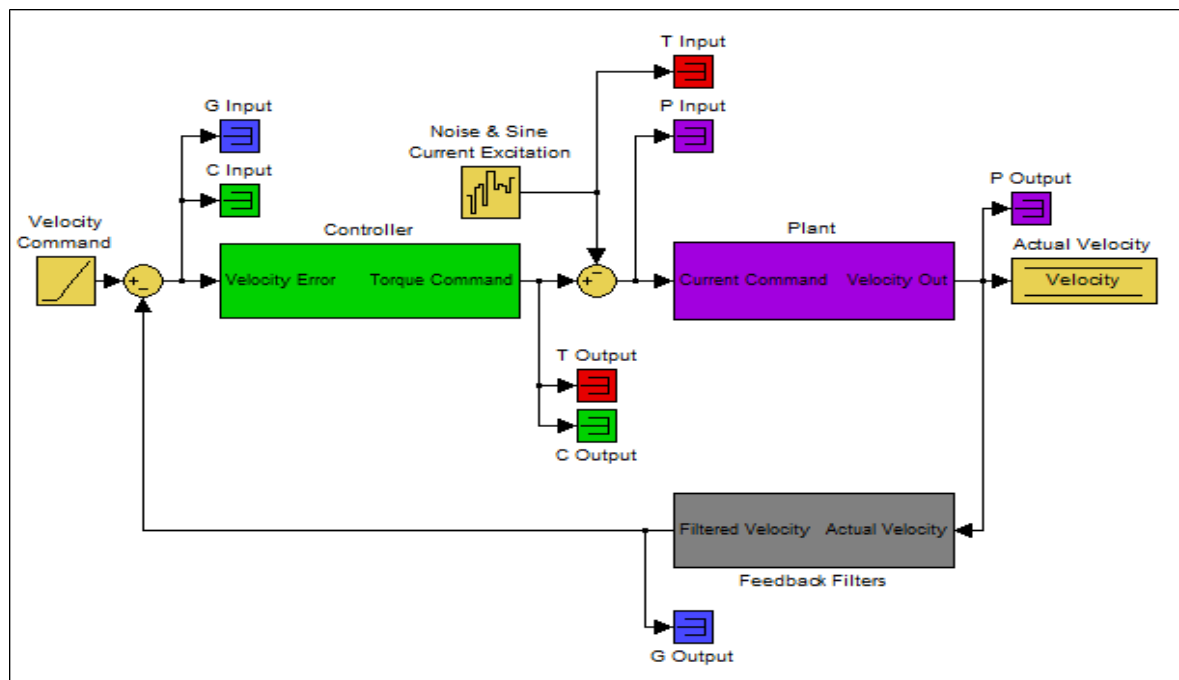
$$\text{Open Loop} = \text{Controller} \times \text{Plant} \times \text{Feedback Filters}$$

The definition of the Closed Loop [T] frequency response is:

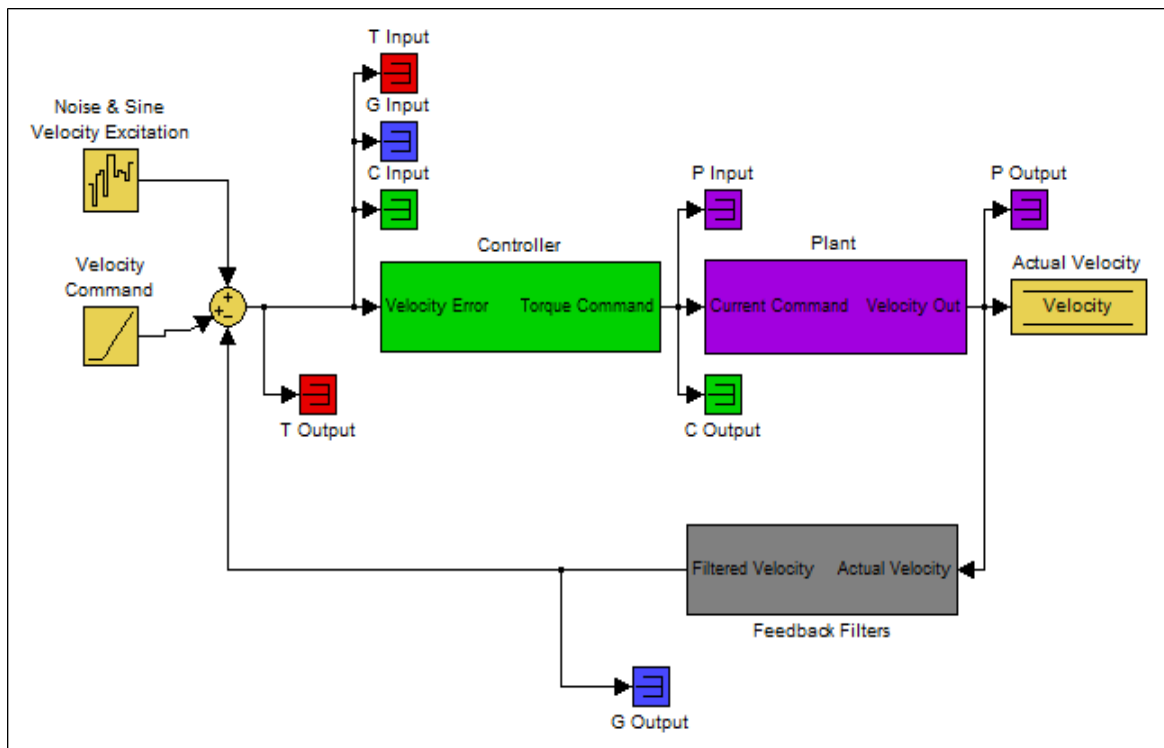
$$\text{ClosedLoop} = \frac{\text{Controller} \times \text{Plant}}{1 + \text{Controller} \times \text{Plant} \times \text{FeedbackFilters}}$$

Below is a diagram of measurement points (input and output) for each of these frequency responses. The input and output markers have been color coded with the color they appear in the PST:

Current Excitation:



Velocity Excitation:

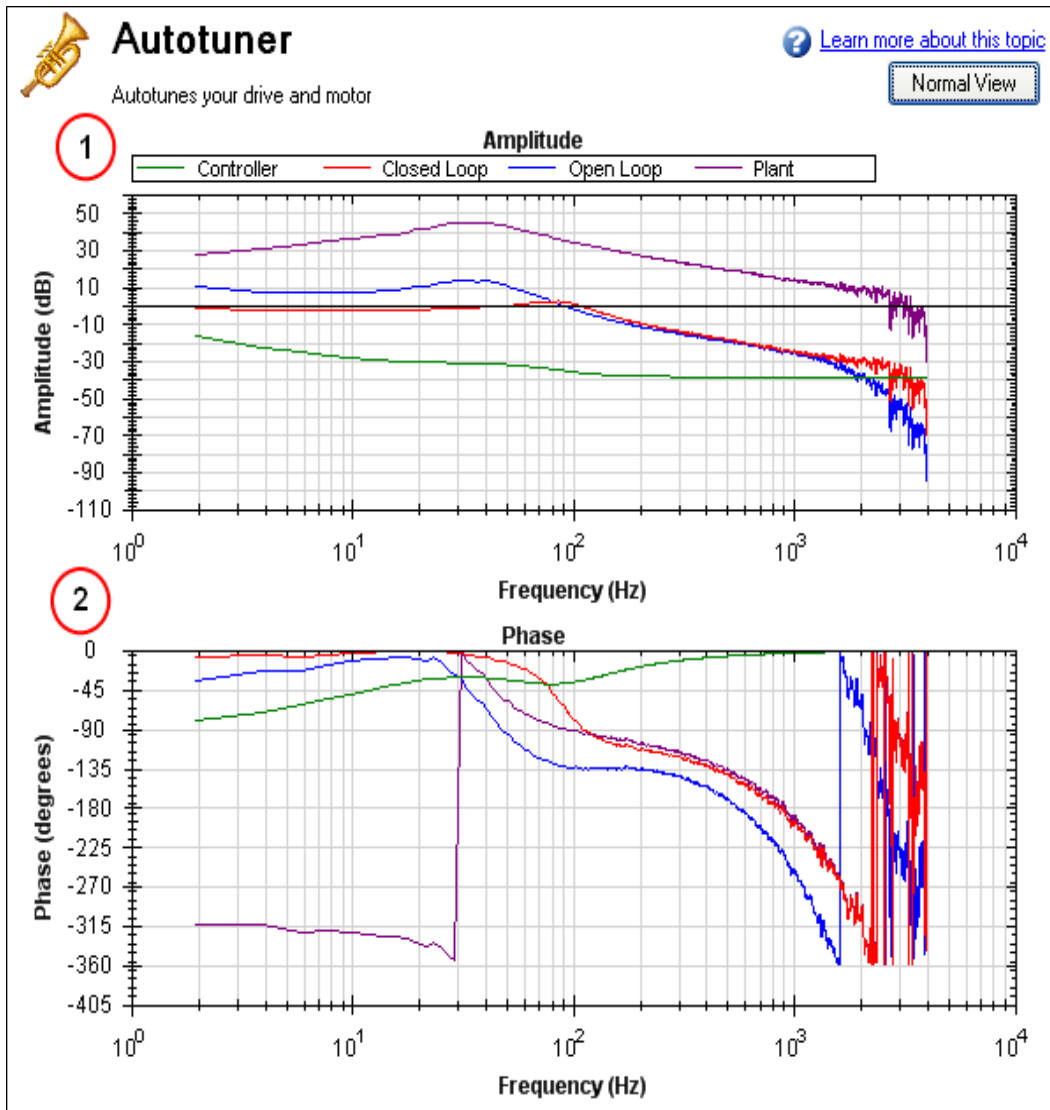


The resulting plots are the frequency response of output/input for each measurement.

For more information regarding these traces, please refer to the [Advanced Tuning For Velocity and Position Loops](#) documentation.

Below is a Bode plot of a motor with no load. The top plot is the magnitude plot (1); this plot shows the gain of the system with respect to frequency. This plot is often used to determine the bandwidth of the servo system.

The lower plot is the phase plot (2). This plot is used in conjunction with the magnitude plot to determine stability, and helps you to understand what kind of latencies exist in the servo system, or if latencies are induced by filters in the velocity loop.



Using the Performance Servo Tuner to Manually Tune Systems

Often, you must manually adjust a control loop in order to obtain optimal machine performance. You can use the Performance Servo Tuner (PST) interface to tune your control loop for best performance. A powerful feature of the manual tuning interface is the ability to simulate the frequency response before it is measured. This feature allows the user to take a base measurement, disable the motor, adjust tuning parameters, and simulate the frequency response of the motor without taking a new measurement. This process saves time and protects equipment from dangerous oscillations.

To begin the manual tuning process, put the Performance Servo Tuner into Bode Plot mode.



Performance Servo Tuner

Autotunes your drive and motor

[Learn more about this topic](#)

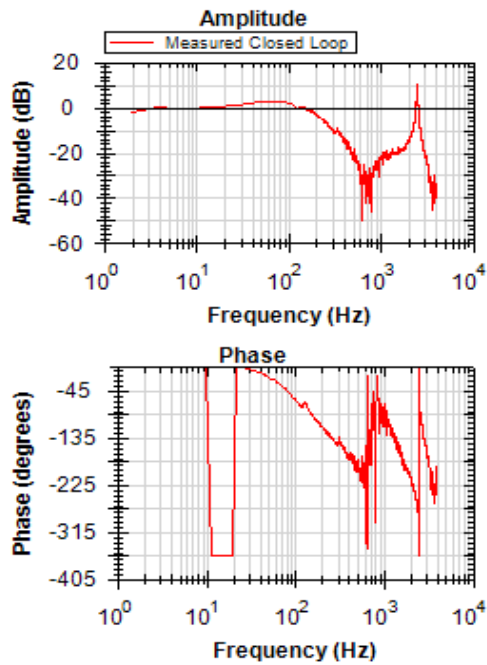
Full View

Tuning Options
 Excitation Level: Start
 Mode: 1

Progress: Complete Complete:

Measurement | Tuning | Plots | Cursors

Current Amplitude: Measurement:
 Velocity Amplitude: Injection Point:
 Smoothing %: Excitation Type:
 FFT Points: Velocity Max: (RPM)
 Number Points: Excite Gap:



Several differences exist between PST and Bode Plot Interfaces:

- When the PST is put into Bode Plot mode, the **Autotuner** tab is removed from the advanced features, and replaced with a **Tuning** tab.
- The **Plots** tab unlocks simulated traces for closed loop, open loop, controller, and feedback filters.

Using the Tuning Simulation

To simulate tuning, there must be a valid Plant Plot in the PST (whether measured with a Bode Plot measurement or a full Autotune).

To selecting simulated plot traces, click on the **Plots** tab and check the following boxes:

Measurement | Tuning | **Plots** | Cursors

Plot Plant

Plot Closed Loop Plot Simulated Closed Loop

Plot Open Loop Plot Simulated Open Loop

Plot Controller Plot Simulated Controller

Plot Feedback Filters Plot Simulated Feedback Filters

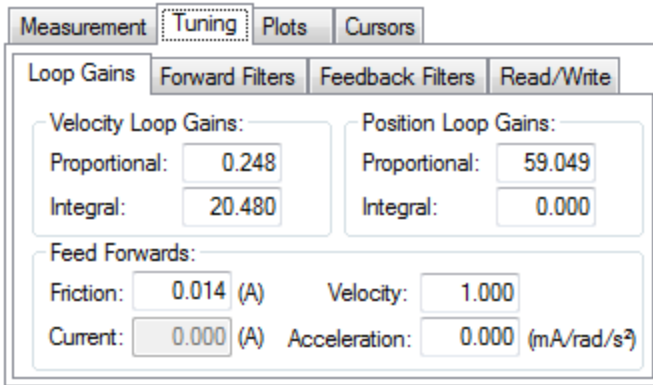
Plot Coherence

These selected boxes are the most common configuration for tuning; however, simulation will occur regardless of the checkboxes selected.

The boxes on the left plot the existing frequency response of the drive based on the tuning parameters that are loaded. The boxes marked "Simulated" (on the right) use the plant data from the measurement and the tuning parameters in the PST to simulate the performance of those tuning parameters without loading them to the drive.

Using the Performance Servo Tuner Manual Tuning Interface

To use the PST manual tuning interface, click on the **Tuning** tab.



Measurement **Tuning** Plots Cursors

Loop Gains Forward Filters Feedback Filters Read/Write

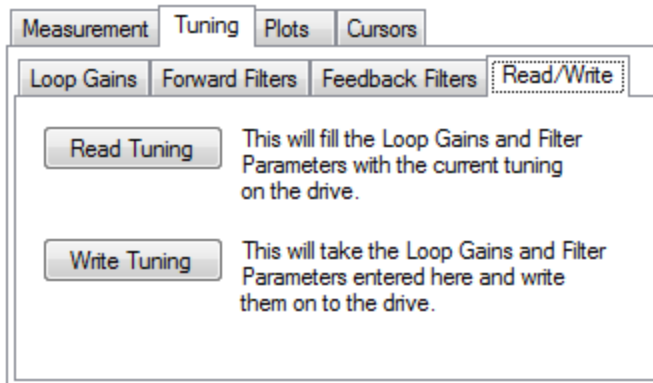
Velocity Loop Gains: Proportional: 0.248 Integral: 20.480

Position Loop Gains: Proportional: 59.049 Integral: 0.000

Feed Forwards: Friction: 0.014 (A) Velocity: 1.000 Current: 0.000 (A) Acceleration: 0.000 (mA/rad/s²)

This tuning interface loads the tuning parameters on the drive each time a measurement is taken. Tuning parameters are split up into Loop Gains (Velocity Loop, Position Loop), Forward Path Biquad Filters, and Feedback Path Biquad Filters.

After modifying tuning gains, you must click on the **Read/Write** tab and click the **Write Tuning** button.



Measurement Tuning Plots Cursors

Loop Gains Forward Filters Feedback Filters **Read/Write**

Read Tuning This will fill the Loop Gains and Filter Parameters with the current tuning on the drive.

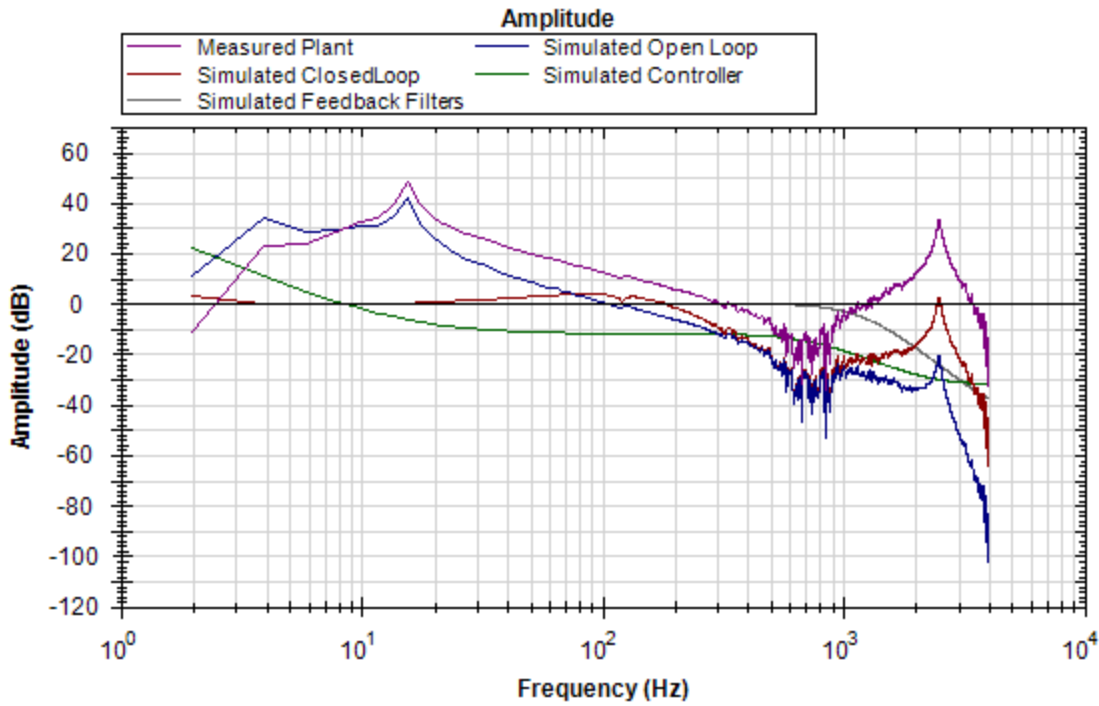
Write Tuning This will take the Loop Gains and Filter Parameters entered here and write them on to the drive.

To restore the tuning on the drive to the PST interface, click the **Read Tuning** button.

Note: If tuning gains are modified and a Bode Measurement is made without clicking the **Write Tuning** button, the PST will overwrite the tuning gains in the interface with the tuning parameters on the drive.

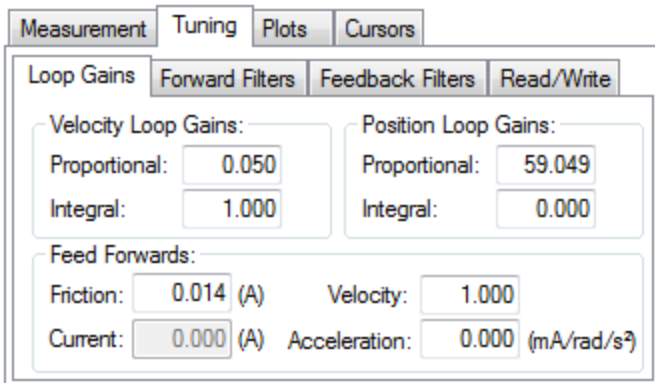
Simulating Modified Loop Gains with the Performance Servo Tuner

Here is the frequency response of a test system after using the PST.

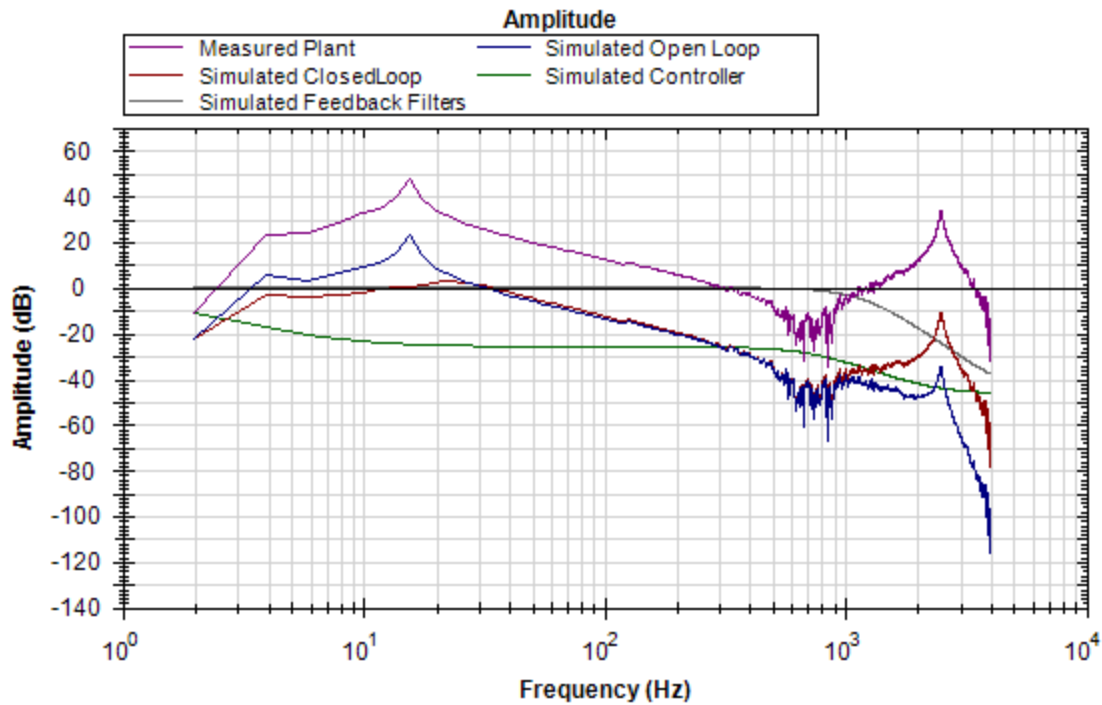


The Velocity Loop Proportional gain here is 0.248. If an application did not need to be tuned as stiff as this, then you could use the PST simulator to detune the motor to the desired bandwidth. A followup Bode Measurement can verify that the simulated response is correct.

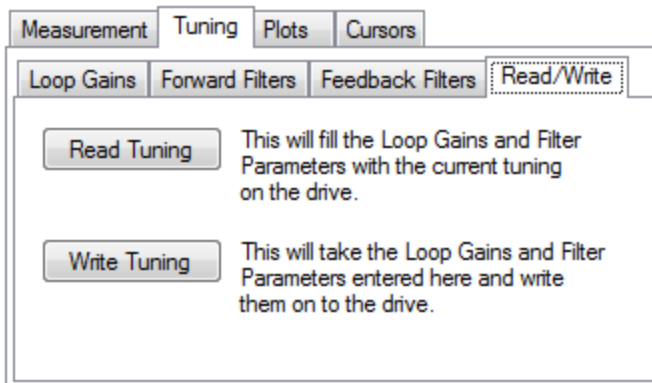
Use the boxes in the **Loop Gains** tab to change tuning gains until the desired frequency response is achieved.



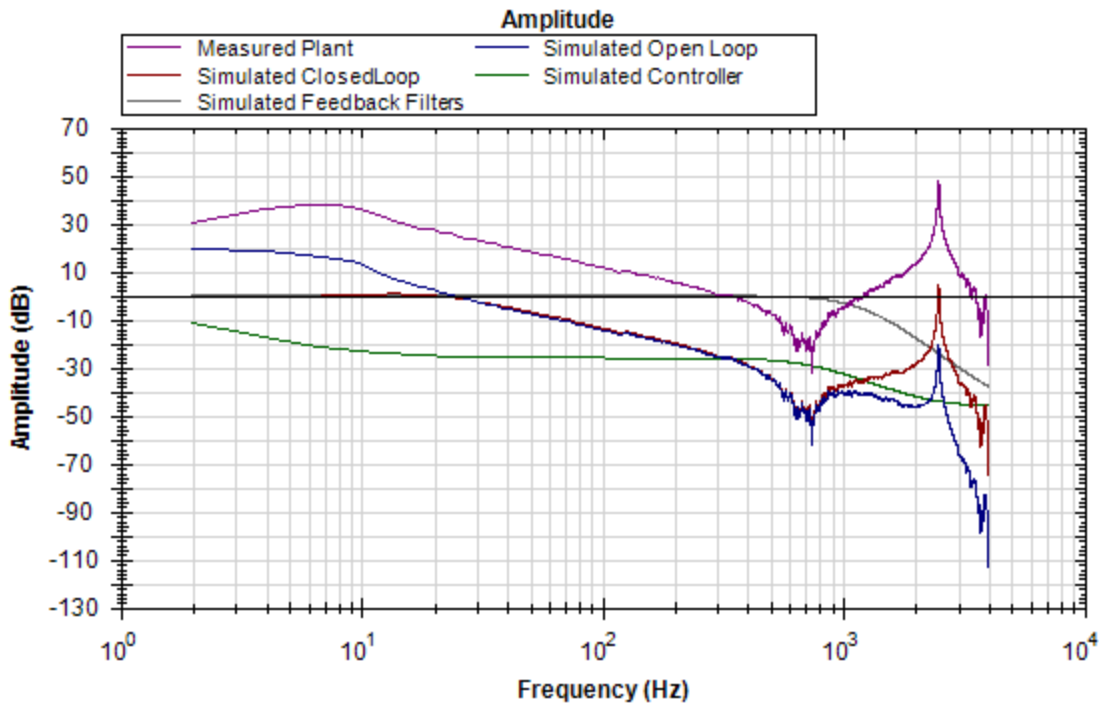
The detuning of velocity loop proportional and integral gains simulated that the bandwidth of the servo has been detuned from ~100 Hz to ~30 Hz.



Next, write the tuning parameters to the drive using the **Write Tuning** button on the **Read/Write** tab.



Now, complete a Bode Plot measurement to compare the simulated result with the new measured result.



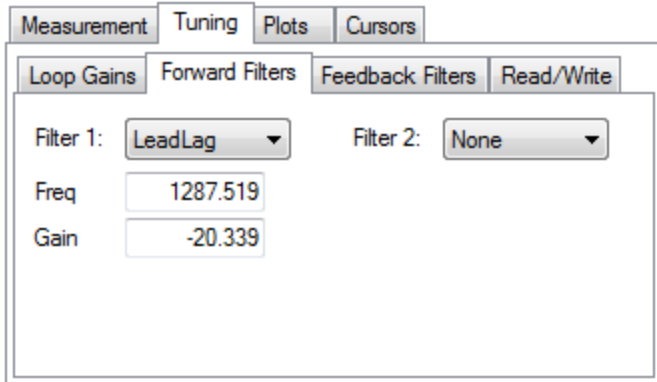
The new measured Bode Plot indicates we achieved slightly lower than 30 Hz bandwidth. The servo is stable, and tuning can be refined until desired performance is reached.

Simulating Filters with the Performance Servo Tuner

Resonances add many challenges to tuning a servo. Using the correct filter in an application can greatly improve system performance when resonances are present.

The Bode plot in this example shows a sharp, high-magnitude resonance at 2500 Hz. Because this is the only resonance, this is an indicator that a resonator (a tunable notch) filter may increase performance.

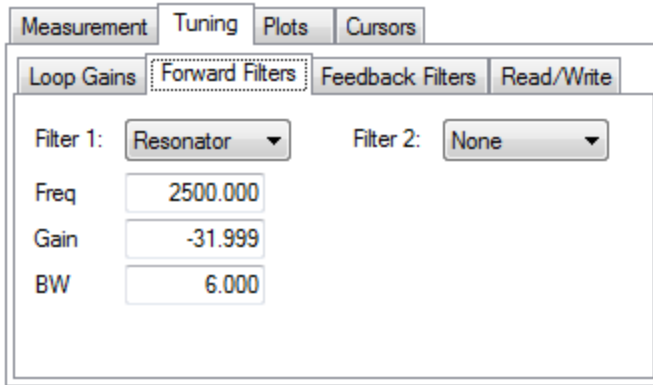
Click on the **Forward Filters** tab:



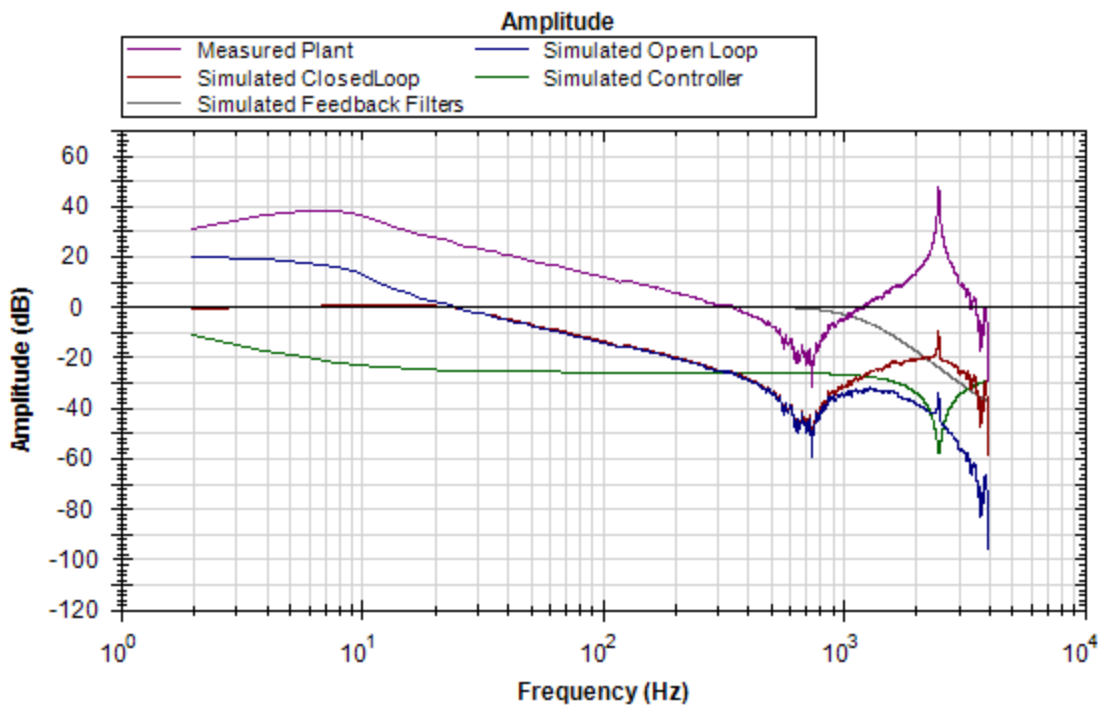
The results of the autotune are still on the drive, and provide adequate tuning. A lead lag filter is the default tuning filter, and is a good general case filter for most servo loops.

Because this test fixture has such a prominent single resonance, we can improve performance (and reduce noise) by placing a notch filter at this resonance.

By tuning a Resonator to best cancel the resonance in the plant, the resonance in the open loop, and therefore the closed loop can be minimized.



The resulting frequency response using the above resonator configuration is shown below:



Notice the attenuation of the resonance in the blue and red traces (open loop and closed loop, respectively).

Using Filters to Reduce Noise

To reduce noise, it is best to place filters in the feedback path. This placement attenuates the noise resulting from a noisy encoder being amplified by the current loop. This noise can be filtered by a forward path filter, however if a filter is placed in the forward path that introduces phase lag (like a lowpass) then your motion profile will exhibit that phase lag in the command signal. If the filter is placed in the feedback path, this lag will be avoided.

14.4 Tuning Guide

14.4.1 Overview

This section covers tuning the velocity and position loops in the AKD. Servo tuning is the process of setting the various drive coefficients that are needed for the drive to optimally control the servo motor for your application. There are different ways to tune, and several are covered here. We will give you guidance on what the different methods of tuning are and when to use them.

The AKD works in three major operation modes: torque, velocity, and position operation mode. No servo loop tuning is required for torque mode. Velocity loop and position loop tuning are covered below.

The AKD has an auto tuner that will provide the tuning that many applications will need. This section describes the tuning process and how to tune the AKD, specifically for cases where the user does not want to use the auto tuner. Tuning in this section will focus on tuning in the time domain. This means that we will look at the velocity or position response vs. time as the criteria we use to decide how well tuned a control loop is tuned.

14.4.2 Determining Tuning Criteria

Choosing the proper specifications for a machine is a prerequisite for tuning. Unless you have a clear understanding of the type of performance needed to push the machine into production, the tuning process will cause more problems and headaches than it solves. Take time to layout ALL the requirements of the machine—nothing is too trivial to consider.

- Determine what the most important criteria are. The machine was likely designed and developed with a certain performance in mind. Include ALL performance criteria in the specification. Do not concern yourself with whether or not the criteria sound scientific. (i.e. If the motion needs to visibly look smooth, put it in the specification. If it can't have any noise, put it in the specification.) At the end of the development phase, the machine's performance should match the performance previously set in the specification. This will ensure that the machine meets its performance goals and that it is ready for production.
- Test the machine with realistic motion. Do not simply tune the machine to make short linear motion, when it will make long, s-curve motions in the real world. Unless you test the machine with realistic motion, there is no way to determine if it is ready for production.
- Determine some specific, quantitative criteria for identifying unacceptable motion. It's better to be able to tell when a motion is unacceptable than to try and figure out the exact point where acceptable motion becomes unacceptable. Here are some examples of motion criteria:
 - a. +/- x position error counts during the entire motion.
 - b. Settling to within +/- x position error counts, within y milliseconds.
 - c. Velocity tolerance of x% measured over y samples.
- DO NOT pick criteria based on what is the most popular technique of the day. It is important to focus on the things that will get the machine into production with reliable performance, based on fundamental understanding of the system.

After you have constructed a detailed servo performance specification, you are now ready to start tuning your system.

14.4.3 Before You Tune

In the worst case, if something goes wrong during tuning, the servo can run away violently. You need to make sure that the system is capable of safely dealing with a servo run away. The drive has several features that can make a servo run away safer:

- Make sure that the limit switches turn the drive off when tripped. If a complete run away occurs, the motor can move to a limit switch very quickly.
- Make sure the max motor speed is set accurately. If a complete run away occurs, the motor can reach max speed quickly and the drive will then disable.

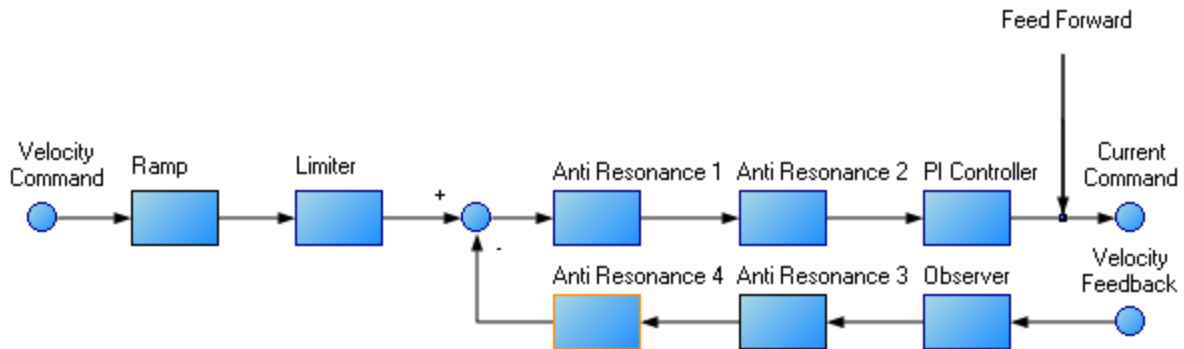
14.4.4 Closed Loop Tuning Methods

The closed loop control loop is responsible for the desired position and / or velocity (trajectory) of the motor and commanding the appropriate current to the motor to achieve that trajectory. The challenge in closed loop control loops is to make a system that not only follows the desired trajectory, but also is stable in all conditions and resist external forces, and do all of this at the same time.

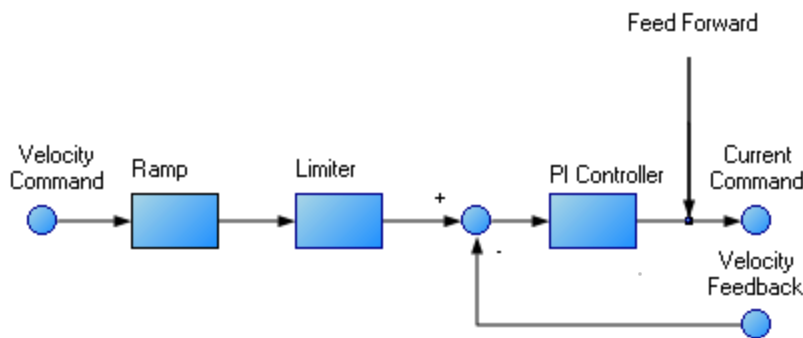
When in velocity operation mode, only the velocity loop is tuned. When in position operation mode, both the velocity and position loops must be tuned.

14.4.4.1 Tuning the Velocity Loop

The velocity loop on the AKD consists of a PI (proportional, integral) in series with two anti resonance filters (ARF) in the forward path and two anti resonance filters in series in the feedback path.



To perform basic tuning of the velocity loop, you can use just the PI block and set ARF1 and ARF2 to unity (no effect) and set the observer to 0 (no effect). Using just the PI block simplifies the process of tuning the velocity loop. To start tuning you can adjust the PI Controller block first. A simplified velocity loop without anti resonant filters and observer is shown below. This is how you can think of the loop before the anti resonant filters and observer is used.



Procedure for simple velocity loop tuning:

1. Set DRV.OPMODE to velocity or position, as appropriate for your application. If DRV.OPMODE is set to position, set VL.KVFF to 1.0.
2. Set VL.KP to 0.
3. Set VL.KI to 0.
4. Set service motion to make a motion that is similar to the move speeds that will be used in the real application. Do not set the service motion to a speed higher than $\frac{1}{2}$ of the maximum motor speed, to allow for safe overshoot during tuning. Set acceleration to an appropriate value for your application. Set service motion to reversing. Set time1 and time2 equal to 3 times the expected settling time for the system. 1.0 second is a reasonable value for time1 and time2, if you don't know the expected settling time.
5. Enable the drive and start the service motion. You should see no motion, as there are no velocity loop tuning gains at this point.

6. When adjusting VL.KP and VL.KI, below record VL.FB and VL.CMD. These are the traces that are used to determine the performance of the velocity loop.

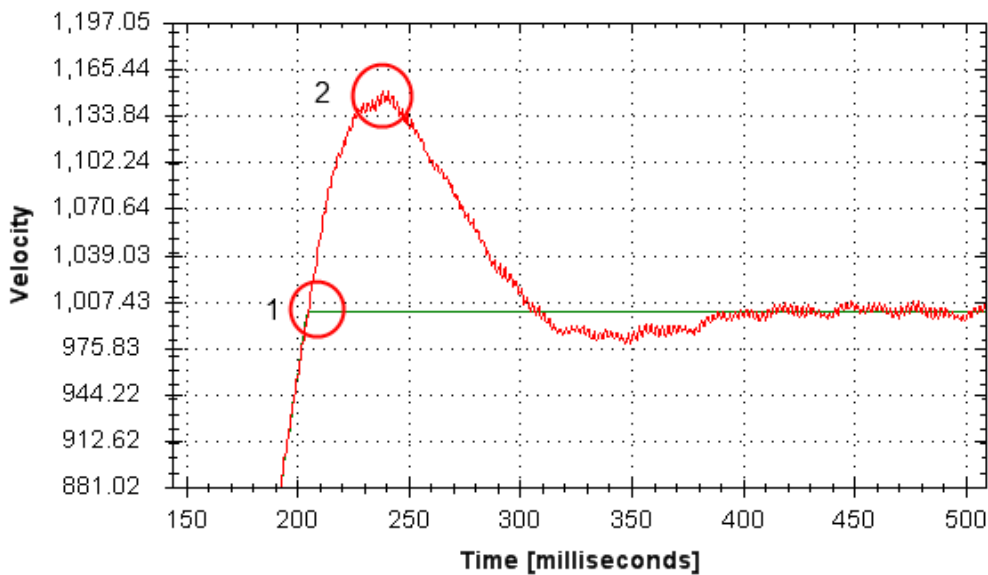
| Channels | | | | | |
|-----------------------|---------------------------|--------|--------------------------|----------|-----------------------------------|
| Time-base and Trigger | | | | | |
| Motion | | | | | |
| Servo Gains | | | | | |
| All Gains | | | | | |
| AR1 | | | | | |
| AR2 | | | | | |
| AR3 | | | | | |
| Id | Source | Color | Hide | Y Axis | |
| 1 | Velocity feedback (VL.FB) | Red | <input type="checkbox"/> | Velocity | Restore Settings Save Settings |
| 2 | Velocity command (VL.CMD) | Green | <input type="checkbox"/> | Velocity | |
| 3 | None | Blue | <input type="checkbox"/> | Default | |
| 4 | None | Purple | <input type="checkbox"/> | Default | |
| 5 | None | Pink | <input type="checkbox"/> | Default | |
| 6 | None | Orange | <input type="checkbox"/> | Default | |

7. Adjust VL.KP. Keep increasing VL.KP by a factor of 2 until you either:

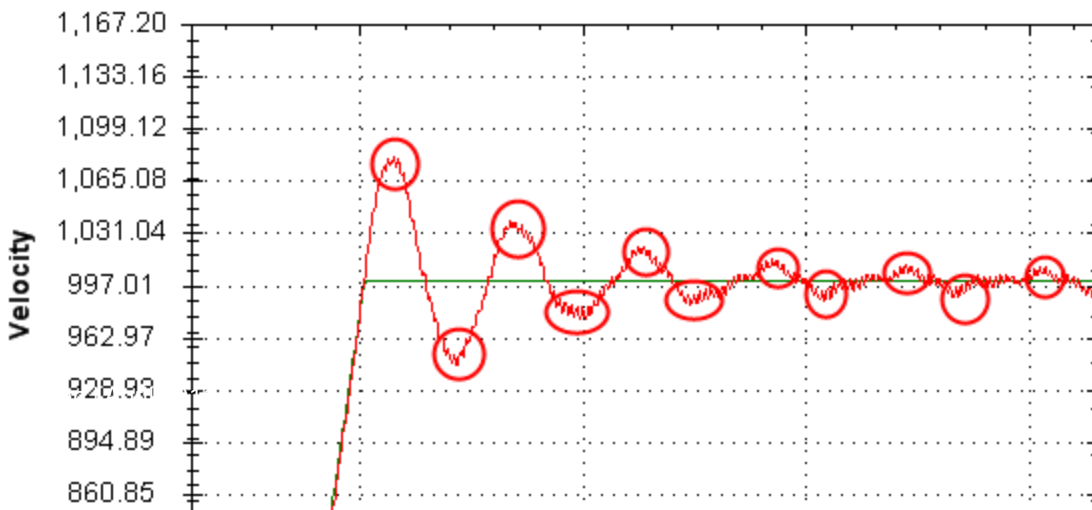
- Hear an objectionable noise from the system (buzzing, humming, etc.)
- See velocity overshoot. No velocity overshoot should be present when using only VL.KP.
- When you reach one of the limits above, decrease VL.KP to the value where there were no objectionable noises or overshoot.

8. Adjust VL.KI. Increase VL.KI by a factor of 1.5 until you either:

- Hear or see objectionable noise or shuddering from the system
- See > 15% overshoot
- Here is an example of 15% overshoot. This is zoomed in view of a service motion commanded to 1000 RPM (location 1), where the overshoot peaks at 1150 RPM (location 2).



- Here is an example of 11 overshoots. Each overshoot is shown by a red circle.



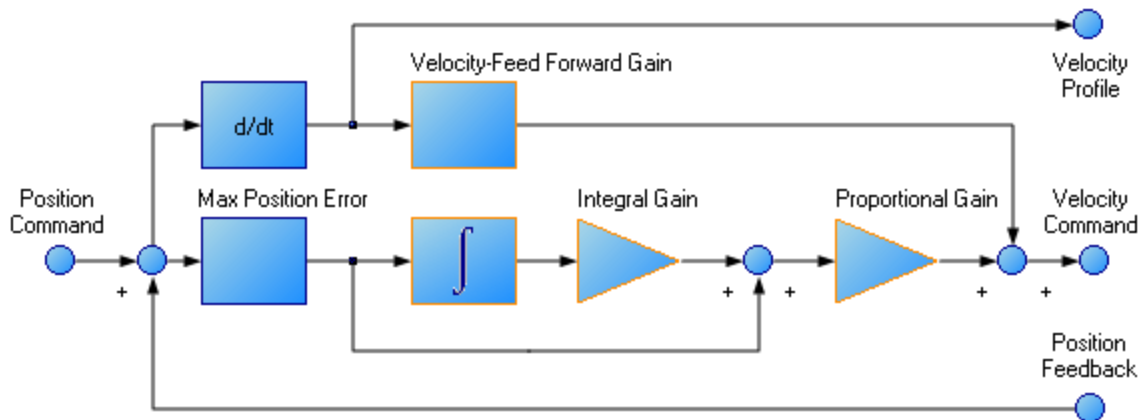
- When you reach one of the limits above, decrease VL.KI to the value where there were no objectionable noises or overshoot.

9. Stop the service motion

14.4.4.2 Tuning the Position Loop

The position loop is a second loop that builds upon a correctly tuned velocity loop to provide accurate control over position. The position loop is a simple element that consists of a PI loop. It is simplest to tune the P and I terms in the velocity loop and use only the P term in the position loop.

At most, use only three non-zero P and I terms from both the velocity loop and the position loop. One combination would be VL.KP, VL.KI, and PL.KP. Another valid combination would be VL.KP, PL.KP, and PL.KI. The VL.KP, VL.KI, and PL.KP combination is shown here.



Procedure for tuning position loop:

1. Set VL.KVFF to 1
2. Increase PL.KP until either:
 - You see 25% overshoot
 - You see > 3 overshoots
 - You hear objectionable noises from the system.
 - When you reach one of the limits above, decrease PL.KP to the value where there were no objectionable noises or overshoot.

14.4.5 Torque Feedforward Tuning Methods

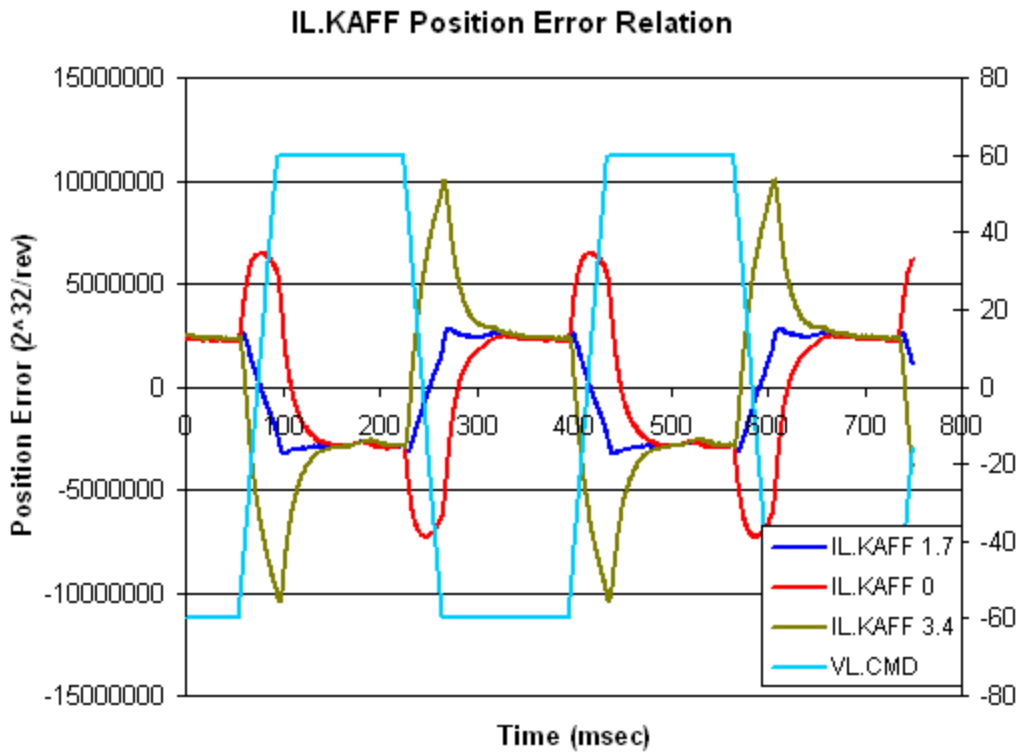
The torque based feedforward terms on the AKD effectively model the physics of your motor and allow the drive to command the appropriate current, even before the encoder has time to send data back to the drive. Torque based feedforward terms allow you to lower following error with virtually no stability penalty.

14.4.5.1 Shape Based Feedforward Tuning

To adjust IL.KAFF:

- Tune the VL.KP and VL.KI as shown above in the velocity loop tuning section. Set DRV.OPMODE to velocity (or set PL.KP and PL.KI to 0 and vl.kvff to 1).
- Set up a short, repeating service motion with accelerations that are representative of the moves you will use in your application (exact values for acceleration are not critical).
- Turn up IL.KAFF until the position error (PL.ERR) is proportional to the inverted velocity command. The adjustment of IL.KAFF will focus on removing bumps on acceleration and deceleration. The pic-

ture below has an ideal value of IL.KAFF of 1.7.



14.4.6 Using Anti-Resonance Filters

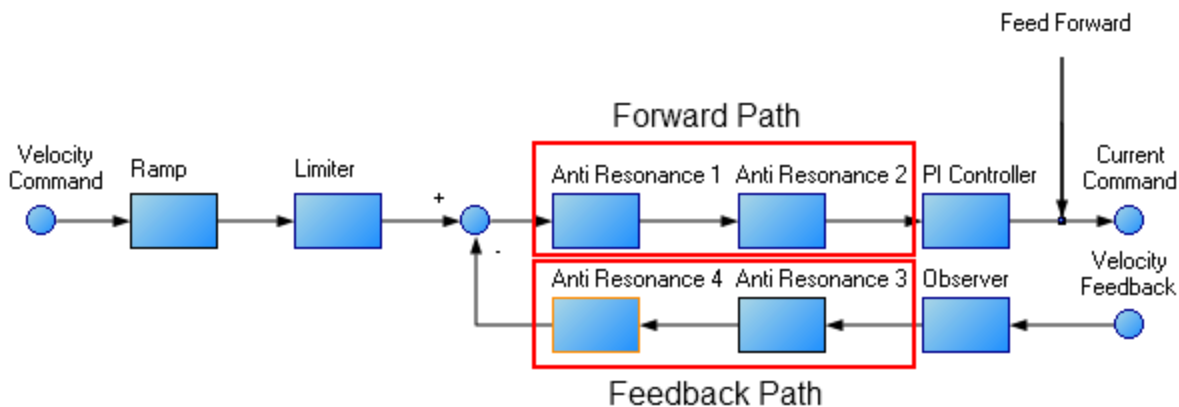
The AKD has four anti resonance filters. Two filters are in the forward path and two are in the feedback path.

Similarities

- Both types are typically used to enhance stability and performance of the system.

Differences

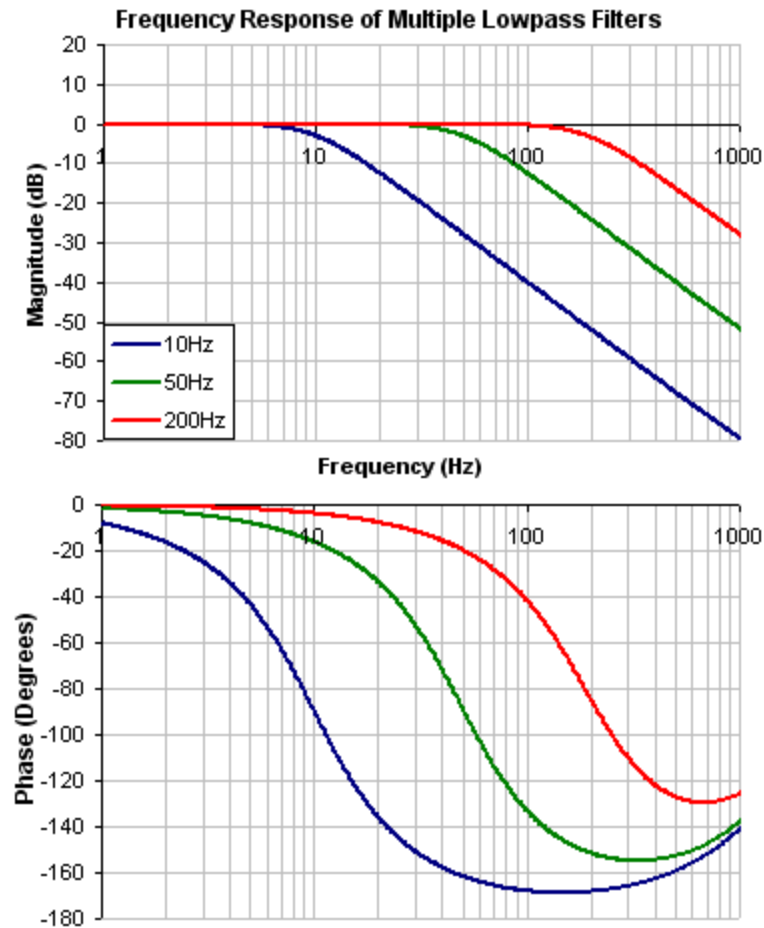
- Forward path filters result in higher phase lag in closed loop system response.
- Forward path filters limit spectrum from reaching the motor / feedback path filters only filter the feedback after it has been to the motor.



1.6.1 Types of Anti Resonance Filters

Low Pass

A low pass filter allows signals through below a corner frequency and attenuates the signals above the same corner frequency. The behavior at the corner frequency can be specified with the low pass Q.



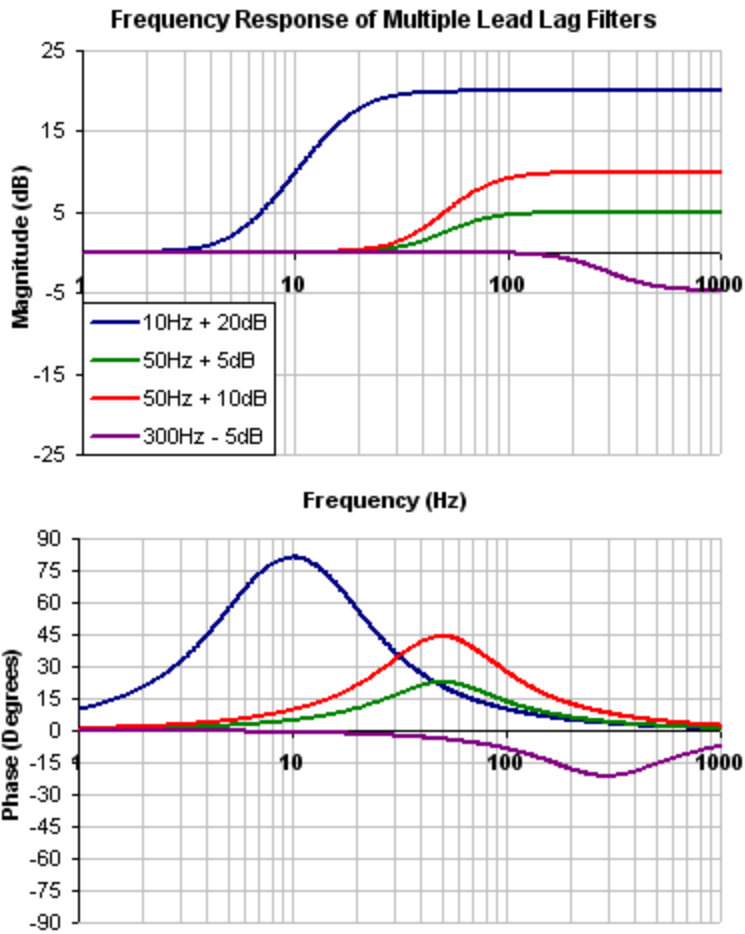
To specify a lowpass filter, you must specify the frequency and Q for both the zero and pole on anti resonance filter 1. To do this, see the following example using the terminal commands that sets:

- Filter Type = Biquad
- Zero frequency = 700 Hz (This is the Lowpass cutoff frequency)
- Zero Q = 0.707
- Pole frequency = 5000 Hz
- Pole Q = 0.707

```
VL.ARTYPE1 0
VL.ARZF1 700
VL.ARZQ1 0.707
VL.ARPF1 5000
VL.ARPQ1 0.707
```

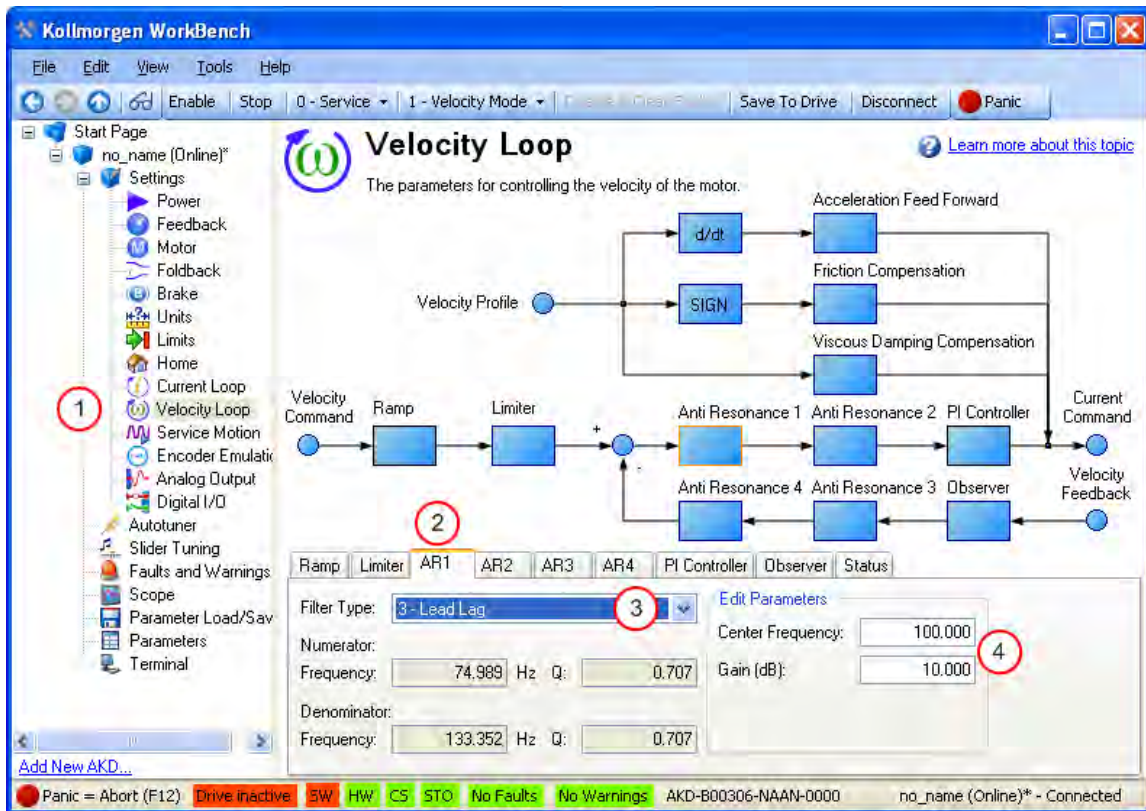
Lead Lag

A lead lag filter is a filter that has 0 dB gain at low frequencies and a gain that you specify at high frequencies. You also specify the frequency that the gain at which the transition occurs.



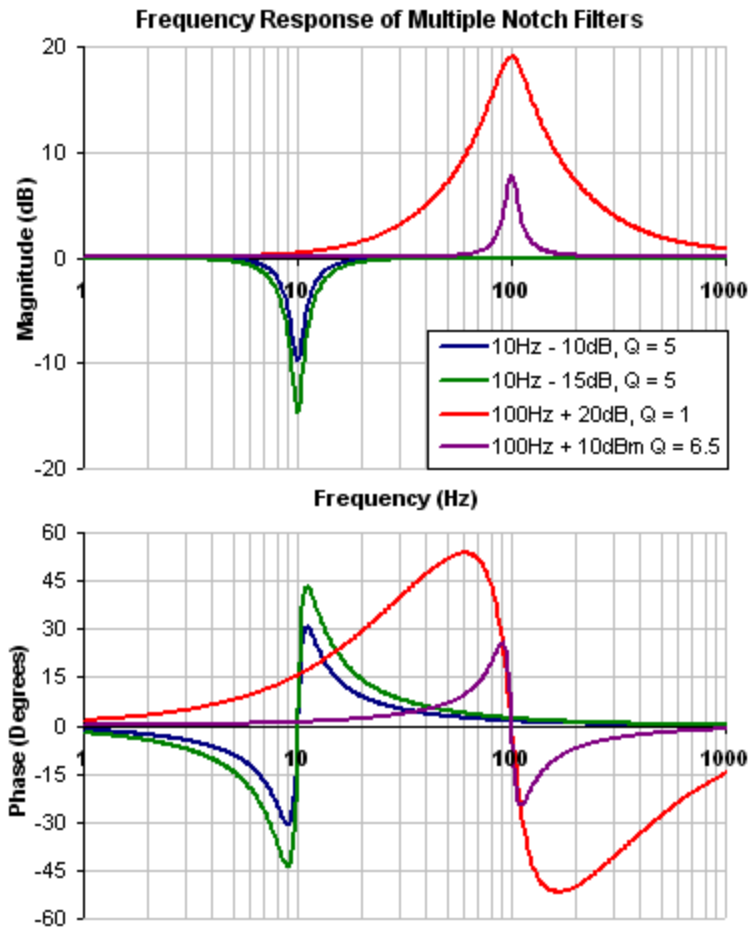
To specify a Lead Lag filter, you must specify the Center Frequency and high frequency Gain (dB). To do this, see the following example by clicking on the Velocity Loop:

Click on Velocity Loop (1), then select the AR1 Tab (2), using the Filter Type drop down, select Lead Lag (3), lastly, enter the desired Center Frequency and Gain of the Lead Lag filter (4).



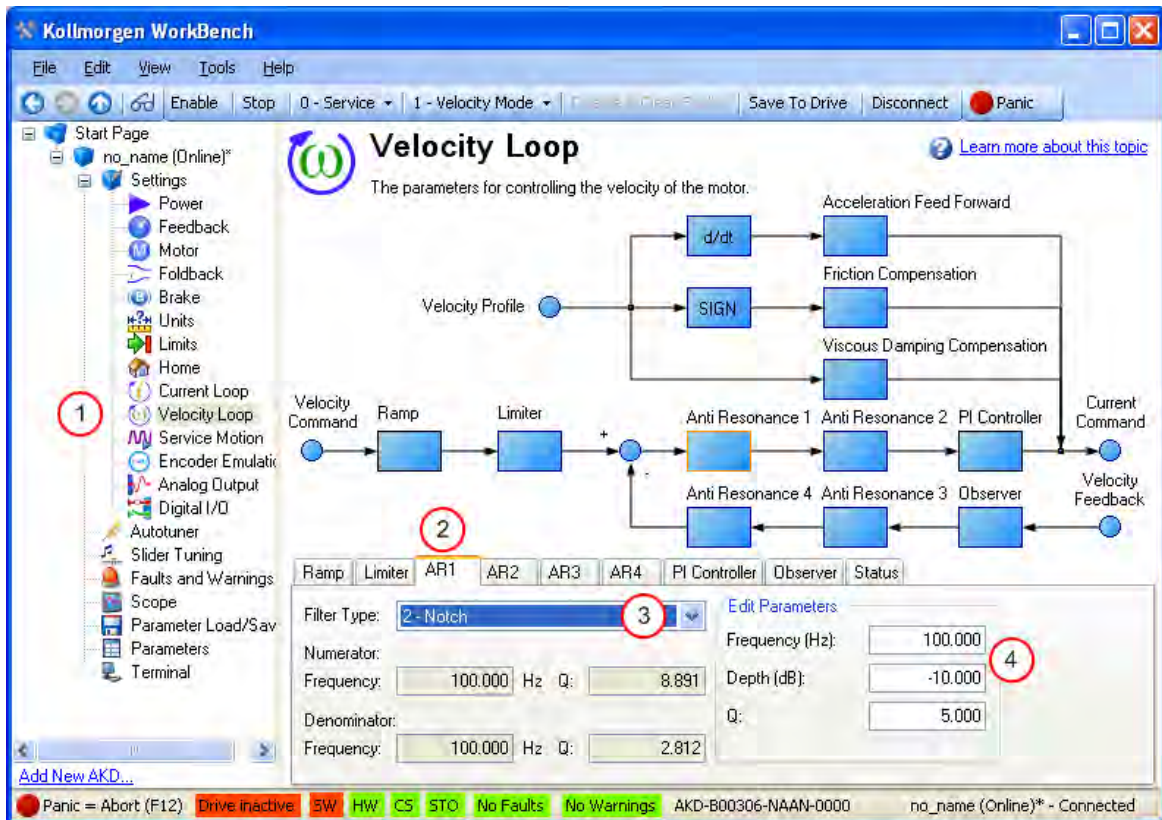
Notch

A notch filter changes gain at a specific frequency. You specify the frequency at which the gain change occurs (Frequency (Hz)), how wide of a frequency range the cut occurs (Q), and how much the gain changes (Notch Depth (dB)).



To specify a notch filter, you must specify the Frequency (Hz), Depth (dB) and Width (Q) of the notch. To do this, see the following example by clicking on the Velocity Loop:

Click on Velocity Loop (1), then select the AR1 Tab (2), using the Filter Type drop down, select Notch (3), lastly, enter the desired Frequency, Depth and Q of the Notch filter (4).



Biquad

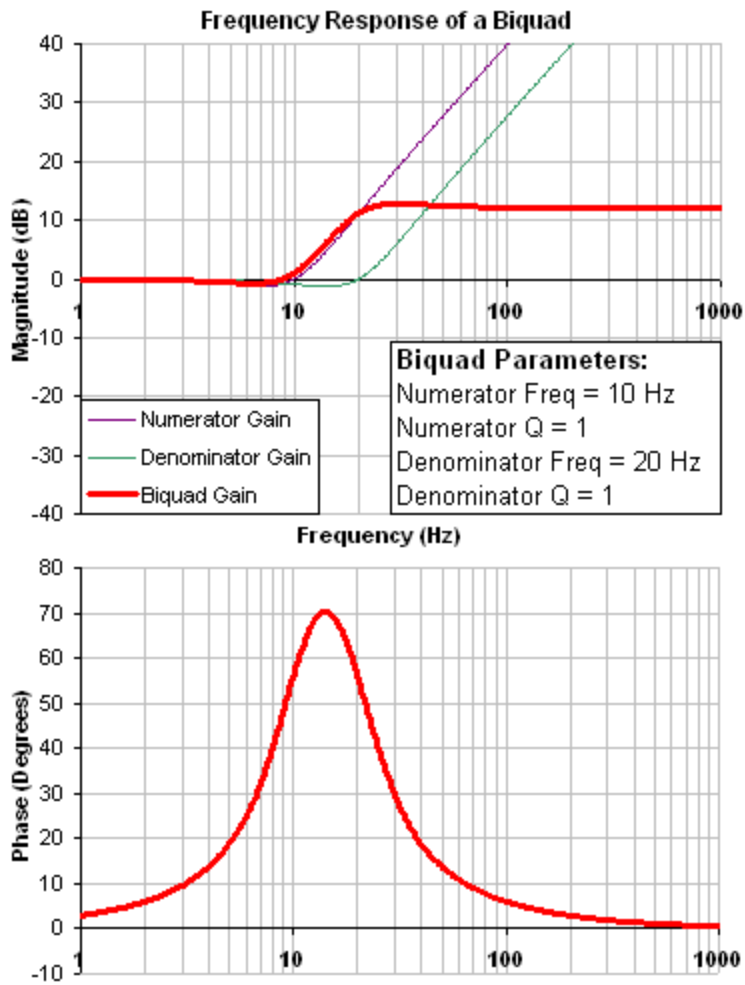
A biquad is a flexible filter that can be thought up as being made up of two simpler filters; a zero (numerator) and a pole (denominator). In fact, the pre-defined filters mentioned above are really just special cases of the biquad.

Both the zero (numerator) and the pole (denominator) have a flat frequency response at low frequencies and a rising frequency response at high frequencies. The transition frequency and damping must be specified for both the numerator and denominator.

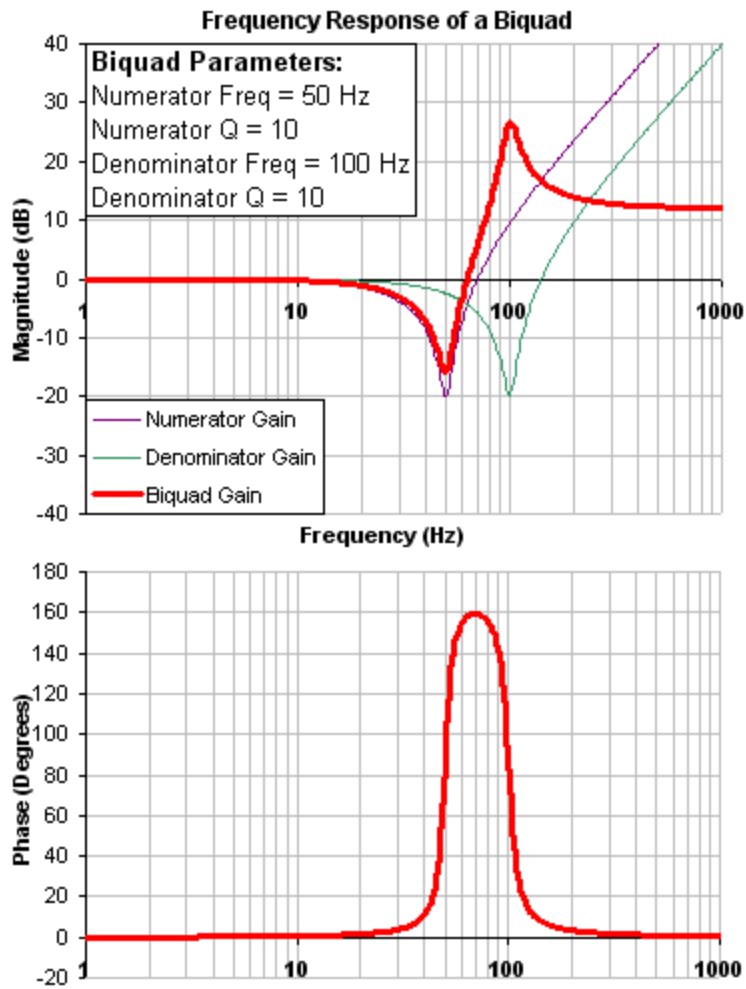
Analyzing the numerator and denominator, the frequency response calculation is simple:

If the numerator and denominator are plotted in dB, the biquad response is numerator – denominator. Understanding how the numerator and denominator work is crucial in understanding how a biquad frequency response is created.

Below is an example of a biquad filter similar to a Lead Lag filter type. To help understand how to determine the frequency response of the biquad, the numerator and denominator response have been plotted. If the denominator is subtracted from the numerator, the biquad response is the result.

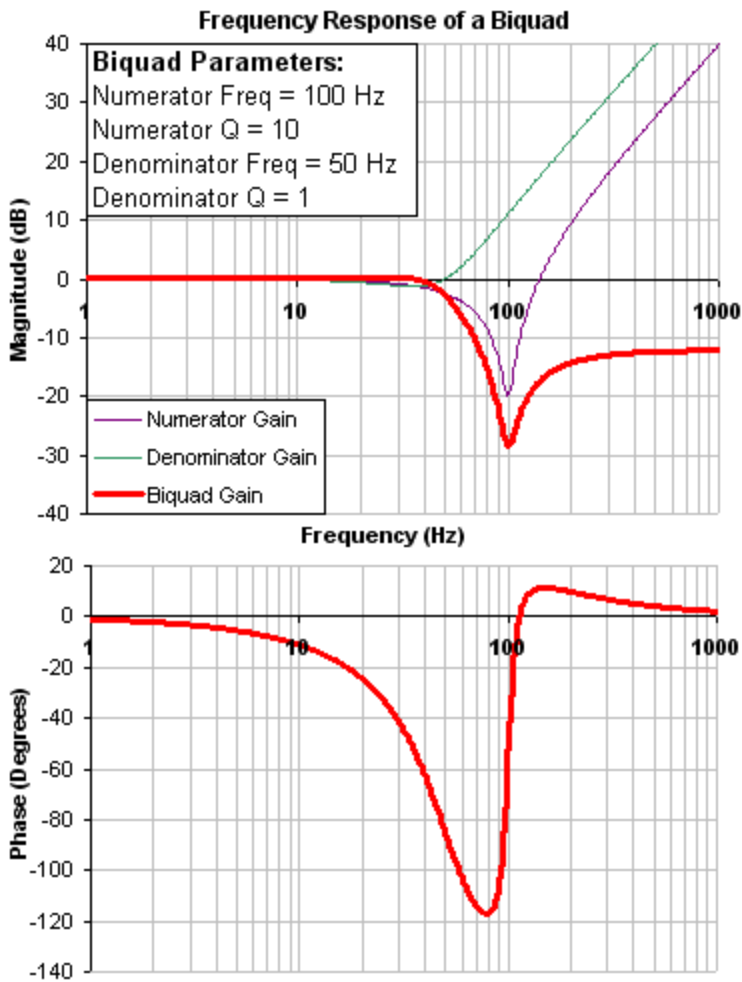


The biquad filter is very flexible, this allows custom filters to be designed. Below is an example of a resonance filter using a biquad, notice how the high Q values affect the numerator and denominator. This gives a biquad frequency response similar to a mechanical resonance.



The previous two examples used a numerator frequency lower than the denominator frequency, yielding a positive gain in high frequencies. If the denominator frequency is lower than the numerator frequency, then high frequencies will have a negative gain.

Below is an example where the numerator frequency is higher than the denominator, notice the high frequencies have a negative gain.



To specify a biquad filter, you must specify the frequency and Q for both the zero and pole on anti resonance filter
 3. To do this, see the following example using the terminal commands that sets:

- Filter Type = Biquad
- Zero frequency = 100 Hz
- Zero Q = 0.7
- Pole frequency = 1000 Hz
- Pole Q = 0.8

```
VLARTYPE3 0
VLARZF3 100
VLARZQ3 0.7
VLARPF3 1000
VLARPQ3 0.8
```

14.4.6.1 Biquad Calculations

In the s-domain, the linear biquad response is calculated:

$$\text{Biquad Frequency Response} = \frac{s^2 + \frac{\omega_N}{Q_N} s + \omega_N^2}{s^2 + \frac{\omega_D}{Q_D} s + \omega_D^2}$$

To convert from idealized s-domain behavior to a more realistic z-domain behavior, we convert using a pole / zero transform. To calculate the frequency response for an individual frequency:

$$t = 62.5 \mu \text{ sec}$$

$$N_{Rad} = 1 - (2Q_N)^{-2}$$

$$N_{2Unscaled} = 1$$

$$\text{if}(N_{Rad} > 0): N_{1Unscaled} = -2e^{-2\omega_N \zeta N t} \cos(\omega_N t \sqrt{1 - \zeta^2})$$

$$\text{if}(N_{Rad} \leq 0): N_{1Unscaled} = -2e^{-2\omega_N \zeta N t} \cosh(\omega_N t \sqrt{1 - \zeta^2})$$

$$N_{0Unscaled} = e^{-2\omega_N \zeta N t}$$

$$D_{Rad} = 1 - (2Q_D)^{-2}$$

$$D_2 = 1$$

$$\text{if}(D_{Rad} > 0): D_1 = -2e^{-2\omega_D \zeta D t} \cos(\omega_D t \sqrt{1 - \zeta^2})$$

$$\text{if}(D_{Rad} \leq 0): D_1 = -2e^{-2\omega_D \zeta D t} \cosh(\omega_D t \sqrt{1 - \zeta^2})$$

$$D_0 = e^{-2\omega_D \zeta D t}$$

$$N_{Scale} = \frac{N_{0Unscaled} + N_{1Unscaled} + N_{2Unscaled}}{D_0 + D_1 + D_2}$$

$$N_2 = N_{2Unscaled} / N_{Scale}$$

$$N_1 = N_{1Unscaled} / N_{Scale}$$

$$N_0 = N_{0Unscaled} / N_{Scale}$$

$$\angle_z = \omega t = 2\pi * freq * t$$

$$Num_{Re} = N_2 \cos(2\angle_z) + N_1 \cos(\angle_z) + N_0$$

$$Num_{Im} = N_2 \sin(2\angle_z) + N_1 \sin(\angle_z)$$

$$Den_{Re} = D_2 \cos(2\angle_z) + D_1 \cos(\angle_z) + D_0$$

$$Den_{Im} = D_2 \sin(2\angle_z) + D_1 \sin(\angle_z)$$

$$Gain_{dB} = 20 \log_{10} \left(\frac{\sqrt{Num_{Re}^2 + Num_{Im}^2}}{\sqrt{Den_{Re}^2 + Den_{Im}^2}} \right)$$

$$Phase_{deg} = \frac{180}{\pi} \left(\tan^{-1} \left(\frac{Num_{Re}}{Num_{Im}} \right) - \tan^{-1} \left(\frac{Den_{Re}}{Den_{Im}} \right) \right)$$

14.4.6.2 Common Uses Of Anti Resonance Filters

Low pass filters in the feedback path. This is a common way to deal with noisy feedback sensors. When used in combination with noisy feedback sensors, significant reduction in audible noise can result.

Lead / lag filters in the forward path. This is a common way to achieve phase lead for control loops without exciting high frequency resonances.

Low pass filters in the forward path. This is a common way to limit high frequency energy from reaching a system that can not productively use energy at these high frequencies. This is also used to lower the effect of system resonances over a wide range of frequencies.

Notch filters are used to cancel system resonances. Notch filters are designed to be the opposite in amplitude of system resonances. Notch filters are applied to very specific frequencies, and therefore you must know your system resonance frequencies accurately to use them effectively.

15 Scope

15.1 Overview

The scope allows you to plot up to six different parameters from the drive. Use **Full View** and **Normal View** to toggle between the scope setup (normal) and a larger view of only the scope output (full). You can configure, save, and restore scope settings from the normal view. The lower right corner of the normal view also includes a box that indicates status and drive and scope control buttons (**Enable Drive**, **Start Recording**, and **Refresh**).

15.2 Using the Scope

You can set up scope plots using the tabs summarized below:

| Tab | Function |
|------------------------------|--|
| Channels | Select data source, plot axes, and plot appearance. |
| Time Base and Trigger | Select how much data to record and when to start recording the data. |
| Service Motion | Generate basic motion. |
| Servo Gains | Adjust the servo loop gains. |
| All Gains | View all current tuning gains in the drive and manually edit gains. |
| AR1, AR2, AR3, AR4 | Adjust filter settings. |
| Save and Print | Save the plot as a raw data file or as an image file; email the plot; print the plot; open the data file in Excel. |
| Measure | Display basic data read from the plots. |
| Cursors | Turn on the cursors and view the data at the cursor positions. |
| Display | Pan, zoom, and control the grid and background color. |

15.2.1 Scope Channels Tab

The **Channels** tab allows you to select and record up to six channels simultaneously. Select the data to record for each channel from the lists in the **Source**, **Color**, **Y-axis**, and **Filter** and **Filter Frequency** columns. Once a recording is shown on the scope screen, you can click **Hide** to remove a channel from the scope display.

| Id | Source | Color | Hide | Y Axis | Filter | Filter Frequency |
|----|-----------------------------|--------|--------------------------|----------|-------------------------------------|------------------|
| 1 | Current feedback (IL.FB) | Red | <input type="checkbox"/> | Current | <input checked="" type="checkbox"/> | 400 |
| 2 | Velocity command (VL.CM...) | Green | <input type="checkbox"/> | Velocity | <input checked="" type="checkbox"/> | 400 |
| 3 | Velocity feedback (VL.FB) | Blue | <input type="checkbox"/> | Velocity | <input checked="" type="checkbox"/> | 400 |
| 4 | None | Purple | <input type="checkbox"/> | Default | <input checked="" type="checkbox"/> | 400 |
| 5 | None | Purple | <input type="checkbox"/> | Default | <input checked="" type="checkbox"/> | 400 |
| 6 | None | Orange | <input type="checkbox"/> | Default | <input checked="" type="checkbox"/> | 400 |

15.2.1.1 Source Column

To set a channel to record, click the source you want to set and choose the appropriate channel. You can choose from None (no data is collected on that channel), preset trace types, or enter a user defined trace. Choosing “<User Defined>” allows you to record data from pre-defined locations. These locations are provided by the factory to collect less common values.

| Id | Source | Color | Hide | Y Axis | Filter | Filter Frequency |
|----|-------------------------------------|--------|--------------------------|----------|-------------------------------------|------------------|
| 1 | Analog output signal (IL.FB) | Red | <input type="checkbox"/> | Current | <input checked="" type="checkbox"/> | 400 |
| 2 | Analog output user value (VL.CM...) | Green | <input type="checkbox"/> | Velocity | <input checked="" type="checkbox"/> | 400 |
| 3 | Analog output user value (VL.FB) | Blue | <input type="checkbox"/> | Velocity | <input checked="" type="checkbox"/> | 400 |
| 4 | Analog output user value (VL.CM...) | Purple | <input type="checkbox"/> | Default | <input checked="" type="checkbox"/> | 400 |
| 5 | Blue voltage (RBUS V(L)) | Purple | <input type="checkbox"/> | Default | <input checked="" type="checkbox"/> | 400 |
| 6 | Current - V winding of motor | Orange | <input type="checkbox"/> | Default | <input checked="" type="checkbox"/> | 400 |
| 6 | Current - V winding of motor | Orange | <input type="checkbox"/> | Default | <input checked="" type="checkbox"/> | 400 |
| 6 | Current command (VL.CM...) | Orange | <input type="checkbox"/> | Default | <input checked="" type="checkbox"/> | 400 |

15.2.1.2 Color Column

For valid sources, you can click on the color in the Color tab and choose a different color than the default, or create a custom color.

15.2.1.3 Hide Column

You can check the **Hide** box to hide a given plot trace. This feature can make it easier to focus on specific data as needed.

15.2.1.4 Y-Axis Column

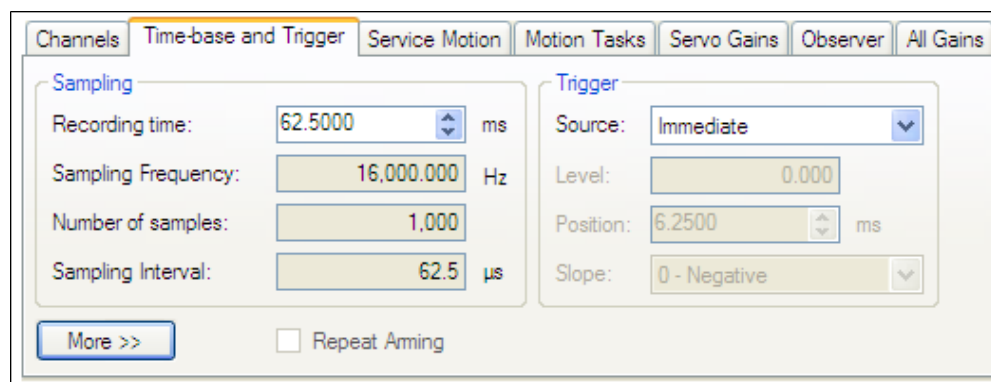
The Y-axis column allows you to choose on which Y-axis the channel will be displayed. Several predefined Y-axis groups exist. Click on the item in the column to change the label for the trace.

15.2.1.5 Filter and Filter Frequency Column

Check this box and use the frequency column to apply a low pass filter to the data collected. The filter is applied when the data is collected. It is not applied to data already collected if this is checked after the data is collected.

15.2.2 Scope Time-base and Trigger Tab

Use the **Time-base and Trigger** tab to select how much data to record and when to start (trigger) recording the data. You can set length of recording in ms and the sampling frequency in Hz. The number of samples is a calculated value displayed for reference. The trigger can be set to trigger immediately when you click **Start Recording** or to trigger when a specified value for a given signal is reached. The default **Time-base and Trigger** view specifies recording time, sampling frequency, and either an immediate trigger or a trigger based on a specified signal. Click the **More** button in this view to specify a given number of samples, sampling frequency, sampling interval, and access additional trigger options.

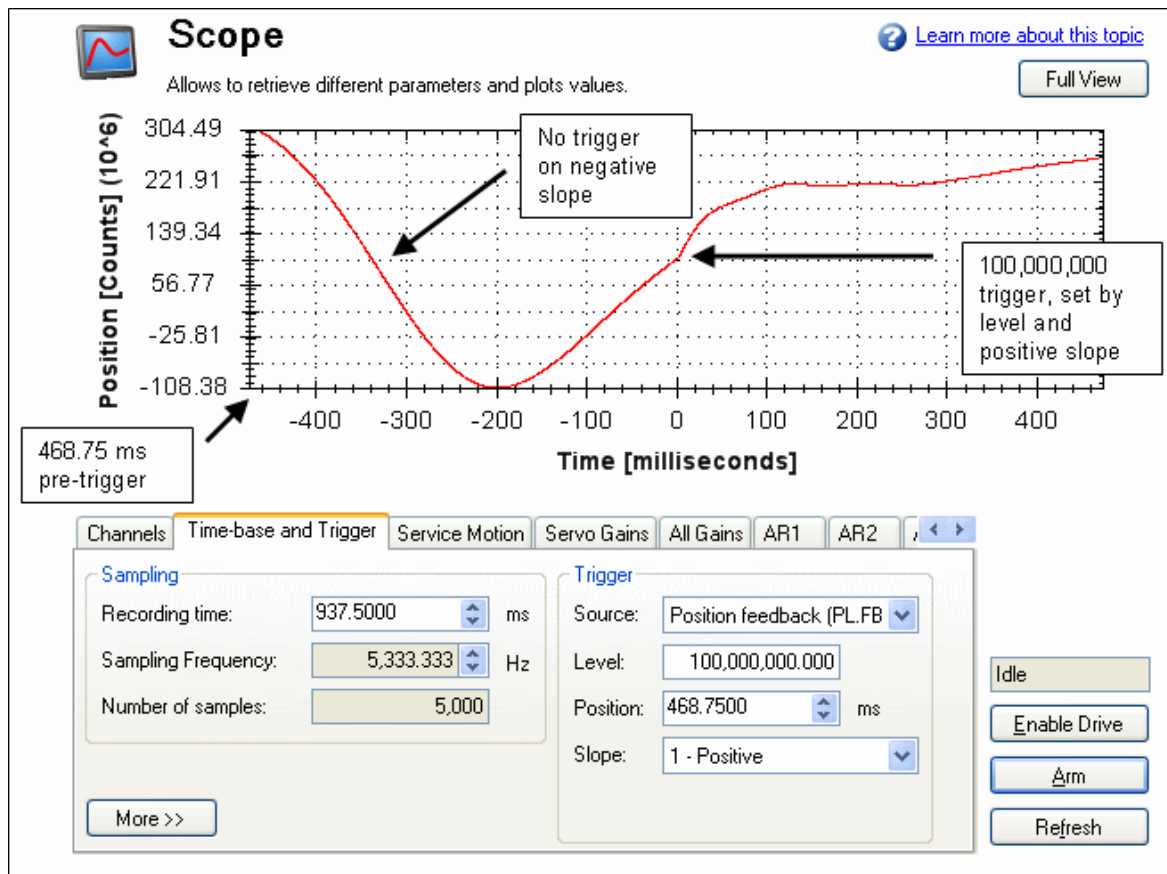


In this tab, you can set length of recording in ms and the sampling frequency in Hz. The number of samples is a calculated value displayed for reference. You can also choose the trigger source to be **Immediate** (triggers as soon as you click the **Start Recording** button) or to be one of many predefined sources.

If you choose a source other than **Immediate**, you can set the level, position, and slope for the trigger value.

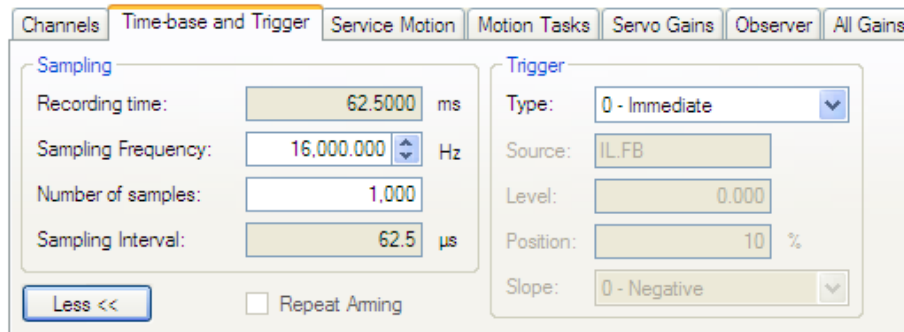
- Level sets the value of the source that triggers the recording to start.
- Position sets the amount of time that the scope displays before the trigger occurred.
- Slope sets whether the source data must pass the level value in a positive or negative direction.

An example of triggering is shown below:



15.2.2.1 Scope Time-base and Trigger, More View

Click **More** to display additional options for configuring the time-base and trigger.



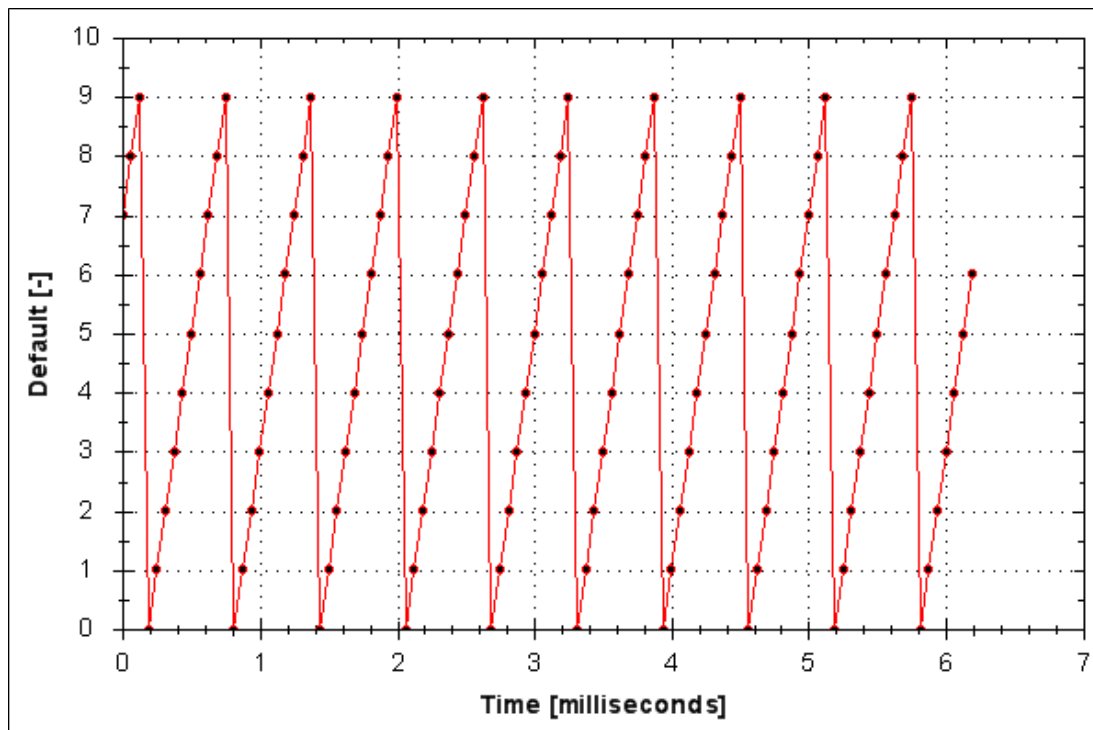
In the **Sampling** area of this view, you can specify the recording length by entering a sampling frequency and a number of samples. Here, the recording time is a calculated value displayed for reference.

What is triggering?

Triggering allows you to precisely control the start point of data collected in the scope. For example, if you are looking for a large spike, you can set the trigger to start the scope to begin recording when it sees the large spike. This section describes the triggering functionality of the scope.

Test Signal

As an example, it is useful to examine variations on a record of a test signal that generates a sawtooth signal. The signal starts at 0 and increases by one every drive sample (1/16,000 second) to a maximum of 9, then returns to 0. This signal continues indefinitely. The record of this signal is shown below.



15.2.2.2 Trigger Type

The **Trigger** area in the **More** view offers more flexibility than the default view. You can specify four types of trigger types (REC.TRIGTYPE):

- **Immediate - 0.** This mode will start recording as soon as the recording command (REC.TRIG) is received by the drive.
- **Command / On Next Command – 1.** This trigger type lets you specify a trigger on the next telnet command received by the drive. This is useful in a telnet session via Hyperterminal (or a similar program). WorkBench is constantly sending telnet commands, so this is not typically used in a WorkBench session.
- **Parameter / On Source Signal – 2.** This trigger type lets you specify a trigger source and set of conditions to trigger recording of data. This is very similar to the triggering used on oscilloscopes.
- **Boolean – 3.** This trigger type lets you trigger on a boolean (0 or 1), such as drive active status.

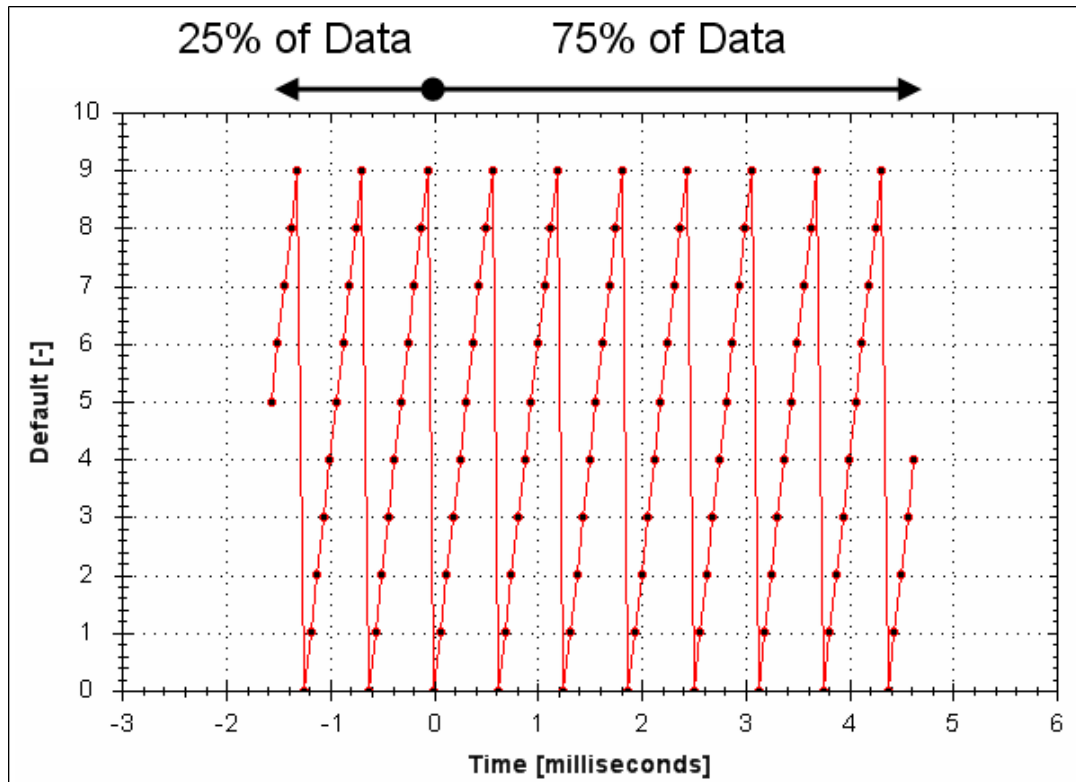
15.2.2.3 Trigger Position

Trigger Position (REC.TRIGPOS) allows you to collect data that occurs before the trigger occurs. If you have a rare condition, you may want to see the conditions that led up to it. Trigger position lets you control how much signal is collected before the trigger condition occurred.

Trigger position is specified in units of percent (%). If you specify a trigger position of X% , X% of the data is before 0 ms in the data time and 100-X% (the rest of the data) is at or greater than 0 ms. In the picture below, trigger position is set to 25% (REC.TRIGPOS 25).

In the WorkBench scope, the 0 time point is clear. When collecting the data via REC.RETRIEVE or similar commands, the time is not returned, so some caution should be used when the trigger point is important to understand.

Trigger position is not used in trigger type “Immediate” (TRIGTYPE 0).



15.2.2.4 Trigger Value

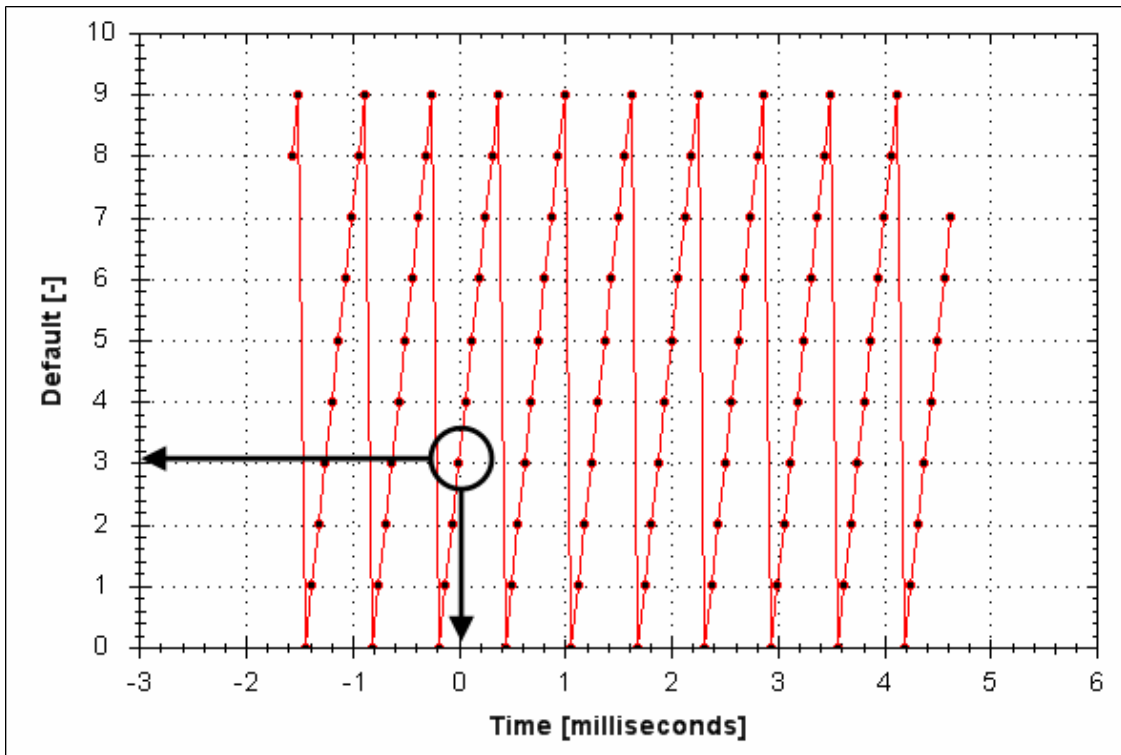
The trigger value (REC.TRIGVAL) specifies a target value that should trigger the recording to start. The trigger value is used in trigger type Parameter / On Next Signal only.

The trigger value is not used in the boolean trigger type. Use the trigger slope to set the polarity of the boolean trigger.

When the trigger slope is positive, the trigger value will trigger when:

- The trigger source is less than the trigger value in the previous recording sample
- The trigger source is greater than or equal to the trigger value in the current recording sample

Below is an example showing triggering of trigger value of 3 (REC.TRIGVAL 3) and positive trigger slope (REC.TRIGSLOPE 1). You can see that the recording triggers at time zero when the source reaches the value of 3.



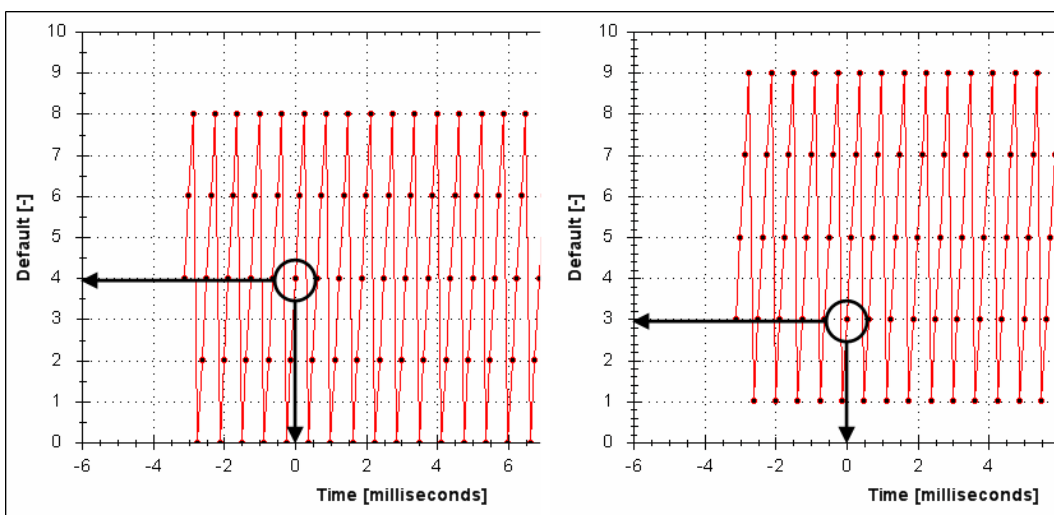
When the trigger slope is negative, the trigger value will trigger when:

- The trigger sources is greater than the trigger value in the previous recording sample.
- The trigger source is less than or equal to the trigger value in the current recording sample.

15.2.2.5 Effects of Recorder Gap

When the recording rate is less than 16,000 Hz ($REC.GAP > 1$), there can be some impact on the triggering of the recorder. When using pretriggering and a recording rate of less than 16,000 Hz, the trigger only evaluates every N samples, where N is the value of REC.GAP. Two effects result from this condition:

1. You cannot be sure of the moment that the recorder is triggered any closer than N samples. An example of this is shown below where the trigger value is set to 3, the trigger slope is positive and the recorder gap is 2.
2. Both examples are the same data, but one instance collected and triggered on the odd data. The other example collected and triggered on the even data.



2. You can miss triggers, whose duration is less than N samples, where N is the value of REC.GAP. This is because the trigger is only evaluated every N samples.

A workaround for the above effects is available by setting the recorder trigger position to zero (REC.TRIGPOS 0). This eliminates conflicts between pretrigger and post-trigger timing and will guarantee trigger evaluation every sample, eliminating the cases above.

15.2.2.6 Trigger Slope

Trigger Slope specifies whether you trigger on a positive or negative change in the trigger source. The effect of the trigger slope is different for trigger type Boolean and On Next Signal modes.

Boolean Trigger Type

When using Boolean type:

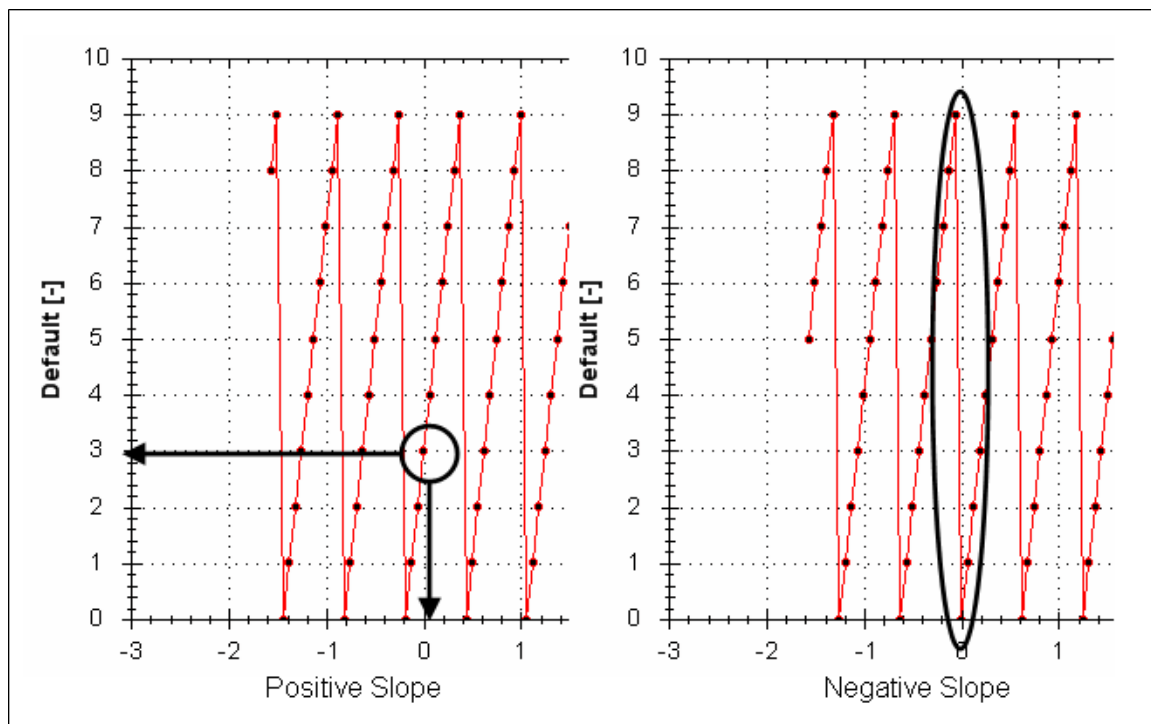
- A positive slope will trigger when the trigger source is 1
- A negative slope will trigger when the trigger source is 0

The boolean trigger type is a state trigger. There is no need to transition from 0 to 1 to trigger with the positive slope. If the trigger source is 1 from the start, the positive slope will immediately trigger.

On Next Signal Trigger Type

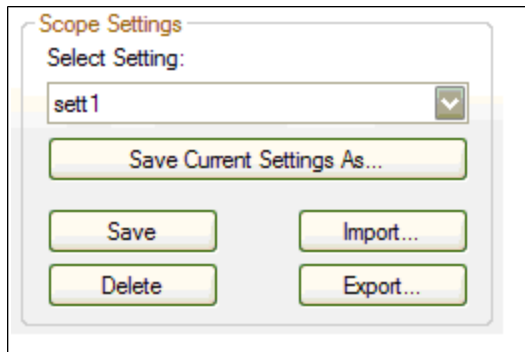
With "On Next Signal" trigger type allows you to specify if the recorder should trigger when the signal crosses the trigger level in the positive or negative direction. The signal only needs to reach the trigger level; it does not need to pass the trigger level.

In the examples below, the trigger value is set to 3 (REC.TRIGVAL 3.000). You can see that with positive slope, the trigger occurs exactly when the signal transitions from 2 to 3, because it reached 3. The negative slope case triggers when the signal transitions from 9 to 0, because it crossed 3 on the way.



15.3 Scope Settings

Scope settings are used to store and retrieve the scope parameters. You can save multiple settings, called "presets", under different names. You can save, delete, import, or export the presets. The settings are stored in WorkBench project file (default.wbproj) and settings are common to all the drives in WorkBench.

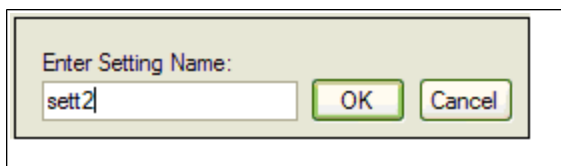


15.3.1 Load a setting (preset) to Scope screen

In Scope Settings section, the existing presets are listed in the **Select Setting** box. To load a setting to the scope screen, select the desired preset from the **Select Setting** list.

15.3.2 Create a new preset

1. Modify any scope parameters.
2. Select **Save and Print** tab.
3. Click **Save current settings As**. The following dialog is displayed:



4. Enter the setting name and click **OK**. The current settings are saved as a preset with the given name and displayed in the list.

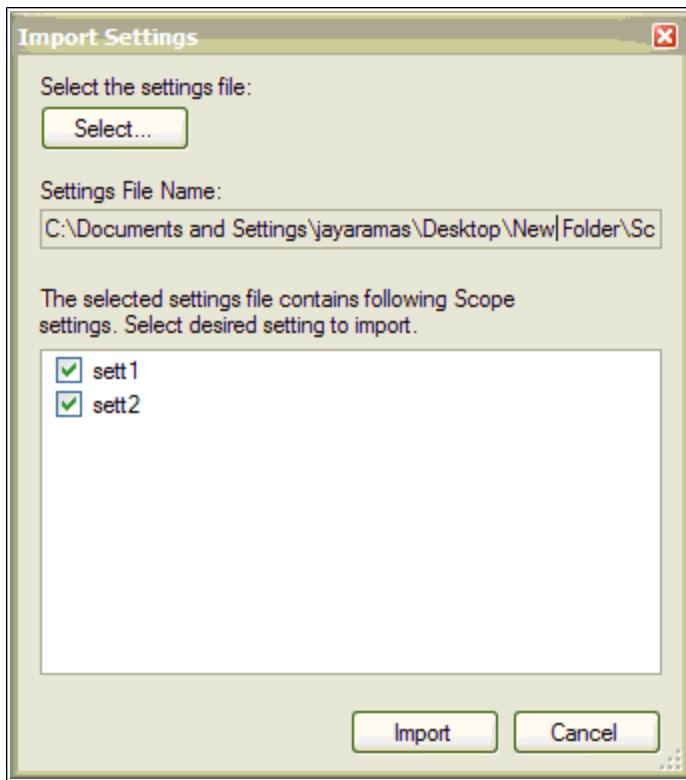
15.3.3 Save or delete preset

Save saves any modification to the open preset. **Delete** deletes the open preset.

15.3.4 Import preset

Import the presets contained in the selected settings file as follows:

1. Click on Import button following dialog will be displayed.

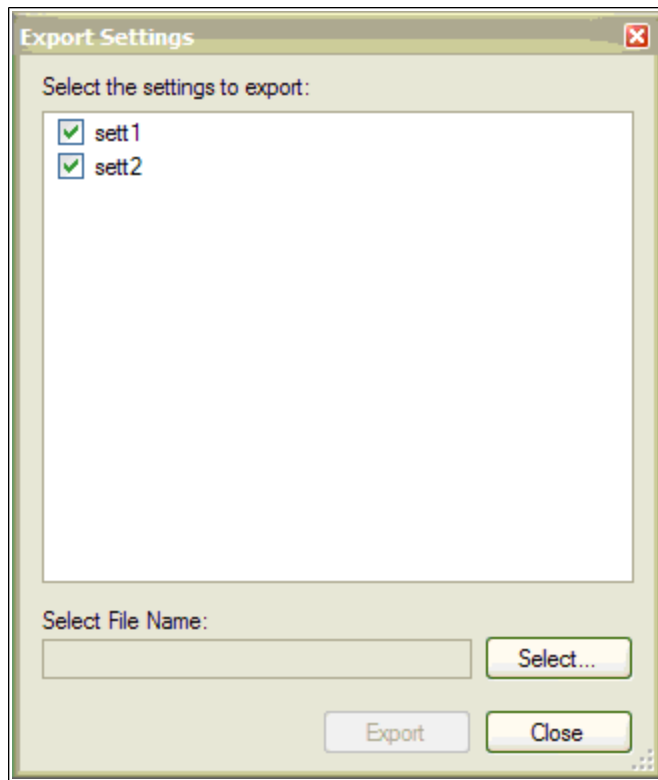


2. Select the settings file by clicking “Select...” button.
3. All the scope presets will be displayed contained in the selected settings file.
4. Select/Deselect the presets and then click on Import.
5. If preset name already exists in application the confirmation message will be shown to user to replace it or to ignore.

15.3.5 Export preset

Export a preset to a file as follows:

1. Click **Export** and the following dialog is displayed:



2. The existing presets are displayed and user can select/deselect the preset to export.
3. Select the file name to export.
4. Click **Export** to export the selected presets to a file.

15.3.6 Scope axis scaling and zooming

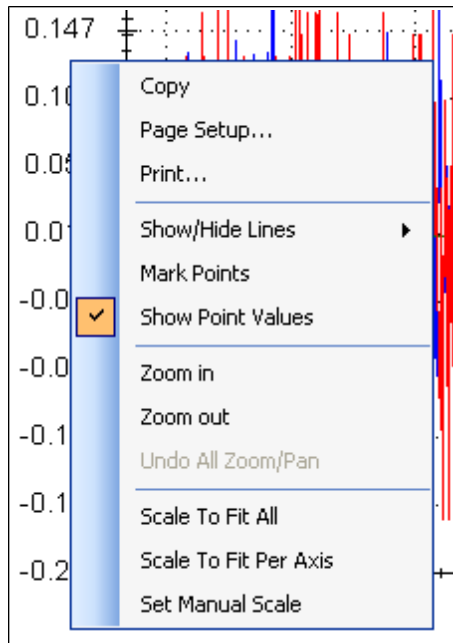
The scope provides two mechanisms to allow the you to visualize the data:

- Scaling: you can choose the scale for the different axes.
- Zooming: you can choose a particular portion of the scope that you want to observe more in details, and then come back to previous scaling.

Two different scaling modes are provided on each axis:

- Manual: you can determine the minimum and maximum value of the axis (X or Y axis).
- Scale to fit: the program will compute a scale for this axis that will display all the curves bound to it (X or Y axis).

These functionalities are accessible through the contextual menu when right-clicking in the axis zone. A simple left-click in the axis zone will provide the manual range functionality. A supplementary functionality allows you to perform a scale to fit on all axes is also available, which allows a good overview.

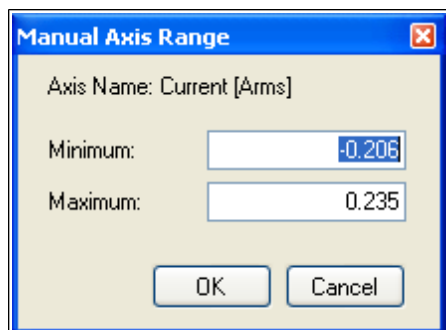


The zoom functionality allows you to navigate in a portion of the graphic. When you reset the zoom, the initial scales are shown.

In the display tab, when “Remember Axis Scale” is set, the scales of the axes are kept between two sequential recordings. You can fine tune the scale to visualize a particular behavior and record a second time and see the same behaviour without having to redo all the tuning. When not checked, a scale to fit all will be performed after each record. This setting is reset when exiting WorkBench and should be explicitly set at next startup.

15.3.7 Manual range per axis

Click **Manual Y Axis** to open a dialog box to set the range for the axis. Enter the Y-axis minimum value and Y-axis maximum value. Click **OK** to reset the Y-axis to new range.



15.3.8 Unit display on Y axis

The unit on the Y-axis is displayed if all scope signals units are identical for that Y-axis. If different units apply to different signals, the units are displayed as [-]. For example, if the velocity Y-axis has signals VL.FB and IL.CMD, then the unit displayed is [-], since the units for these parameters are different. If IL.CMD is hidden, then the correct unit for VL.FB, rpm, is displayed.

Related Parameters:

BODE Parameters

This page intentionally left blank.

16 Using Parameters and the Terminal Screen

| | |
|---|------------|
| 16.1 Terminal | 218 |
| 16.2 Viewing Parameters | 221 |
| 16.3 Parameter List | 221 |
| 16.4 Parameter Load/Save | 221 |
| 16.5 Parameter Comparer | 221 |
| Summary of Parameters and Commands | 225 |

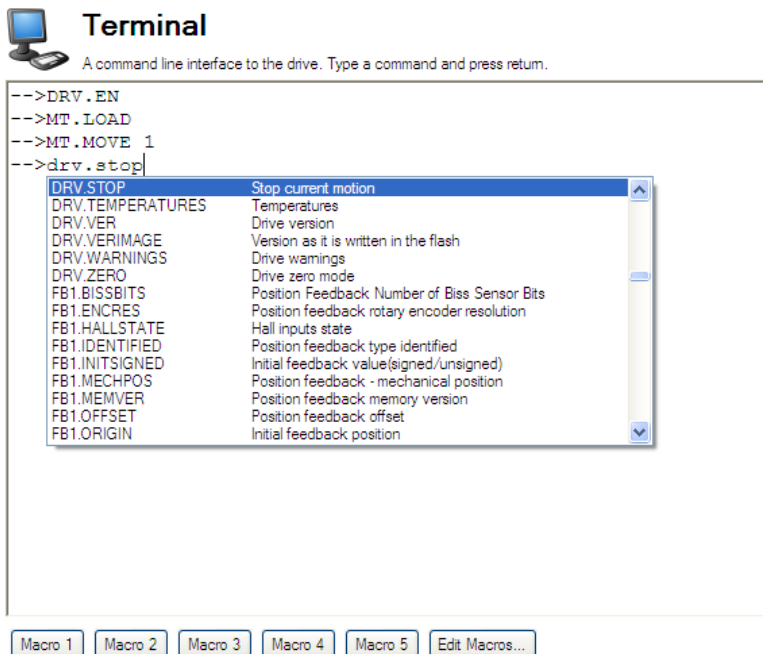
16.1 Terminal

16.1.1 Overview

The terminal mode provides a quick and easy way to issue commands to the drive directly. Typically used by "power users" who are familiar with the command set, the terminal mode can help in setup, troubleshooting, and other diagnostic actions. When using the terminal mode, WorkBench shows the parameter and command set in a popup view and uses an autocomplete tool to help you select the proper parameter or command. Right-click in the command entry area to open a popup menu for editing commands and for clearing the screen. The terminal also provides a macro editor that allows a series of commands to be executed via a single command (called a macro). Macros are useful when you must frequently execute a sequence of commands.

16.1.2 Using the Terminal

Click Terminal in the navigation tree to issue parameters and commands in the terminal mode. You can enter parameters and commands at the prompt as shown in this example:



The terminal supports the following keyboard shortcuts:

| Keyboard Shortcut | Description |
|-------------------|--|
| F2 | Executes the last command. |
| Up Arrow | Gets the previous command from the command history. |
| Down Arrow | Gets the next command from the command history. |
| CTRL+J | Shows the list of commands that the drive supports. |
| CTRL+L | Opens the Parameter Load/Save view. |
| ESC | Hides the command list if command list is open. Clears the line if command list is not open. |
| Enter (Return) | Executes the current command. This command sends the text you have typed to the drive and then prints the text the drive returns on the next line. |

16.1.3 Macros

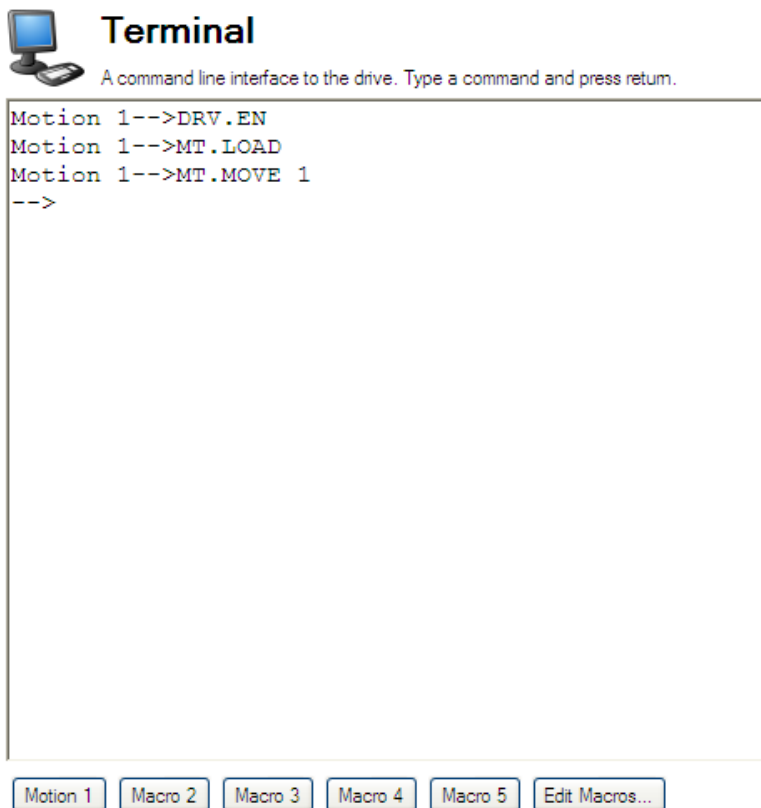
Macros are a short sequence of instructions that can be sent to the drive using a single button. You can create up to five macros within the terminal mode using the macro editor. Each of the macros can be given a name and will appear as a button below the Terminal screen area. When you select the button, the command set tied to that macro will appear in the Terminal area and the commands will be executed. If there are commands which are not entered correctly, the terminal screen area will indicate the errors.

You can access macros and the macro editor from the buttons are located at the bottom of the terminal screen. Click a macro button to execute the associated macro. The **Edit Macros** button opens the macro editor used to create and edit macros. You can also assign custom names to the macro buttons with this editor.

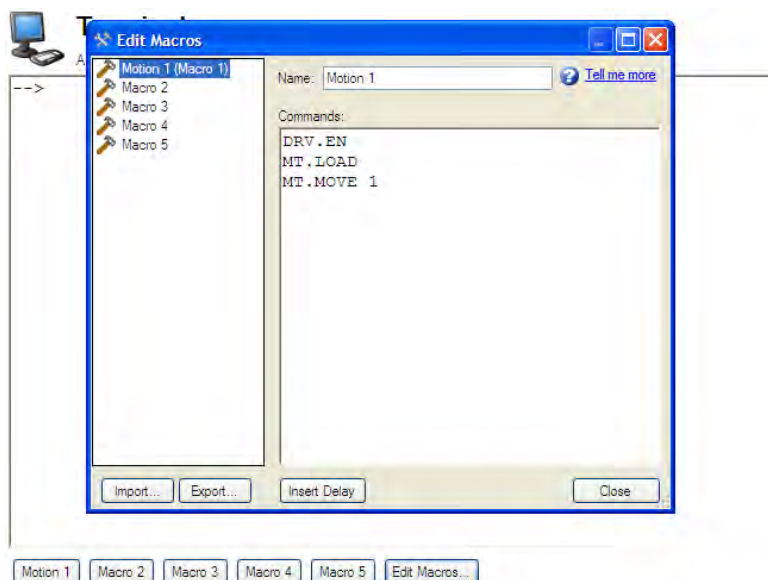
16.1.3.1 Creating a Macro from Terminal commands

You can create a macro from a sequence of terminal commands as follows:

1. Enter the sequence of commands.



2. Copy the sequence of commands, then click **Edit Macro**. Select a macro from the tree on the left, then paste the sequence of commands into the Commands area of the Edit Macros window.



3. Select **Save** to save your macro.

16.1.3.2 Macro Editor

The macro editor allows you to create and modify up to five macros. WorkBench automatically saves the macros that you create. If you exit WorkBench and then start WorkBench again, the macros you defined will still be available.

To create or edit a macro from the editor, open the **Terminal** screen and click on **Edit Macros** (located at the bottom of the screen). Select the macro you wish to edit from the tree on the left, then use the features described below to build the macro. You can add comments in the macro after the ";" symbol.

| Button or Dialog Box | Description |
|----------------------|--|
| Name | Allows you to give each macro a unique name. This name will be shown on the button in the terminal view. |
| Commands | Displays the commands that are sent to the drive when you use this macro. The response from the drive is displayed in the terminal. |
| Import | Overwrites the selected macro with the contents of a macro file exported from another computer. |
| Export | Sends the selected macro to a text file so that you can import it into WorkBench running on another computer. |
| Insert Delay | Inserts a step into the macro that causes a delay for a specified number of milliseconds before continuing. A line that starts with #delay 5000 will pause the execution of the macro for 5000 milliseconds. |
| Close | Closes this window and returns to the terminal view. |

16.2 Viewing Parameters

You can view and edit parameters in the **Parameters** screen. You can view and write parameters in the **Terminal** screen.

16.3 Parameter List

This screen displays a list of the current values of all the parameters that the drive supports. You can sort some of the rows by clicking the column headers.

| Button or Dialog Box | Description |
|----------------------|--|
| Refresh | Reads all the parameters from the drive and update the contents of the table. |
| Print | Sends the data that you see on the screen to the printer. |
| eMail | Opens an email message and attaches a comma separated file to the email message. |
| More/Less | Adds two more columns to the table with the range and default values for each parameter. |

16.4 Parameter Load/Save

To copy a configuration (all the drive parameters that are stored in the drives nonvolatile memory) from one drive to another, click the **Save to File** button on the first drive and then use the **Load from File** on the second drive. If you wish the second drive to keep these new parameters after the drive is turned off, then you can save the parameters to the nonvolatile memory with **Save To Drive**.

| Button or Dialog Box | Description |
|------------------------|--|
| Save To File | Copies all the drive parameters and creates a file. |
| Load From Drive | Sets all the drive parameters within the drive to the values contained within the file you select. |
| Save to Drive | Saves all drive parameters into the nonvolatile memory of the drive. Each time the drive powers on, it will start with these saved parameters. |
| Default | Returns all the drive parameters back to their default values. Any changes you have made are lost. |

16.5 Parameter Comparer

The **Parameter Comparer** tool is used to compare parameters from different drives and to compare motion tasks of different drives. You can select this tool from the menu bar, **Tools> Parameter Comparer**.

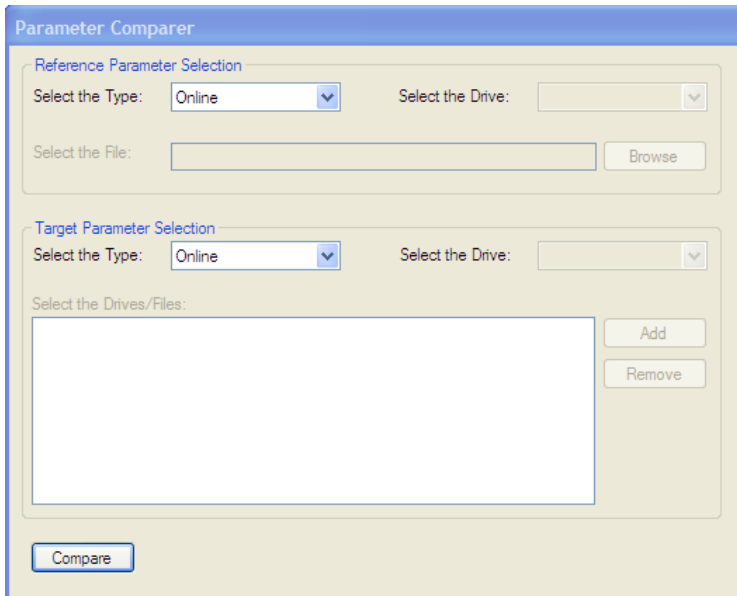
Parameters can be compared between the following:

- Drives (Online / Offline)
- Drives and Files
- File and Drives
- File and Files

The **Parameter Comparer** screen includes the three sections shown below:

- Reference Parameter Selection
- Target Parameter Selection

- Display Comparison

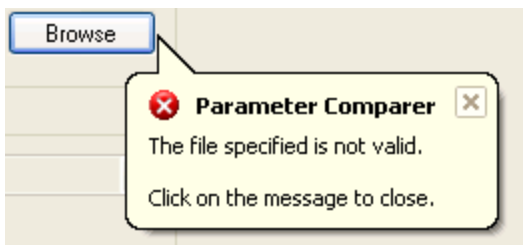


16.5.1 Reference Parameter Selection

You can select the drive type (Online/Offline) and file to compare in the **Reference Parameter Selection** area. When a drive type (Online/Offline) is selected, the drives connected to the WorkBench are displayed in the **Select the drive** box. By default, the first drive is selected. If no drives are connected, then the **Select the drive** box will be in disabled state.

If **File Selection** is chosen from the **Select the Type** box, then the **Select the drive** box is disabled and the **Select the File** box and **Browse** button become enabled for the user to browse the parameter file (*.akd).

A warning message is displayed if you select an incorrect file, as shown below:



16.5.2 Target Parameter Selection

Target Parameter Selection includes the following:

- **Select the Type** box to select the type (drive (online/offline) and File).
- **Select the Drive** box to select the drives connected to the WorkBench.
- **Select the Drives/Files** box to load files or drives which are to be compared.
- **Add** button to add Drives/Files selected to the list box.
- **Remove** button to remove the selected item from the list box.

Many target drives/files parameters can be compared with one reference drive/file parameters as shown below.

Parameter Comparer

Reference Parameter Selection

Select the Type: Select the Drive:

Select the File:

Target Parameter Selection

Select the Type: Select the Drive:

Select the Drives/Files:

| | |
|-------|---------------------------------------|
| One | <input type="button" value="Add"/> |
| Two | <input type="button" value="Remove"/> |
| Three | |

Parameter Comparer

Reference Parameter Selection

Select the Type: Select the Drive:

Select the File:

Target Parameter Selection

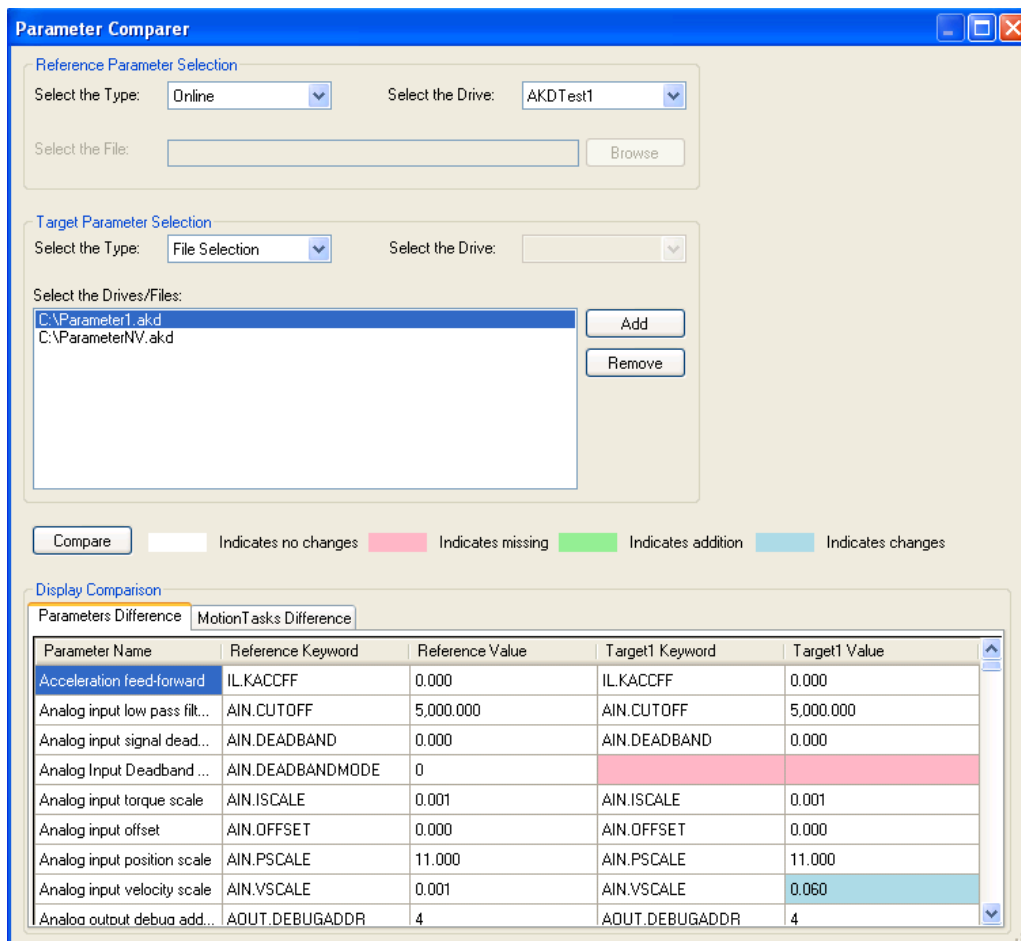
Select the Type: Select the Drive:

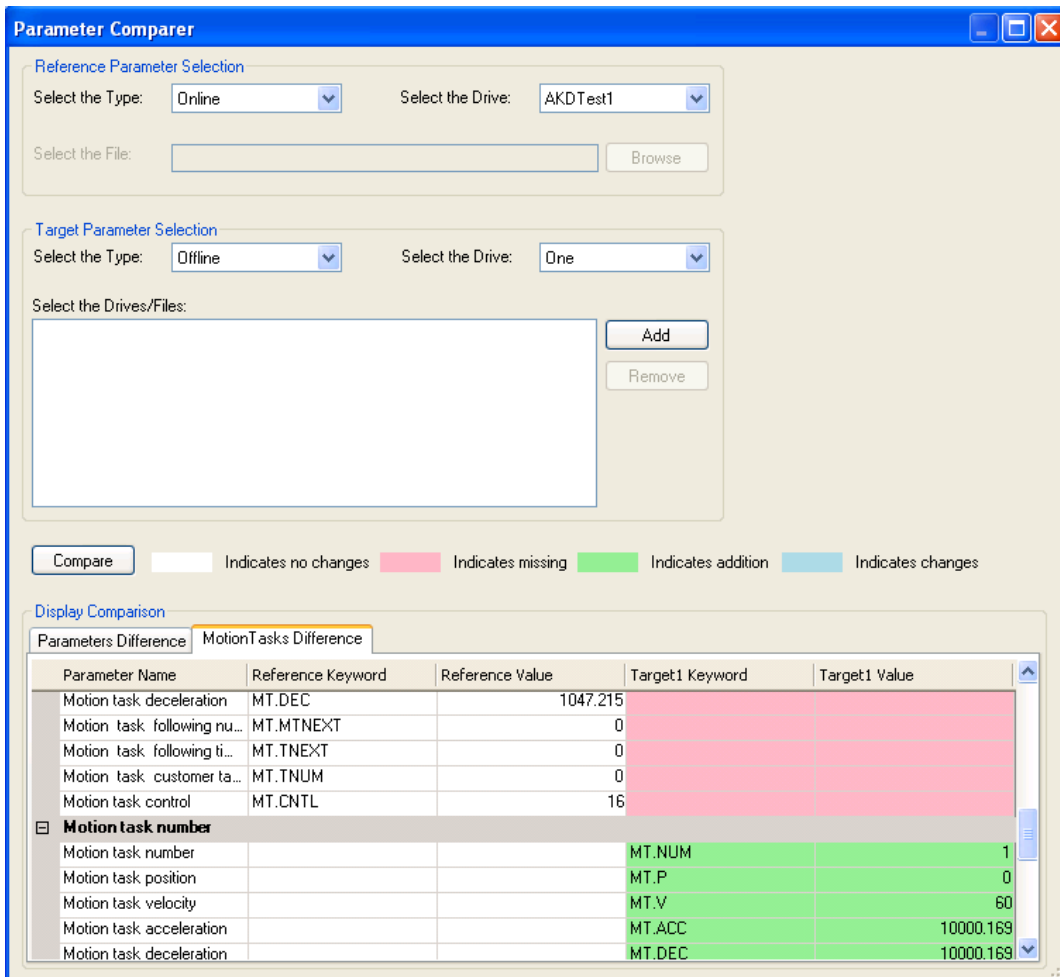
Select the Drives/Files:

| | |
|--------------------|---------------------------------------|
| C:\Parameter1.akd | <input type="button" value="Add"/> |
| C:\ParameterNV.akd | <input type="button" value="Remove"/> |

16.5.3 Display the comparison

Clicking **Compare** displays the comparison shown below:





A separate section is shown for each motion task.

Motion Tasks Difference and **Parameters Difference** are color coded as shown in the table below.

| | |
|--|---|
| | Keyword exists in the reference file /drive and does not exist in the Target files/drives. |
| | Parameter does not exist in reference file/drive and exists in Target files/drives. |
| | Keyword exists in both reference and target files/drives, and the value or the parameter keyword changes. |
| | Reference and target parameter do not vary. |

If you click the **Compare** button without selecting the drive or file, you will receive a message prompting you to make a selection.

Summary of Parameters and Commands

This table contains an alphabetical list of parameters and commands, with a brief description for each. The parameter name and description are linked to the parameter tables. The description field also notes if a parameter is not active in all opmodes. Generally speaking, all parameters and commands are active in all opmodes, with the following exceptions:

| Parameter or Command | Active in Opmodes |
|------------------------------------|-------------------|
| GEAR (all parameters and commands) | 2 (position) only |
| HOME (all parameters and commands) | 2 (position) only |
| MT (all parameters and commands) | 2 (position) only |
| SM.I1, SM.I | 0 (torque) only |

| Parameter or Command | Active in Opmodes |
|----------------------------------|------------------------------------|
| SM.V1, SM.V2 | 1 (velocity) only |
| SM.VPM1, SM.VPM2 | 2 (position) only |
| VL (all parameters and commands) | 1 (velocity) and 2 (position) only |

| Parameter or Command | Type | Description |
|----------------------------------|---------|--|
| Analog Input (AIN) | | |
| AIN.CUTOFF | NV | Sets the analog input low-pass filter cutoff frequency. |
| AIN.DEADBAND | NV | Sets the analog input signal deadband. |
| AIN.DEADBANDMODE | NV | Sets the analog input deadband mode. |
| AIN.ISCALE | NV | Sets the analog current scale factor. |
| AIN.OFFSET | NV | Sets the analog input offset. |
| AIN.PSCALE | NV | Sets the analog position scale factor. |
| AIN.VALUE | R/O | Reads the value of the analog input signal. |
| AIN.VSCALE | NV | Sets analog velocity scale factor. |
| AIN.ZERO | Command | Zeroes the analog input signal. |
| Analog Input/Output (AIO) | | |
| AIO.ISCALE | NV | Sets the analog current scale factor. |
| AIO.VSCALE | NV | Sets velocity scale factor. |
| AIO.PSCALE | NV | Sets position scale factor. |
| Analog Output (AOUT) | | |
| AOUT.DEBUGADDR | NV | Sets the memory address to debug. |
| AOUT.DEBUGDATATYPE | NV | Sets the data type of the value to be debugged. |
| AOUT.DEBUGSCALE | NV | Sets the scale to be used for debug. |
| AOUT.ISCALE | NV | Sets the analog current scale factor. |
| AOUT.MODE | NV | Sets the analog output mode. |
| AOUT.OFFSET | NV | Sets the analog input offset. |
| AOUT.PSCALE | NV | Sets the analog position scale factor. |
| AOUT.VALUE | NV | Reads the analog output value. |
| AOUT.VALUEEU | R/W | Sets the analog output value. |
| AOUT.VSCALE | NV | Sets velocity scale factor for analog output. |
| Bode plot (BODE) | | |
| BODE.EXCITEGAP | R/W | Controls how often the excitation is updated. |
| BODE.FREQ | R/W | Sets the frequency of the sine excitation source. |
| BODE.IAMP | R/W | Sets current command value used during the Bode procedure. |
| BODE.IFLIMIT | | Sets the current fault duration limit in seconds for the BODE.MODE 5 stability test. |
| BODE.IFTHRESH | | Sets the current fault threshold for the BODE.MODE 5 stability test. |
| BODE.INJECTPOINT | R/W | Sets whether the excitation uses current or velocity excitation type. |
| BODE.MODE | R/W | Sets the mode of the excitation. |
| BODE.MODETIMER | R/W | Sets the watchdog timer of the excitation. |
| BODE.PRBDDEPTH | R/W | Sets the length of the PRB signal before it repeats. |
| BODE.VAMP | R/W | Sets the amplitude of the excitation when in velocity mode. |

| Parameter or Command | Type | Description |
|--------------------------------|------|---|
| BODE.VFLIMIT | | Sets the velocity fault duration limit (seconds) for the BODE.MODE 5 stability test |
| BODE.VFTHRESH | | Sets the current fault threshold for the BODE.MODE 5 stability test. |
| Capture (CAP) | | |
| CAP0.EDGE, CAP1.EDGE | R/W | Selects the capture edge. |
| CAP0.EN, CAP1.EN | R/W | Enables or disables the related capture engine. |
| CAP0.EVENT, CAP1.EVENT | | Controls the precondition logic. |
| CAP0.FILTER, CAP1.FILTER | R/W | Controls the precondition logic. |
| CAP0.MODE, CAP1.MODE | NV | Selects the captured value. |
| CAP0.PLFB, CAP1.PLFB | R/O | Reads captured position value. |
| CAP0.PREEDGE, CAP1.PREEDGE | | Selects the capture precondition edge. |
| CAP0.PREFILTER, CAP1.PREFILTER | R/W | Sets the filter for the precondition input source. |
| CAP0.PRESELECT, CAP1.PRESELECT | R/W | Sets the precondition trigger. |
| CAP0.STATE, CAP1.STATE | R/O | Indicates whether or not trigger source was captured. |
| CAP0.T, CAP1.T | R/O | Reads time capture (if time capture was configured). |
| CAP0.TRIGGER, CAP1.TRIGGER | R/W | Specifies the trigger source for the position capture. |
| Controlled Stop (CS) | | |
| CS.DEC | NV | Sets the deceleration value for the controlled stop process. |
| CS.STATE | NV | Returns the internal status of the controlled stop process. |
| CS.TO | NV | Sets the time value for the drive velocity to be within CS.VTHRESH. |
| CS.VTHRESH | NV | Sets the velocity threshold for the controlled stop. |
| Digital Input (DIN) | | |
| DIN.ROTARY | R/O | Reads the rotary knob value. |
| DIN.STATES | R/O | Reads the digital input states. |
| DIN1.FILTER TO DIN7.FILTER | R/W | Filter mode for digital inputs 1 to 7. |
| DIN1.INV TO DIN7.INV | R/W | Sets the indicated the polarity of a digital input mode. |
| DIN1.MODE TO DIN7.MODE | NV | Sets the digital input modes. |
| DIN1.PARAM TO DIN7.PARAM | R/W | Sets a value used as an extra parameter for digital inputs nodes. |
| DIN1.STATE TO DIN7.STATE | R/O | Reads a specific digital input state. |
| Digital Output (DOUT) | | |
| DOUT.CTRL | NV | Sets the source of digital outputs (firmware or fieldbus). |
| DOUT.RELAYMODE | R/W | Indicates faults relay mode. |
| DOUT.STATES | R/O | Reads the state of the two digital outputs. |
| DOUT1.MODE AND DOUT2.MODE | NV | Sets the digital output mode. |

| Parameter or Command | Type | Description |
|-------------------------------------|---------|--|
| DOUT1.PARAM AND DOUT2.PARAM | NV | Sets extra parameters for the digital outputs. |
| DOUT1.STATE AND DOUT2.STATE | R/O | Reads the digital output state. |
| DOUT1.STATEU AND DOUT2.STATEU | R/W | Sets the state of the digital output node. |
| Drive (DRV) | | |
| DRV.ACC | NV | Describes the acceleration ramp for the velocity central loop. |
| DRV.ACTIVE | R/O | Reads the enable status of an axis. |
| DRV.BLINKDISPLAY | Command | Causes the display to blink for 10 seconds. |
| DRV.BURNINOFFT (Password Protected) | R/W | Sets the burn-in test mode off time. |
| DRV.CLRFAULTHIST | Command | Clears the fault history log in the NV. |
| DRV.CLRFAULTS | Command | Tries to clear all active faults in the drive. |
| DRV.CMDDELAY | R/W | Issues a delay before next command is executed. |
| DRV.CMDSOURCE | NV | Sets the command source (service, fieldbus, analog input, gearing, digital, or Bode). |
| DRV.CRASHDUMP | Command | Retrieves diagnostic information after the drive crashes. |
| DRV.DBILIMIT | NV | Sets the maximum amplitude of the current for dynamic braking. |
| DRV.DEC | NV | Sets the deceleration value for the velocity loop. |
| DRV.DIR | R/W | Changes drive direction. |
| DRV.DIS | Command | Disables the axis (software). |
| DRV.DISMODE | NV | Selects among disable immediately or stop and then disable options. |
| DRV.DISSOURCES | R/O | Returns the possible reason for a drive disable. |
| DRV.DISTO | R/W | Sets the emergency timeout |
| DRV.EMUEDIR | R/W | Sets the direction of the emulated encoder output (EEO) signal. |
| DRV.EMUEMODE | R/W | Sets the mode of the emulated encoder output (EEO) connector. |
| DRV.EMUEMTURN | R/W | Defines the location of the index pulse on the EEO (emulated encoder output) when DRV.EMU-EMODE=2. |
| DRV.EMUERES | R/W | Sets the resolution of the EEO (emulated encoder output). |
| DRV.EMUEZOFFSET | R/W | Sets the location of the EEO (emulated encoder output) index pulse (when DRV.EMU-EMODE=1). |
| DRV.EN | Command | Enables the axis (software). |
| DRV.ENDEFAULT | R/W | Sets the default state of the software enable. |
| DRV.FAULTHIST | R/O | Reads the last 10 faults from NV memory. |
| DRV.FAULTS | R/O | Reads the active faults. |
| DRV.FLASHREAD (Password Protected) | R/O | Reads a value from the serial flash memory. |
| DRV.HANDWHEEL | R/O | Reads the EEO input value. |

| Parameter or Command | Type | Description |
|-------------------------|---------|---|
| DRV.HELP | R/O | Reads the minimum, maximum, and default values for a specific parameter or command. |
| DRV.HELPALL | R/O | Retrieves the minimum, maximum, default, and actual values for all available parameters and commands. |
| DRV.ICONT | R/O | Reads the continuous rated current value. |
| DRV.INFO | R/O | Reads general information about the drive. |
| DRV.IPEAK | R/O | Reads the peak rated current value. |
| DRV.IZERO | R/W | Sets the current that will be used during the DRV.ZERO procedure. |
| DRV.LIST | R/O | Reads the list of available parameters and commands. |
| DRV.LOGICVOLTS | | Reads the logic voltages. |
| DRV.NAME | NV | Sets and reads the name of the drive. |
| DRV.NVLIST | R/O | Lists the NV parameters and values from the RAM. |
| DRV.NVLOAD | W/O | Loads all data from the NV memory of the drive into the RAM parameters. |
| DRV.NVSAVE | Command | Saves the drive parameters from the RAM to the NV memory. |
| DRV.ONTIME | R/O | Returns how long the drive has been running since last power up. |
| DRV.OPMODE | NV | Sets the drive operation mode (current, velocity, or position). |
| DRV.READFORMAT | R/W | Sets the value returned to either decimal or hexadecimal. |
| DRV.RSTVAR | Command | Sets default values in the drive without re-booting the drive and without resetting the NV memory. |
| DRV.RUNTIME | R/O | Returns how long the drive has been running since first activated. |
| DRV.SETUPREQBITS | R/O | Reads the bitwise set status of parameters that must be set before the drive can be enabled |
| DRV.SETUPREQLIST | R/O | Reads the list of parameters that must be set before the drive can be enabled. |
| DRV.STOP | Command | This command stops all drive motion. |
| DRV.TEMPERATURES | R/O | Reads the temperature of drive components. |
| DRV.VER | R/O | Reads the drive version. |
| DRV.VERIMAGE | R/O | Returns the version data from each image. |
| DRV.TEMPERATURES | R/O | Reads the temperature of drive components. |
| DRV.WARNINGS | R/O | Reads the active warnings. |
| DRV.ZERO | R/W | Sets the zero mode. The procedure is activated when the drive is enabled. |
| Feedback 1 (FB1) | | |
| FB1.ENCRES | NV | Sets the resolution of the motor encoder. |
| FB1.HALLSTATE | R/O | Reads the Hall switch values (encoder feedback |
| FB1.HALLSTATEU | R/O | Reads the state of Hall switch U. |
| FB1.HALLSTATEV | R/O | Reads the state of Hall switch V. |
| FB1.HALLSTATEW | R/O | Reads the state of Hall switch W. |

| Parameter or Command | Type | Description |
|--------------------------------|------|--|
| FB1.IDENTIFIED | R/O | Reads the type of feedback device used by the drive/motor. |
| FB1.INITSIGNED | NV | Sets initial feedback value as signed or unsigned. |
| FB1.LDLL | R/O | Reads the motor line-to-line inductance from the FPGA. |
| FB1.LQLL | R/O | Reads the motor line-to-line inductance in the back EMF axis (q axis) from the SFD memory from the FPGA. |
| FB1.MECHPOS | R/O | Reads the mechanical position. |
| FB1.MEMDUMP | W/O | Dumps the memory ID data values of a feedback with memory |
| FB1.MEMVER | R/O | Returns the memory feedback version. |
| FB1.OFFSET | NV | Sets position feedback offset. |
| FB1.POLES | R/O | Reads the number of feedback poles. |
| FB1.OFFSET | NV | Sets position feedback offset. |
| FB1.POLES | R/O | Reads the number of feedback poles. |
| FB1.SELECT | NV | Sets user entered type or identified type (-1). |
| Feedback 2 (FB2) | | |
| FB2.ENCRESP | NV | Sets the secondary feedback (FB2) resolution. |
| FB2.MODE | R/W | Sets the mode for the second feedback inputs, EEO connector (X9) and high speed opto inputs (pins 9 and 10 on X7). |
| FB2.SOURCE | R/W | Sets the source for the second feedback input. Choices are the EEO connectors (X9) which are RS485 inputs, or the X7 connector's high speed opto inputs (pins 9 and 10). |
| Fieldbus (FBUS) | | |
| FBUS.PARAM1 TO FBUS.PARAM20 | NV | Set fieldbus specific meanings. |
| FBUS.PLLSTATE | R/O | Returns the status of the PLL |
| FBUS.PLLTHRESH | NV | Sets number of successful synchronized cycles needed to lock the PLL. |
| FBUS.SAMPLEPERIOD | NV | Sets fieldbus sample period. |
| FBUS.SYNCACT | R/O | Reads actual distance from the desired sync distance. |
| FBUS.SYNCDIST | NV | Sets time target for synchronization. |
| FBUS.SYNCWIND | NV | Sets symmetrically arranged window around the desired sync distance. |
| FBUS.TYPE | R/O | Shows the active fieldbus type. |
| Gearing (GEAR) | | |
| GEAR.ACCMAX | R/W | Sets the maximum allowed acceleration value; active in opmode 2 (position) only. |
| GEAR.DECMAX | R/W | Sets the maximum allowed deceleration value; active in opmode 2 (position) only. |
| GEAR.IN | R/W | Sets the denominator of the electronic gearing ratio; active in opmode 2 (position) only. |
| GEAR.MODE | R/W | Selects electronic gearing mode; active in opmode 2 (position) only. |

| Parameter or Command | Type | Description |
|-------------------------------------|---------|--|
| GEAR.MOVE | Command | Starts the electronic gearing; active in opmode 2 (position) only. |
| GEAR.OUT | R/W | Sets the numerator of the electronic gearing ratio; active in opmode 2 (position) only. |
| GEAR.VMAX | R/W | Reads the maximum allowed velocity value; active in opmode 2 (position) only. |
| Homing (HOME) | | |
| HOME.ACC | R/W | Sets homing acceleration; active in opmode 2 (position) only. |
| HOME.AUTOMOVE | R/W | Sets homing auto move flag. |
| HOME.DEC | R/W | Sets homing deceleration; active in opmode 2 (position) only. |
| HOME.DIR | NV | Sets homing direction; active in opmode 2 (position) only. |
| HOME.DIST | R/W | Sets homing distance; active in opmode 2 (position) only. |
| HOME.FEEDRATE | R/W | Sets homing velocity factor; active in opmode 2 (position) only. |
| HOME.IPEAKHOME.IPEAK | R/W | Sets the current limit during homing procedure to a mechanical stop; active in opmode 2 (position) only. |
| HOME.MODEHOME.MODE | R/W | Selects the homing mode; active in opmode 2 (position) only. |
| HOME.MOVEHOME.MOVE | Command | Starts a homing procedure; active in opmode 2 (position) only. |
| HOME.PHOME.P | R/W | Sets home position; active in opmode 2 (position) only. |
| HOME.PERRTHRESH HOME.PERRTHRESH | R/W | Sets the position lag threshold; active in opmode 2 (position) only. |
| HOME.REQUIRE | NV | Defines if the axis must be homed before a motion task can be executed. |
| HOME.SETHOME.SET | Command | Immediately sets the home position; active in opmode 2 (position) only. |
| HOME.VHOME.V | R/W | Sets homing velocity; active in opmode 2 (position) only. |
| Hardware Limit Switch (HWLS) | | |
| HWLS.NEGSTATE | R/O | Reads the status of the negative hardware limit switch. |
| HWLS.POSSTATE | R/O | Reads the status of the positive hardware limit switch. |
| Current Loop (IL) | | |
| IL.BUSFF | R/O | Displays the current feedforward value injected by the fieldbus. |
| IL.CMD | R/O | Reads the value of the q-component current controller inside the FPGA. |
| IL.CMDU | R/W | Sets the user current command. |
| IL.DCMD (Password Protected) | R/O | Reads the value of the d-component current controller inside the FPGA. |
| IL.DCMD (Password Protected) | R/O | Reads the value of the d-component current controller inside the FPGA. |
| IL.DCMDU (Password Protected) | R/W | Sets user d-component current command. |

| Parameter or Command | Type | Description |
|----------------------------------|------|---|
| IL.DEADBAND (Password Protected) | R/O | Reads the deadband of two IGBTs in series connection. |
| IL.FB | R/O | Reads the actual value of the d-component current. |
| IL.DFOLDD (Password Protected) | R/O | Reads the motor foldback maximum time at motor peak current. |
| IL.DFOLDR (Password Protected) | R/O | Reads the motor foldback recovery time. |
| IL.DFOLDT | R/O | Reads the motor foldback time constant of the exponential current drop (foldback). |
| IL.DIFOLD | R/O | Reads the drive foldback current limit. |
| IL.DLIMITN | R/W | Sets the negative user (application-specific) d-component current limit. |
| IL.DLIMITP | R/W | Sets the positive user (application-specific) d-component current limit. |
| IL.FB | R/O | Reads the actual value of the d-component current. |
| IL.FOLDFTHRESH | NV | Reads the foldback fault level. |
| IL.FOLDWTHRESH | NV | Sets the foldback warning level. |
| IL.IFOLD | R/O | Reads the overall foldback current limit. |
| IL.INTEN (Password Protected) | NV | Enables/disables the integrator part of the PI loop. |
| IL.IUOFFSET (Password Protected) | R/W | Sets the offset added to the sigma-delta measured current value in the u-winding. |
| IL.IUFB | R/O | Reads the sigma-delta measured current in the u-winding of the motor. |
| IL.IVFB | R/W | Sets the sigma-delta measured current in the u-winding of the motor. |
| IL.IVOFFSET (Password Protected) | R/W | Sets the offset added to the sigma-delta measured current value in the v-winding. |
| IL.KP | NV | Sets the proportional gain of the q-component of the PI regulator. |
| IL.KPDRATIO | NV | Sets the proportional gain of the d-component current PI-regulator as a percentage of IL.KP |
| IL.KPLOOKUPINDEX | R/W | Sets the index into the Current Loop Gain Scheduling Table. |
| IL.KPLOOKUPVALUE | R/W | Sets the value of the current loop gain scheduling index. |
| IL.KPLOOKUPVALUES | R/W | Gets the Current Loop Gain Scheduling Table. |
| IL.KVFF | R/W | Current loop velocity feed-forward gain. |
| IL.LIMITN | NV | Sets the negative user (application-specific) current limit. |
| IL.LIMITP | NV | Sets the positive user (application-specific) current limit. |
| IL.MFOLDD | NV | Sets the motor foldback maximum time at motor peak current. |
| IL.MFOLDR | R/O | Sets the motor foldback recovery time. |
| IL.MFOLDT | NV | Sets the motor foldback time constant of the exponential current drop (foldback). |
| IL.MIFOLD | R/O | Sets the motor foldback current limit. |

| Parameter or Command | Type | Description |
|---------------------------------|------|--|
| IL.PWMFREQ (Password Protected) | NV | Reads and sets the PWM frequency of the IGBTs. |
| IL.VDCMD | R/O | Sets the output of the d-component PI-regulator. |
| IL.VCMD | R/O | Sets the output of the q-component PI regulator. |
| IL.VUFB | R/O | Reads the measured voltage on the u-winding of the motor. |
| IL.VVFB | R/O | Reads the measured voltage on the v-winding of the motor. |
| IL.DLIMITN | R/W | Sets the negative user (application-specific) d-component current limit. |
| IL.DLIMITP | R/W | Sets the positive user (application-specific) d-component current limit. |
| | | |
| IL.FB | R/O | Reads the actual value of the d-component current. |
| LOAD.INERTIA | NV | Sets the load inertia. |

| Motor (MOTOR) Parameters | | |
|--------------------------|---------|---|
| MOTOR.AUTOSET | NV | Determines which drive parameters are calculated automatically. |
| MOTOR.BRAKE | NV | Sets the presence or absence of a motor brake. |
| | | |
| MOTOR.ICONT | NV | Sets the motor continuous current. |
| MOTOR.INERTIA | NV | Sets the motor inertia. |
| MOTOR.IPEAK | NV | Sets the motor peak current. |
| MOTOR.KE | | Sets the motor back EMF constant. |
| MOTOR.KT | NV | Sets the torque constant of the motor. |
| MOTOR.LQLL | NV | Sets the line-to-line motor Lq. |
| MOTOR.VMAX | NV | Sets the maximum motor speed. |
| MOTOR.NAME | NV | Sets the motor name. |
| MOTOR.PHASE | NV | Sets the motor phase. |
| MOTOR.PITCH | NV | Sets the motor pitch. |
| MOTOR.POLES | NV | Sets the number of motor poles. |
| MOTOR.R | NV | Sets the stator winding resistance phase-phase in ohms. |
| MOTOR.TBRAKEAPP | NV | The delay time used for applying the motor brake. |
| MOTOR.TBRAKERLS | NV | The delay time used for releasing the motor brake. |
| MOTOR.TEMP | R/O | Reads the motor temperature represented as the resistance of the motor PTC. |
| MOTOR.TEMPFAULT | NV | Sets the motor temperature fault level. |
| MOTOR.TEMPWARN | NV | Sets the motor temperature warning level. |
| MOTOR.TYPE | NV | Sets the motor type. |
| MOTOR.VOLTMAX | NV | Sets the motor maximum voltage. |
| Motion Task (MT) | | |
| MT.ACC | R/W | Specifies motion task acceleration; active in opmode 2 (position) only. |
| MT.CLEAR | Command | Clears motion tasks from the drive; active in opmode 2 (position) only. |

| | | |
|---------------------------|---------|--|
| MT.CNTL | R/W | Sets motion task control word; active in opmode 2 (position) only. |
| MT.CONTINUE | Command | Continues a stopped motion task; active in opmode 2 (position) only. |
| MT.DEC | R/W | Sets motion task deceleration; active in opmode 2 (position) only. |
| MT.EMERGMT | R/W | Selects a motion task to be triggered after an emergency stop procedure; active in opmode 2 (position) only. |
| MT.HOMEREQUIRE | NV | Defines if the axis is required to be homed before a motion task can be executed. |
| MT.LIST | Command | Lists all initialized motion tasks in the drive; active in opmode 2 (position) only. |
| MT.MTNEXT | R/W | Specifies following motion task number; active in opmode 2 (position) only. |
| MT.TNEXT | R/W | Specifies following motion task time; active in opmode 2 (position) only. |
| MT.LIST | Command | Lists all initialized motion tasks in the drive; active in opmode 2 (position) only. |
| MT.LOAD | Command | Reads/loads a motion task number from the drive; active in opmode 2 (position) only. |
| MT.MOVE | Command | Starts a motion task; active in opmode 2 (position) only. |
| MT.NUM | R/W | Sets the motion task number; active in opmode 2 (position) only. |
| MT.PARAMS | Command | Shows a motion task; active in opmode 2 (position) only. |
| MT.P | R/W | Sets the motion task position; active in opmode 2 (position) only. |
| MT.SET | Command | Sets the motion task in the drive; active in opmode 2 (position) only. |
| MT.TNUM | R/W | Motion task customer table number. |
| MT.V | R/W | Sets the motion task velocity; active in opmode 2 (position) only. |
| Position Loop (PL) | | |
| PL.CMD | NV | Reads the position command directly from the entry to the position loop. |
| PL.ERR | NV | Reads the position error present when the drive is controlling the position loop. |
| PL.ERRFTHRESH | NV | Sets the maximum position error. |
| PL.ERRWTHRESH | NV | Sets the position error warning level. |
| PL.FB | R/O | Reads the position feedback value. |
| PL.FBSOURCE | NV | Sets the feedback source for the position loop. |
| PL.INTINMAX | NV | Limits the input of the position loop integrator by setting the input saturation. |
| PL.INTOUTMAX | NV | Limits the output of the position loop integrator by setting the output saturation. |
| PL.KI | NV | Sets the integral gain of the position loop. |
| PL.KP | NV | Sets the proportional gain of the position regulator PID loop. |

| Programmable Limit Switch (PLS) | | |
|--|---------|---|
| PLS.EN | R/W | Enables programmable limit switch (PLS). |
| PLS.MODE | NV | Selects programmable limit switch mode. |
| PLS.P1 TO PLS.P8 | NV | Sets the trigger point for programmable limit switches. |
| PLS.STATE | R/O | Reads the programmable limit switch state. |
| Recorder (REC) | | |
| REC.ACTIVE | R/O | Indicates if data recording is in progress (active). |
| REC.CH1 to REC.CH6 | R/W | Sets recording channels 1 to 6. |
| REC.DONE | R/O | Checks whether or not the recorder has finished recording. |
| REC.GAP | R/W | Specifies the gap between consecutive samples. |
| REC.NUMPOINTS | R/W | Sets the number of points to record. |
| REC.OFF | R/W | Turns the recorder OFF. |
| REC.RETRIEVE | R/O | Transfers all the recorded data to the communication channel. |
| REC.STOPTYPE | R/W | Sets the recorder stop type. |
| REC.TRIG | Command | Triggers the recorder. |
| REC.TRIGSLOPE | R/W | Sets the trigger slope. |
| REC.TRIGTYPE | R/W | Sets the trigger type. |
| REC.TRIGVAL | R/W | Sets the trigger value. |
| Regen Resistor (REGEN) | | |
| REGEN.POWER | R/O | READS REGEN RESISTOR'S CALCULATED POWER. |
| REGEN.REXT | N/V | SETS THE EXTERNAL, USER-DEFINED REGEN RESISTOR RESISTANCE. |
| REGEN.TEXT | R/W | Sets the external regen resistor thermal protection time constant. |
| REGEN.TYPE | N/V | SETS THE REGEN RESISTOR TYPE. |
| REGEN.WATTEXT | R/W | SETS THE REGEN RESISTOR'S POWER FAULT LEVEL FOR AN EXTERNAL REGEN RESISTOR. |
| Service Motion (SM) | | |
| SM.I1 | R/W | Sets service motion current 1; active in opmode 0 (torque) only. |
| SM.I2 | R/W | Sets service motion current 2; active in opmode 0 (torque) only. |
| SM.MODE | R/W | Sets the service motion mode. |
| SM.MOVE | Command | Starts the service motion. |
| SM.T1 | R/W | Sets the service motion time 1. |
| SM.T2 | R/W | Sets the service motion time 2. |
| SM.V1 | R/W | Sets service motion velocity 1; active in opmode 1 (velocity) and 2 (position). |
| SM.V2 | R/W | Sets service motion velocity 2; active in opmode 1 (velocity) and 2 (position). |
| Units (UNIT) | | |
| UNIT.ACCLINEAR | NV | Sets the linear acceleration/deceleration units. |
| UNIT.ACCROTARY | NV | Sets the rotary acceleration/deceleration units. |

| | | |
|---------------------------|-----|---|
| UNIT.LABEL | NV | Sets user-defined name for user-defined position units. |
| UNIT.PIN | NV | Sets gear IN for the unit conversion. |
| UNIT.PLINEAR | NV | Sets the linear position units. |
| UNIT.POUT | NV | Sets gear out for the unit conversion. |
| UNIT.PROTARY | NV | Sets the position units when the motor type (MOTOR.TYPE) is rotary. |
| UNIT.VLINEAR | NV | Sets the linear velocity units. |
| UNIT.VROTARY | NV | Sets the velocity units when the motor type (MOTOR.TYPE) is rotary. |
| Bus voltage (VBUS) | | |
| VBUS.OVFTHRESH | R/O | Reads the over voltage fault level. |
| VBUS.OVWTHRESH | N/V | Sets voltage level for over voltage warning. |
| VBUS.RMSLIMIT | R/O | Reads the limit for the bus capacitors load. |
| VBUS.UVFTHRESH | R/O | Sets the under voltage fault level. |
| VBUS.UVMODE | | Indicates undervoltage (UV) mode. |
| VBUS.UVWTHRESH | NV | Sets voltage level for undervoltage warning. |
| VBUS.VALUE | R/O | Reads DC bus voltage. |
| Velocity Loop (VL) | | |
| VL.ARPF1 TO VL.ARPF4 | R/W | Sets the natural frequency of the pole (denominator) of anti-resonance (AR) filters 1, 2, 3, and 4; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ARPQ1 TO VL.ARPQ4 | R/W | Sets the Q of the pole (denominator) of anti-resonance (AR) filter 1; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ARTYPE1 TO VL.ARTYPE4 | NV | Indicates the method used to calculate BiQuad coefficients; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ARZF1 TO VL.ARZF4 | R/W | Sets the natural frequency of the zero (numerator) of anti-resonance (AR) filter 1; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ARZQ1 TO VL.ARZQ4 | R/W | Sets the Q of the zero (numerator) of anti-resonance filter #1; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.BUSFF | R/O | Displays the velocity loop feedforward value injected by the field-bus; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.CMD | R/O | Reads the actual velocity command; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.CMDU | R/W | Sets the user velocity command; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ERR | R/O | Sets the velocity error; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.FB | R/O | Reads the velocity feedback; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.FBFILTER | R/O | Filters VL.FB value; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.FBSOURCE | NV | Sets feedback source for the velocity loop; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.FBUNFILTERED | R/O | Reads the velocity feedback. |

| | | |
|----------------------------|---------|---|
| VL.FF | R/O | Displays the velocity loop overall feedforward value; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.GENMODE | NV | Selects mode of velocity generation (Observer, d/dt); active in opmodes 1 (velocity) and 2 (position) only. |
| VL.KBUSFF | R/W | Sets the velocity loop acceleration feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.KI | NV | Sets the velocity loop integral gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.KP | NV | Sets velocity loop proportional gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.KVFF | R/W | Sets the velocity loop velocity feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.LIMITN | NV | Sets the velocity lower limit; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.LIMITP | NV | Sets the velocity high limit; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.LMJR | R/W | Sets the ratio of the estimated load moment of inertia relative to the motor moment of inertia; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.THRESH | NV | Sets the over speed fault value; active in opmodes 1 (velocity) and 2 (position) only. |
| Wake and Shake (WS) | | |
| WS.ARM | Command | Sets wake and shake to start at the next drive enable. |
| WS.DISTMIN | R/W | Sets the minimum movement required for wake and shake. |
| WS.IMAX | R/W | Sets maximum current used for wake and shake. |
| WS.MODE | R/W | Sets the method used for wake and shake. |
| WS.DISTMAX | R/W | Sets maximum movement allowed for wake and shake. |
| WS.NUMLOOPS | R/W | Sets the number of repetitions for wake and shake. |
| WS.STATE | R/O | Reads wake and shake status |
| WS.T | R/W | Sets wake and shake current-vector appliance time |
| WS.TDELAY1 | NV | Delay for wake and shake timing |
| WS.TDELAY2 | NV | Sets the delay for wake and shake timing. |
| WS.TDELAY3 | NV | Sets the delay for wake and shake between loops in mode 0. |
| WS.VTHRESH | NV | Defines the maximum allowed velocity for Wake & Shake |

This page intentionally left blank.

17 Faults and Warnings

| | | |
|-------------|---|------------|
| 17.1 | Fault and Warning Messages | 240 |
| 17.2 | Clearing Faults | 249 |
| 17.3 | Parameter and Command Error Messages | 250 |
| 17.4 | CANopen Emergency Messages and Error Codes | 253 |
| 17.5 | Unknown Fault | 256 |

17.1 Fault and Warning Messages

When a fault occurs, the drive fault relay is opened, the output stage is switched off (motor loses all torque), or the load is dynamically braked. The specific drive behavior depends on the type of fault. The LED display on the front panel of the drive shows the number of the fault that occurred. If a warning is issued prior to the fault, the warning is shown on the LED and has the same number as the associated fault. Warnings do not trip the power stage of the drive or fault relay output.

The left side of the LED displays F for a fault or n for a warning. The right side displays the fault or warning number as follows: 1-0-1-[break]. The highest priority fault is displayed on the LED. Multiple faults may be present when a fault condition is occurring. Check the AKDWorkBench Fault Screen or read the status of DRV.FAULTS through the controller or HMI for the entire list of faults.

NOTE

More information about fault messages and clearing faults can be found in the WorkBench online help. Procedures for clearing faults are described in the online help topic entitled "Faults and Warnings".

| Fault | Message/Warning | Cause | Remedy |
|--------------|---|---|--|
| F0 | | Reserved. | N/A |
| F101 | FPGA type mismatch. | Installed firmware is not compatible with the drive hardware. | Load compatible firmware into the drive. |
| n101 | The FPGA is a lab FPGA. | The FPGA is a lab version FPGA. | Load the released FPGA version that is compatible with the operational firmware. |
| F102 | Resident firmware failed. | Software failure detected. | Restart drive. If issue persists, contact technical support. |
| n102 | Operational FPGA is not a default FPGA. | The FPGA minor version is larger than the operational firmware default FPGA minor version | Load the released FPGA version that is compatible with the operational firmware. |
| F103 | Resident FPGA failed. | Software failure detected. Load resident FPGA failure occurred (several cases according to flowchart, including incompatible image to FPGA type and fieldbus type). | Restart drive. If issue persists, contact technical support. |
| F104 | Operational FPGA failed. | Software failure detected. Load operational FPGA failure occurred (several cases according to flowchart). | Restart drive. If issue persists, contact technical support. |
| F105 | NV memory stamp invalid. | NV memory stamp is corrupted or invalid. | Reset the drive to default memory values using Parameter Load in WorkBench. |
| F106 | NV memory data (might occur when downloading firmware). | NV memory data is corrupted or invalid. When this fault occurs after a firmware download, it is not an indication of a problem (clear the fault and perform a "save" to the drive). | Reset the drive to default memory values using Parameter Load in WorkBench. |
| F107 n107 | Positive switch limit exceeded. | Positive software position limit is exceeded. | Move the load away from the limits. |
| F108 n108 | Negative switch limit exceeded. | Negative software position limit is exceeded. | Move the load away from the limits. |
| F121 | Homing error. | Drive did not finish homing sequence. | Check homing sensor. |

| Fault | Message/Warning | Cause | Remedy |
|--------------|--|--|---|
| F123 n123 | Invalid motion task. | Invalid motion task. | Check motion task settings and parameters to make sure that the values entered will produce a valid motion task. |
| F125 n125 | Synchronization lost. | The fieldbus lost synchronization. | Check fieldbus connection (X5 and X6 if you are using EtherCAT; X12 and X13 if you are using CANopen) or the settings of your EtherCAT or CANopen master. |
| F126 n126 | Too much movement. | Too much movement was created during a Bode plot. Motor is unstable and is not following drive instructions. | Check that the system is closed loop stable. Refer to the system tuning guide. |
| F127 | Incomplete emergency stop procedure. | Incomplete emergency stop procedure (problem with the emergency stop motion task). | Disconnect power from drive and check emergency stop procedure. |
| F128 | MPOLES/FPOLES not an integer. | Ratio of motor poles to feedback poles must be a whole number. | Change to a compatible feedback device. |
| F129 | Heartbeat lost. | Heartbeat lost. | Check CANopen cabling. Reduce bus load or increase the heartbeat update time. |
| F130 | Secondary feedback supply over current. | Problem in secondary feedback detected. | Check X9 connection. |
| F131 | Emulated encoder line break. | Problem in secondary feedback detected. | Check secondary feedback (X9 connection). |
| F132 | Emulated encoder Z break. | Problem in secondary feedback detected. | Check secondary feedback (X9 connection). |
| F133 | Fault number changed to to F138. See F138 for details. | | |
| F134 | Secondary feedback illegal state. | Problem in secondary feedback detected. | Check X9 connection. |
| F135 n135 | Homing is needed. | Attempt to issue motion task before the axis is homed. Axis must be homed before motion task can start. | Change opmode or home axis. |
| F136 | FPGA version mismatch. Firmware and FPGA versions are not compatible | The FPGA version does not match the firmware FPGA version constants. | Load the FPGA version that is compatible with the firmware. |
| n137 | Homing and feedback mismatch | The configured homing mode is not supported by the motor feedback type used. | Change homing mode. |
| F138 | Instability during autotune | Drive current (IL.CMD) or velocity feedback (VL.FB) exceeds allowable limit (BODE.IFLIMIT or BODE.VFLIMIT). This fault only occurs in BODE.MODE 5. This fault often occurs when complex mechanics, belts, and compliant loads are present. | Change BODE.MODE if appropriate. If BODE.MODE 5 is appropriate and the fault occurs at the end of an Autotuning, then the motor is not robustly stable. You can manually adjust Autotuner settings. Manual tuning may be required to make the motor stable. |

| Fault | Message/Warning | Cause | Remedy |
|-------|--|---|--|
| n151 | Not enough distance to move; motion exception. | <p>For trapezoidal and customer table motion tasks: The target velocity specified in the motion task cannot be reached via using the selected acceleration and deceleration since the distance to travel is not sufficient.</p> <p>For a 1:1 profile: The selected acceleration and deceleration will be extended since there is too much distance to travel and the motion task would exceed its maximum allowed velocity.</p> | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. Check motion task settings and parameters to make sure that the values entered will produce a valid motion task. |
| n152 | Not enough distance to move; following motion exception. | A new motion task activated, when one motion task is already active and the target position specified in the motion task parameters cannot be reached with specified target velocity, acceleration and deceleration parameters. The motion task will directly decelerate to into the target position or ramps down to velocity 0 and start another move to reach target position of the next motion task. | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. Check motion task settings and parameters to make sure that the values entered will produce a valid motion task. |
| n153 | Velocity limit violation, exceeding max limit. | A new target velocity calculated internally due to an exception, and is being limited due to user velocity limit. | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. Check motion task target velocity settings and parameters to make sure that the values entered will not exceed the VL.LIMITP and VL.LIMITN setting. |
| n154 | Following motion failed; check motion parameters. | Activation of the following motion task failed due to incompatible parameters, or motion task does not exist. | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. Check following motion task settings and parameters to make sure that the values entered will produce a valid motion task. |
| n156 | Target position crossed due to stop command. | The motion task crosses the target position after triggering a DRV.STOP command. This situation can happen when processing a change-on-the-fly motion task and triggering a DRV.STOP command close to the target position of the currently running motion task. | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |
| n157 | Homing index pulse not found. | A homing mode with index detection is activated, and index pulse is not detected while moving across the range determined by the hardware limit switches. | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |

| Fault | Message/Warning | Cause | Remedy |
|-------|--|---|---|
| n158 | Homing reference switch not found. | A homing mode with reference switch detection is activated and the reference switch is not detected while moving across the range determined by the hardware limit switches. | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |
| n159 | Failed to set motion task parameters | Invalid motion task parameters assignment. This warning can appear upon an MT.SET command. | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. Check motion task settings and parameters. |
| n160 | Motion task activation failed. | Activation of the motion task failed due to incompatible parameters, or motion task does not exist. This warning can appear upon an MT.MOVE command. | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. Check motion task settings and parameters to make sure that the values entered will produce a valid motion task. |
| n161 | Homing procedure failed. | Homing error observed during the operation of homing procedure. | Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |
| n163 | MT.NUM exceeds limit. | This warning appears with n160. This warning is triggered when you try to trigger a motion task > 128 (such as MT.MOVE 130). | Trigger only motion tasks between 0 and 128. Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |
| n164 | Motion task is not initialized. | This warning appears with n160. This warning is triggered when you try to trigger a non-initialized motion task. | Initialize the motion task first before starting the task. Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |
| n165 | Motion task target position is out. | This warning appears with n160. This warning is triggered when you try to trigger a motion task with an absolute target position outside of the selected modulo range (see also MT.CNTL). | Move the absolute target position of the motion task within the modulo range. Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |
| n168 | Invalid bit combination in the motion task control word. | This warning appears with n160. This warning is triggered when you try to trigger a motion task with an invalid bit combination in the motion task control word (see also MT.CNTL). | Correct the MT.CNTL setting for the specific motion task. Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |
| n169 | 1:1 profile cannot be triggered on the fly. | This warning appears with n160. This warning is triggered when you try to trigger a 1:1 profile table motion task while another motion task is currently running. | 1:1 profile table motion tasks should be started from velocity 0. Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |

| Fault | Message/Warning | Cause | Remedy |
|------------------------|--|--|---|
| n170 | Customer profile table is not initialized. | This warning appears with n160. This warning is triggered when you try to trigger a motion task that uses a customer profile table for generating the velocity profile and when the selected profile table is empty (see MT.CNTL and MT.TNUM). | Change the MT.TNUM parameter for this specific motion task in order to use an initialized profile table. Activation of any new motion or using of DRV.CLRFAULTS will clear the warning. |
| F201 | Internal RAM failed. | Hardware failure detected. | Restart drive. If issue persists, contact technical support. |
| F202 | External RAM failed. | Hardware failure detected. | Restart drive. If issue persists, contact technical support. |
| F203 | Code integrity failed. | Software failure detected. FPGA register access failure occurred. | Restart drive. If issue persists, contact technical support. |
| F204 to F232 | EEPROM failure detected | EEPROM failure detected | Restart drive. If issue persists, exchange drive. |
| F234-F237 n234-n237 | Temperature sensor high. | High temperature limit reached. | Check cabinet ventilation system. |
| F240-F243 n240-n243 | Temperature sensor low. | Low temperature limit reached. | Check cabinet ventilation system. |
| F245 | External fault. | This fault is user generated and is caused by user settings. | Users can configure a digital input to trigger this fault (DINx.MODE = 10). The fault occurs according to this input setting. Clear the input to clear the fault. |
| F247 | Bus voltage exceed allowed thresholds. | Hardware problem in bus measurement. | Troubleshoot and repair hardware problem. |
| F301 n301 | Motor overheated. | Motor overheated. | Check ambient temperature. Check motor mounting heat sink capability |
| F302 | Over speed. | Motor exceeded VL.THRESH value. | Look for overshoot or lower requested speed. |
| F303 | Runaway. | Motor did not follow command values. | Gains are too low; motor is being over driven. |
| F304 n304 | Motor foldback. | Maximum motor power has been exceeded; the power has been limited to protect the motor | Motion is requiring too much power. Change move profile to reduce load on motor. Check for load jamming or sticking. Check that current limits are set correctly. |
| F305 | Brake open circuit. | Motor brake open circuit. | Check cabling and general functionality. |
| F306 | Brake short circuit. | Motor brake short circuit. | Check cabling and general functionality. |

| Fault | Message/Warning | Cause | Remedy |
|-----------|-----------------------------------|---|---|
| F307 | Brake closed during enable state. | Motor brake closed unexpectedly. | Check cabling and general functionality. |
| F308 | Voltage exceeds motor rating. | Drive bus voltage exceeds the motor's defined voltage rating. | Make sure that the motor fits the driving rating. |
| F401 | Failed to set feedback type. | Feedback is not connected or wrong feedback type selected | Check primary feedback (X10 connection). |
| F402 | Analog signal amplitude fault. | Analog signal amplitude is too low. Analog fault (resolver signal amplitude or sin/cos amplitude) | Check primary feedback (X10 connection), resolver and sine/cos encoder only. |
| F403 | EnDat communication fault. | General communication problem with feedback. | Check primary feedback (X10 connection), EnDat only |
| F404 | Hall error. | Hall sensor returns invalid Hall state (111, 000); either all Hall sensors are on or off. Legal Hall states are 001, 011, 010, 110, 100, and 101. This fault can be caused by a broken connection in any one of the Hall signals. | Check the feedback wiring; check all feedback connectors to ensure all pins are positioned correctly. |
| F405 | BiSS watchdog fault. | Bad communication with the feedback device. | Check primary feedback (X10 connection), Biss only. |
| F406 | BiSS multicycle fault. | | |
| F407 | BiSS sensor fault. | | |
| F408-F416 | SFD feedback fault. | Bad communication with the SFD device. | Check primary feedback (X10 connection). If fault persists, internal feedback failure. Return to manufacturer for repair. |
| F417 | Broken wire in primary feedback. | In primary feedback, a broken wire was detected (incremental encoder signal amplitude). | Check feedback cable continuity. |
| F418 | Primary feedback power supply. | Power supply fault for primary feedback. | Check primary feedback (X10 connection). |
| F419 | Encoder init procedure failed | Phase find procedure did not complete successfully. | Check encoder wiring, reduce/balance motor load prior to phase finding. |
| F424 | Resolver amplitude low. | Resolver signal amplitude is below minimum level. | Check primary feedback (X10 connection). |
| F425 | Resolver amplitude high. | Resolver signal amplitude is above maximum level. | Check primary feedback (X10 connection). |
| F426 | Resolver error. | Resolver excitation fault. | Check primary feedback (X10 connection). |
| F427 | Analog low. | Analog signal amplitude low. | Check primary feedback (X10 connection). |
| F428 | Analog high. | Analog signal amplitude high. | Check primary feedback (X10 connection). |
| F429 | Incremental low. | Incremental encoder signal amplitude is below minimum level. | Check primary feedback (X10 connection). |
| F430 | Incremental high. | Incremental encoder signal amplitude is above maximum level. | Check primary feedback (X10 connection). |
| F431 | Secondary feedback Hall error. | Secondary feedback illegal Hall state (000,111). | Check X9 connection. |

| Fault | Message/Warning | Cause | Remedy |
|--------------|--|--|---|
| F432 | Communication fault. | General communication problem with secondary feedback. | Check secondary feedback (X10 connection). |
| F437 | Close to limit. | Drive or motor over current or over speed warning. | Check for increased load, jamming or sticking. Is position error set too low? |
| F438 n439 | Following error (numeric) | Motor did not follow command values. Motor exceeded maximum allowed position following error (numeric). | Check for increased load, jamming or sticking. Is position error set too low? |
| F439 n439 | Following error (user). | Motor did not follow command values. Motor exceeded maximum allowed position following error (user). | Check feedback commutation setup and tuning parameters. |
| F450 | Following error (presentation). | Motor did not follow command values. Motor exceeded maximum allowed position following error (presentation). | Check feedback commutation setup and tuning parameters. |
| F473 | Wake and Shake. Insufficient movement | There was less movement than defined by WS.DISTMIN. | Increase WS.IMAX and/or WS.T |
| F475 | Wake and Shake. Excess movement. | WS.DISTMAX has been exceeded. | Increase WS.DISTMAX value or reduce WS.IMAX or WS.T. |
| F476 | Wake and Shake. Fine-coarse delta too large. | The angle difference between the coarse and fine calculation was larger than 72 deg. | Modify WS.IMAX or WS.T and try again. |
| F478 n478 | Wake and Shake. Over-speed. | WS.VTHRESH was exceeded. | Increase WS.VTHRESH value or reduce WS.IMAX or WS.T. |
| F479 n479 | Wake and Shake. Loop angle delta too large. | The angle between complete loops was larger than 72 deg. | Modify WS.IMAX or WS.T and try again. |
| F480 | Fieldbus command velocity too high. | Fieldbus command velocity exceeds VL.LIMITP. | Lower fieldbus command trajectory, or increase the value of VL.LIMITP. |
| F481 | Fieldbus command velocity too low. | Fieldbus command velocity exceeds VL.LIMITN. | Increase fieldbus command trajectory, or decrease the value of VL.LIMITN. |
| F482 | Commutation not initialized. | The motor requires the commutation initialization (there are no encoder commutation tracks, Hall sensors, etc.) and no successful Wake and Shake sequence has been performed | Clear any faults, activate the Wake and Shake procedure (WS.ARM) and enable the drive. |
| F483 | Motor U phase missing. | No current was detected in the motor's U phase during initialization (Mode 0 only). | Check the motor connections and WS.IMAX (very low current may produce this error). |
| F484 | Motor V phase missing. | No current was detected in the motor's V phase during initialization (Mode 0 only). | Check the motor connections and WS.IMAX (very low current may produce this error). |
| F485 | Motor W phase missing. | No current was detected in the motor's W phase during initialization (Mode 0 only). | Check the motor connections and WS.IMAX (very low current may produce this error). |
| F501 n501 | Bus over voltage. | Bus voltage too high. Usually, this problem is load related. | Reduce load or change motion profile. Check system regen capacity; add capacity if needed. Check mains voltage. |

| Fault | Message/Warning | Cause | Remedy |
|---------------|--|---|---|
| F502 | Bus under voltage. Warning issued prior to fault. | Bus voltage below threshold value. | Check mains voltage. |
| F503 n503 | Bus capacitor overload. | Single phase AC input on a drive only rated for three-phase input or excessive single-phase power load. | Check mains voltage. |
| F504- F518 | Internal supply voltage fault | Internal supply voltage fault detected | Check wiring for electromagnetic compatibility (EMC). If issue persists exchange drive. |
| F519 | Regen short circuit. | Regen resistor short circuit. | Regen IGBT short circuit. Contact technical support. |
| F520 | Regen overload. | Regen resistor overload. | Motor is being overhauled or motor is being stopped too quickly. |
| F521 n521 | Regen over power. | Too much power stored in regen resistor. | Either get larger regen resistor or use DC bus sharing to dissipate power. |
| F523 | Bus over voltage FPGA | Bus over voltage hard fault. | Check mains voltage and check system brake capacity. |
| F524 n524 | Drive foldback. | Maximum drive power has been exceeded. The power has been limited to protect the drive. | Motion requires too much power. Change profile to reduce load . |
| F525 | Output over current. | Current exceeds drive peak. | Check for short or feedback faults. |
| F526 | Current sensor short circuit. | Current sensor short circuit. | Restart drive. If issue persists, contact technical support. |
| F527 | Iu current AD converter stuck. | Hardware failure detected. | Restart drive. If issue persists, contact technical support. |
| F528 | Iv current AD converter stuck. | Hardware failure detected. | Restart drive. If issue persists, contact technical support. |
| F529 | Iu current offset limit exceeded. | Hardware failure detected. | Restart drive. If issue persists, contact technical support. |
| F530 | Iv current offset limit exceeded. | Hardware failure detected. | Restart drive. If issue persists, contact technical support. |
| F531 | Power stage fault. | Hardware failure detected. | Restart drive. If issue persists, replace drive. |

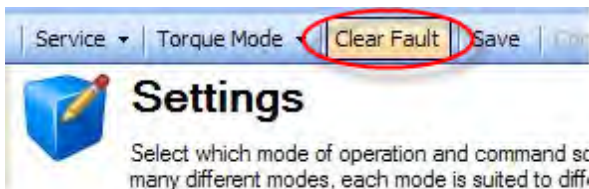
| Fault | Message/Warning | Cause | Remedy |
|--------------|---|---|---|
| F532 | Drive motor parameters setup incomplete. | Before a motor can be enabled, you must configure a minimum set of parameters. These parameters have not been configured. | Issue the command <code>DRV.S-ETUPREQLIST</code> to display the list of the parameters that you must configure. Configure these parameters either manually or automatically. You can manually configure these parameters in three ways: (1) set each parameter individually; (2) use the setup wizard to select the motor; or (3) select the motor type from the motor data base in the Motor window (<code>MOTOR.AUTASET</code> must be set to 0 (FALSE)). If you use the Motor window, you must first select the feedback type. If the motor has Biss Analog, Endat, or SFD feedback (feedback with memory), then these parameters are set automatically when <code>MOTOR.AUTASET</code> is set to 1 (TRUE). |
| F534 | Failed to read motor parameters from feedback device. | Motor either does not have motor feedback memory, or the motor feedback memory is not programmed properly so the parameters cannot be read. | Try to read parameters again by clicking the Disable and Clear Faults button, or by issuing the <code>DRV.CLRFAULTS</code> command. If this attempt is not successful, then set <code>MOTOR.AUTASET</code> to 0 (false) and program the parameters using the setup wizard or manually set up the parameters. If the motor has motor memory (Biss Analog, Endat, and SFD motors have motor memory), return the motor to have the memory programmed. |
| F602 | Safe torque off. | Safe torque off function has been triggered. | Reapply supply voltage to STO if safe to do so. |
| F701 | Fieldbus runtime. | Runtime communication fault. | Check fieldbus connections (X11), settings, and control unit. |
| F702 n702 | Fieldbus communication lost. | All fieldbus communication was lost. | Check fieldbus connections (X11), settings, and control unit. |
| F703 | Emergency timeout occurred while axis should disable | Motor did not stop in the timeout defined. | Change timeout value, change stop parameters, improve tuning. |

17.2 Clearing Faults

You can clear drive faults in four ways: 1. using the WorkBench toolbar; 2. using the WorkBench terminal screen, 3. using the CANopen interface; or 4. using an external digital signal. In the first three methods, clearing the fault automatically disables the drive (by issuing the DRV.DIS command). After the fault is cleared, you must then enable the drive again (by issuing the DRV.EN command). In the fourth method, clearing the fault does not automatically disable the drive. Once the fault is cleared, the drive will immediately become enabled again when no fault conditions exist.

The external digital signal method is useful when the application does not require a field bus, therefore the controller cannot issue fieldbus commands (for example, DRV.EN). The fault clearing function is simpler in this case. The controller can clear a fault condition with a single output and no fieldbus command is required. The hardware enable input is still utilized for safety. Many machines are set to disable when a fault occurs, therefore, the enable circuit is connected only after the fault clear circuit is toggled.

1. Using the WorkBench toolbar. Click the **Clear Fault** button to clear the fault, then click the **Enable** button to enable the drive again.



2. Using an ASCII command. In WorkBench terminal screen, enter DRV.CLRFAULTS, then enter DRV.EN to enable the drive.
3. Using the CANopen interface (Object 6040h: Controlword DS402). The ASCII control commands are built up from the logical combination of the bits in the control word and external signals (enable output stage). You can configure the **Reset Fault** bits in the control word to clear the fault. See Object 6040h: Control word (DS402) in the CANopen manual for details on clearing faults using this method. After the fault is cleared, you must enable the drive.
4. Using an external digital signal. First, provide a disable command to the drive, since this command is not automatically issued when the fault is cleared using an external digital signal. You must provide this disable command after the drive has disabled itself and before the clear fault command is issued in order for the clear fault command to be accepted.

Digital input mode 1 clears a fault condition. In the WorkBench terminal screen, enter DINx.MODE 1, with x set to the corresponding digital input number. See the Installation Manual (10.3 Digital Inputs (X7/X8) for digital input numbers and corresponding pin connections.

Example:

DIN2.MODE 1 sets digital input 2 (pin 9 on connector X7) to perform the clear fault function. Any free digital input shown in Table x can be set to mode 1 (Reset faults).

17.3 Parameter and Command Error Messages

| Error Message | Meaning | Remedy | Occurrence |
|---|--|--|---|
| Error: Parameter or command was not found. | Parameter or command string was not recognized as a known command. | Check the entered string for accuracy. Consult the reference guide or enter DRV.LIST in the WorkBench terminal screen to find valid parameters and commands. | General. |
| Error: Not a trigger parameter. | Parameter cannot be used for triggering the scope. | Use a different parameter as a trigger value. | Recorder parameters. |
| Error: Parameter not recordable. | Parameter cannot be scoped. | Do not scope this parameter. | Recorder parameters. |
| Error: Wrong argument for parameter or command. | Argument was not accepted. | Check the entered argument for accuracy. Consult the reference guide or enter DRV.LIST in the WorkBench terminal screen to find valid parameter and command arguments. | General. |
| Error: No arguments allowed for parameter or command. | Parameter or command supports no arguments. | Enter parameter or command again with no arguments. | Commands and read-only parameters |
| Error: Too many arguments for parameter or command. | Too many arguments provided for this parameter or command. | Enter again with fewer arguments. | All parameters and commands with no or one argument. |
| Error: No float allowed. | Float number was entered, but only integer numbers are allowed. | Enter integer value instead. | All parameters and commands with integer arguments. |
| Error: Parameter or command is read-only. | Argument was not accepted because the parameter or command is read-only. | Enter again with no arguments. | Commands and all read-only parameters. |
| Error: Parameter is temporarily write protected. | Argument was not accepted because the parameter is currently read-only. | Check reference guide or WorkBench help to determine if the parameter is read-only, read-write, or nonvolatile. | Some parameters that come from the Smart Feedback Device (SFD). |
| Error: Argument greater than maximum value. | Entered argument was beyond maximum value. | Enter a value that is within the allowable range. See reference guide or WorkBench help for ranges. | Commands and read-write parameters. |
| Error: Too many characters in parameter or command. | Parameter or command string was too long. | Shorten string and enter again. | General. |
| Error: Argument less than minimum value. | Entered argument was too small. | Enter value that is within the limits. | Commands and read-write parameters. |
| Error: Argument is out of data range. | Entered argument was not within the data range limits. | Enter value that is within the limits. | Commands and read-write parameters. |
| Error: Parameter or command is password protected. | Use of this parameter or command requires a password. | Contact technical support to obtain password. Enter valid password. | Password protected parameters and commands. |

| Error Message | Meaning | Remedy | Occurrence |
|---|--|--|---|
| Error: Wrong operation mode for parameter or command. | Requested parameter or command cannot be performed in this operation mode. | Change to valid operation mode. | |
| Error: Drive enabled. Disable drive and continue. | Action can only be performed if drive is disabled. | Disable drive and repeat action. | |
| Error: Drive disabled. Enable drive and continue. | Action can only be performed if drive is enabled. | Enable drive and repeat action. | |
| Error: Data value is invalid. | Entered data was not valid. | Provide valid data. | |
| Error: Argument must be an even number. | Argument is not an even number. | Enter a value divisible by two without remainder. | |
| Error: Argument must be an odd number. | Argument must be an odd number. | Enter a value not divisible by two with remainder. | |
| Error: Axis is not homed. | Axis is not homed. | Home axis and repeat action. | |
| Error: All recorder channels empty. | No data is specified to record. | Specify the data to be recorded and start again. | Recorder commands. |
| Error: Process already active. | Required action is already running. | Wait until action is finished and start again if needed. | |
| Error: Motor is currently in motion. | Action can only be performed if motor is not moving. | Stop motion and repeat action. | Motion tasks, service motion, and others. |
| Error: EEPROM is bad or does not exist. | EEPROM is damaged. | Restart drive. If fault persists, replace the drive and consult technical support. | |
| Error: Unknown board. | | | |
| Error: Fault exists. Correct fault condition and then clear faults. | A fault is present in the system. | Clear the fault, disable the drive, and then enable the drive again. | |
| Error: Cannot clear faults. Issue software or hardware disable first. | | Clear hardware or software enable and repeat action. | |
| Error: EEPROM is busy. | | | |
| Error: Invalid motor/feedback poles ratio. | | | |
| Error: Unknown Fault. | Situation should not happen. | Clear fault. If fault recurs, consult technical support. | Unknown. |
| Error: Invalid motor/feedback poles ratio. | | | |
| Error: Invalid motion task parameter. | | | |
| Error: Invalid motion task number. | | | |

| Error Message | Meaning | Remedy | Occurrence |
|--|---------|--------|------------|
| Error: Invalid motion task velocity. | | | |
| Error: Invalid motion task acceleration or deceleration. | | | |
| Error: Invalid motion task customer profile table number. | | | |
| Error: Invalid motion task following number. | | | |
| Error: Function not available for the active command source. | | | |
| Error: Invalid Bode plot mode for this function. | | | |
| Error: Invalid sine-sweep Bode plot mode. | | | |
| Error: Bode plot start frequency greater than or equal to end frequency. | | | |
| Error: Function not available while a controlled stop is in process. | | | |
| Error: Not enough memory available. | | | |
| Error: Function not available while a Bode-plot measurement is active. | | | |
| Error: Iu/Iv offset calculation not finished. | | | |
| Error: Buffer overflow. | | | |
| Error: Cannot save to EEPROM while inrush relay is closed. | | | |
| Error: Test mode is off. | | | |
| Error: Cannot change digital input mode. Issue software or hardware disable first. | | | |
| Error: Internal drive procedure active: controlled stop, burn-in, phase find, or zero. | | | |
| Error: General motion fault. | | | |

17.4 CANopen Emergency Messages and Error Codes

Emergency messages are triggered by internal equipment errors. They have a high ID-priority to ensure quick access to the bus. An emergency message contains an error field with pre-defined error/fault numbers (2 bytes), an error register (1byte), the error category (1 byte), and additional information.

Error numbers from 0000h to 7FFFh are defined in the communication or drive profile. Error numbers from FF00h to FFFFh have manufacturer-specific definitions. The following table describes the various error codes:

| Error Code | Fault/Warning Code | Description |
|------------|--------------------|---|
| 0x0000 | 0 | Emergency error free |
| 0x3110 | 523 | Bus over voltage |
| 0x3120 | - | Bus under voltage |
| 0x3130 | 503 | Phase failure |
| 0x3180 | 503 | Warning: Supply missing phase |
| 0x3210 | 501 | DC link over-voltage |
| 0x3220 | 502 | DC link under-voltage |
| 0x4210 | 234 | Excess temperature, device (control board) |
| 0x4310 | 235 | Excess temperature, drive (heat sink) |
| 0x5113 | 512 | Supply low voltage, U3 = supply +5 V |
| 0x5114 | 505 | Supply low voltage, manufacturer specific: supply 1V2 |
| 0x5115 | 507 | Supply low voltage, manufacturer specific: supply 2V5 |
| 0x5116 | 509 | Supply low voltage, manufacturer specific: supply 3V3 |
| 0x5117 | 514 | Supply low voltage, manufacturer specific: supply 12 V |
| 0x5118 | 516 | Supply low voltage, manufacturer specific: supply -12V |
| 0x5119 | 518 | Supply low voltage, manufacturer specific:supply 3V3 analogue |
| 0x5180 | 504 | Supply overvoltage, manufacturer specific:supply 1V2 |
| 0x5181 | 506 | Supply overvoltage, manufacturer specific:supply 2V5 |
| 0x5182 | 508 | Supply overvoltage, manufacturer specific:supply 3V3 |
| 0x5183 | 510 | Supply overvoltage, manufacturer specific:supply 5V0 |
| 0x5184 | 513 | Supply overvoltage, manufacturer specific:supply 12 V |
| 0x5185 | 104 | Supply overvoltage, manufacturer specific:supply -12V |
| 0x5186 | 517 | Supply overvoltage, manufacturer specific:supply 3V3 analogue |
| 0x5510 | 201 | Hardware memory, RAM |
| 0x5530 | 105 | Hardware memory, EEPROM |
| 0x5580 | 106 | Hardware memory, non-volatile memory data |
| 0x5581 | 202 | Hardware memory, external Ram for resident firmware failed |
| 0x5582 | 203 | Hardware memory, code integrity failed for resident firmware |
| 0x5583 | 102 | Hardware memory, resident firmware failed |
| 0x5584 | 103 | Hardware memory, resident FPGA failed |
| 0x5585 | 104 | Hardware memory, operational FPGA failed |
| 0x7180 | 301 | Motor overheat |
| 0x7182 | 305 | Motor open circuit |
| 0x7183 | 306 | Motor short circuit |
| 0x7184 | 307 | Motor brake applied |
| 0x7303 | 426 | Resolver 1 fault |
| 0x7305 | 417 | Incremental sensor 1 fault |
| 0x7380 | 402 | Feedback 1, analogue fault |
| 0x7381 | 403 | Feedback 1, EnDat communication fault |
| 0x7382 | 404 | Feedback 1, illegal hall |
| 0x7383 | 405 | Feedback 1, BiSS watchdog |

| Error Code | Fault/Warning Code | Description |
|------------|--------------------|---|
| 0x7384 | 406 | Feedback 1, BiSS multi cycle |
| 0x7385 | 407 | Feedback 1, BiSS sensor |
| 0x7386 | 408 | Feedback 1. SFD configuration |
| 0x7387 | 409 | Feedback 1, SFD UART overrun |
| 0x7388 | 410 | Feedback 1, SFD UART frame |
| 0x7389 | 412 | Feedback 1, SFD UART parity |
| 0x738A | 413 | Feedback 1. SFD transfer timeout |
| 0x738C | 415 | Feedback 1, SFD mult. corrupt position |
| 0x738D | 416 | Feedback 1, SFD Transfer incomplete |
| 0x738E | 418 | Feedback 1, power supply fault |
| 0x738F | 401 | Feedback 1, failed to set feedback |
| 0x73A0 | 424 | Feedback 2, resolver amplitude low |
| 0x73A1 | 425 | Feedback 2, resolver amplitude high |
| 0x73A2 | 425 | Feedback 2, resolver fault |
| 0x73A3 | 427 | Feedback 2, analogue low |
| 0x73A4 | 428 | Feedback 2, analogue high |
| 0x73A5 | 429 | Feedback 2, incremental low |
| 0x73A6 | 430 | Feedback 2, incremental high |
| 0x73A7 | 431 | Feedback 2, halls |
| 0x73A8 | 432 | Feedback 2, communication |
| 0x73A9 | - | Reserved |
| 0x73AA | - | Reserved |
| 0x8130 | 129 | Life Guard Error or Heartbeat Error |
| 0x8311 | 304 | Excess torque |
| 0x8331 | 524 | Torque fault |
| 0x8480 | 302 | Velocity overspeed |
| 0x8580 | 107 | Software limit switch, positive |
| 0x8581 | 108 | Software limit switch, negative |
| 0x8611 | 439 | Following error |
| 0x8686 | 151 | Not enough distance to move; Motion Exception |
| 0x8687 | 152 | Not enough distance to move; Following Motion Exception |
| 0x8688 | 153 | Velocity Limit Violation, Exceeding Max Limit |
| 0x8689 | 154 | Following Motion Failed; Check Motion Parameters |
| 0x868A | 156 | Target Position crossed due to Stop command |
| 0x86A0 | 157 | Homing Index pulse not found |
| 0x86A1 | 158 | Homing Reference Switch not found |
| 0x86A2 | 159 | Failed to set motion task parameters |
| 0x86A3 | 160 | Motion Task Activation Failed |
| 0x86A4 | 161 | Homing Procedure Failed |
| 0x8780 | 125 | Fieldbus synchronization lost |
| 0x8AF0 | 137 | Homing and Feedback mismatch |
| 0xFF00 | 701 | Fieldbus runtime fault |
| 0xFF01 | 702 | Fieldbus communication lost |
| 0xFF02 | 529 | Iu offset limit exceeded |
| 0xFF03 | 530 | Iv offset limit exceeded |
| 0xFF04 | 521 | Stored energy reached critical point |
| 0xFF05 | 527 | Iu detection stuck |
| 0xFF06 | 528 | Iv detection stuck |

| Error Code | Fault/Warning Code | Description |
|------------|--------------------|--|
| 0xFF07 | 525 | Control output over current |
| 0xFF08 | 526 | Current sensor short circuit |
| 0xFF09 | 128 | Axis dpoles |
| 0xFF0A | 531 | Power stage fault |
| 0xFF0B | 602 | Safe torque off |
| 0xFF10 | 414 | Warning: SFD single corrupted position. |
| 0x8380 | 524 | Warning: Drive foldback |
| 0x8387 | 304 | Warning: Motor foldback |
| 0xFF13 | 438 | Warning: Control position deviation |
| 0x8582 | 107 | Warning: Positive software position limit is exceeded. |
| 0x8583 | 108 | Warning: Negative software position limit is exceeded. |
| 0x8684 | 123 | Warning: Motion global warning |
| 0x7786 | 301 | Warning: Motor overheated. |
| 0x4380 | 234 | Warning: Control temperature sensor 1 high |
| 0x4387 | 235 | Warning: Power temperature sensor 1 high. |
| 0x4382 | 236 | Warning: Power temperature sensor 2 high. |
| 0x4393 | 237 | Warning: Power temperature sensor 3 high. |
| 0x4394 | 240 | Warning: Control temperature sensor 1 low. |
| 0x4385 | 241 | Warning: Control temperature sensor 1 low. |
| 0x4396 | 242 | Warning: Power temperature sensor 2 low. |
| 0x4392 | 243 | Warning: Power temperature sensor 3 low. |
| 0x3280 | 502 | Warning: Bus under voltage. |
| 0x3780 | 503 | Warning: Input phase loss. |
| 0x3287 | 521 | Warning: Dynamic Braking I ² T. |
| 0x8787 | 125 | Warning: Fieldbus PLL unlocked. |

17.5 Unknown Fault

This fault message occurs when an undefined fault condition is encountered.

17.5.0.1 Remedies

1. Click **Clear Fault**.

18 Troubleshooting the AKD

Drive problems occur for a variety of reasons, depending on the conditions in your installation. The causes of faults in multi-axis systems can be especially complex. If you cannot resolve a fault or other issue using the troubleshooting guidance presented below, customer support can give you further assistance.

| Problem | Possible Causes | Remedy |
|--|---|---|
| HMI message: Communication fault | <ul style="list-style-type: none"> — wrong cable used — cable plugged into wrong position on drive or PC — wrong PC interface selected | <ul style="list-style-type: none"> — plug cable into the correct sockets on the drive and PC — select correct interface |
| Motor does not rotate | <ul style="list-style-type: none"> — drive not enabled — software enable not set — break in setpoint cable — motor phases swapped — brake not released — drive is mechanically blocked — motor pole no. set incorrectly — feedback set up incorrectly | <ul style="list-style-type: none"> — apply ENABLE signal — set software enable — check setpoint cable — correct motor phase sequence — check brake control — check mechanism — set motor pole no. — set up feedback correctly |
| Motor oscillates | <ul style="list-style-type: none"> — gain is too high (speed controller) — feedback cable shielding broken — AGND not wired up | <ul style="list-style-type: none"> — reduce VL.KP (speed controller) — replace feedback cable — join AGND to CNC-GND |
| Drive reports following error | <ul style="list-style-type: none"> — Irms or Ipeak set too low — current or velocity limits apply — accel/decel ramp is too long | <ul style="list-style-type: none"> — verify motor/drive sizing — verify that IL.LIMITN, IL.LIMITP, VL.LIMITN, or VL.LIMITP is not limiting the drive — reduce DRV.ACC/DRV.DEC |
| Motor overheating | <ul style="list-style-type: none"> — motor operating above its rating | <ul style="list-style-type: none"> — verify motor/drive sizing — verify motor continuous and peak current values are set correctly |
| Drive too soft | <ul style="list-style-type: none"> — Kp (speed controller) too low — Ki (speed controller) too low — filters set too high | <ul style="list-style-type: none"> — increase VL.KP (speed controller) — increase VL.KI (speed controller) — refer to documentation regarding reducing filtering (VL.AR*) |
| Drive runs roughly | <ul style="list-style-type: none"> — Kp (speed controller) too high — Ki (speed controller) too high — filters set too low | <ul style="list-style-type: none"> — reduce VL.KP (speed controller) — reduce VL.KI (speed controller) — refer to documentation regarding increasing filtering (VL.AR*) |
| During the installation, a dialog box saying “Please wait while the installer finishes determining your disk space requirements” appears and never disappears. | <ul style="list-style-type: none"> — MSI installer issue. | <ul style="list-style-type: none"> — Cancel the installation. — Make sure that you have enough disk space on your hard disk (~500MB to allow Windows .NET update if necessary), if not make some space. — Relaunch the installer (you may need to try several times, the problem is random). |

19 Firmware and Firmware Updates

| | | |
|------|--|-----|
| 19.1 | Downloading Firmware..... | 259 |
| 19.2 | Firmware Compatibility..... | 259 |
| 19.3 | Invalid Firmware..... | 259 |
| 19.4 | Forcing the drive into firmware download mode..... | 260 |
| 19.5 | Forced download of AKD firmware..... | 260 |

19.1 Downloading Firmware

To download firmware, click **More** on the **AKD Overview** screen (see 6.2 AKD Overview). The information displayed includes the current firmware version. Click **Download** to display the **Download Firmware** view and update the firmware.

When you download the firmware, you may see a **Save** dialog box; see 13.4 Save On Firmware Download for more information about saving your drive configuration prior to a firmware download.

During the download, you cannot perform any actions with the drive.

CAUTION

While WorkBench is downloading firmware to your drive, do not remove the 24 V logic power. If you remove the 24 V logic power during a firmware download, a severe drive crash can occur. If a crash occurs, the drive will restart in a special mode and WorkBench will prompt you to reload the firmware.

During the firmware download, the Download Firmware view displays a progress bar and the following messages as the download proceeds:

1. **Reading the firmware file.** This step duration depends on where the file is physically stored.
2. **Resetting the drive.** This step takes about 10 seconds.
3. **Erasing the old firmware.** This step takes about 20 seconds.
4. **Downloading the new firmware to the drive.** The drive downloads the new firmware and then programs the new firmware into the nonvolatile memory of the drive. This step takes about 20 seconds.
5. **Resetting the drive.** This step takes about 10 seconds.

During the download process, the drive LED displays [dL]. Additional codes may appear during the download; see 3.2 Display Codes for a description of display codes related to a firmware download.

19.2 Firmware Compatibility

WorkBench is compatible with all previous drive firmware versions. Any new WorkBench versions issued are compatible with older firmware versions. WorkBench forward compatibility, however, cannot be ensured. When a behavior in the firmware changes, an older WorkBench version may not function correctly with the new firmware. Kollmorgen recommends that you update WorkBench when you install new firmware.

The table below describes compatibility between WorkBench and firmware versions:

| | Firmware version M_01-01-00-zzz | Firmware version M_01-02-00-zzz | Firmware version M_01-03-00-zzz |
|---------------------------|------------------------------------|------------------------------------|------------------------------------|
| WorkBench Version 1.1.x.x | √ | x | x |
| WorkBench Version 1.2.x.x | √ | √ | x |
| WorkBench Version 1.3.x.x | √ | √ | √ |

Any new firmware not listed here may have compatibility issues with WorkBench. Please check for latest version of WorkBench on the Kollmorgen website:

http://www.kollmorgen.com/akd_software

19.3 Invalid Firmware

When WorkBench connects to a drive, it checks to make sure that the drive is executing a compatible version of the drive firmware. If the drive is executing invalid firmware, then WorkBench cannot operate correctly and shows this window.

| Button | Description |
|----------|---|
| Download | This command allows you to select a different version of the firmware and download it to the drive. |

| Button | Description |
|------------|--|
| Disconnect | This command stops all communications with the drive and return to the disconnected state. |

19.4 Forcing the drive into firmware download mode

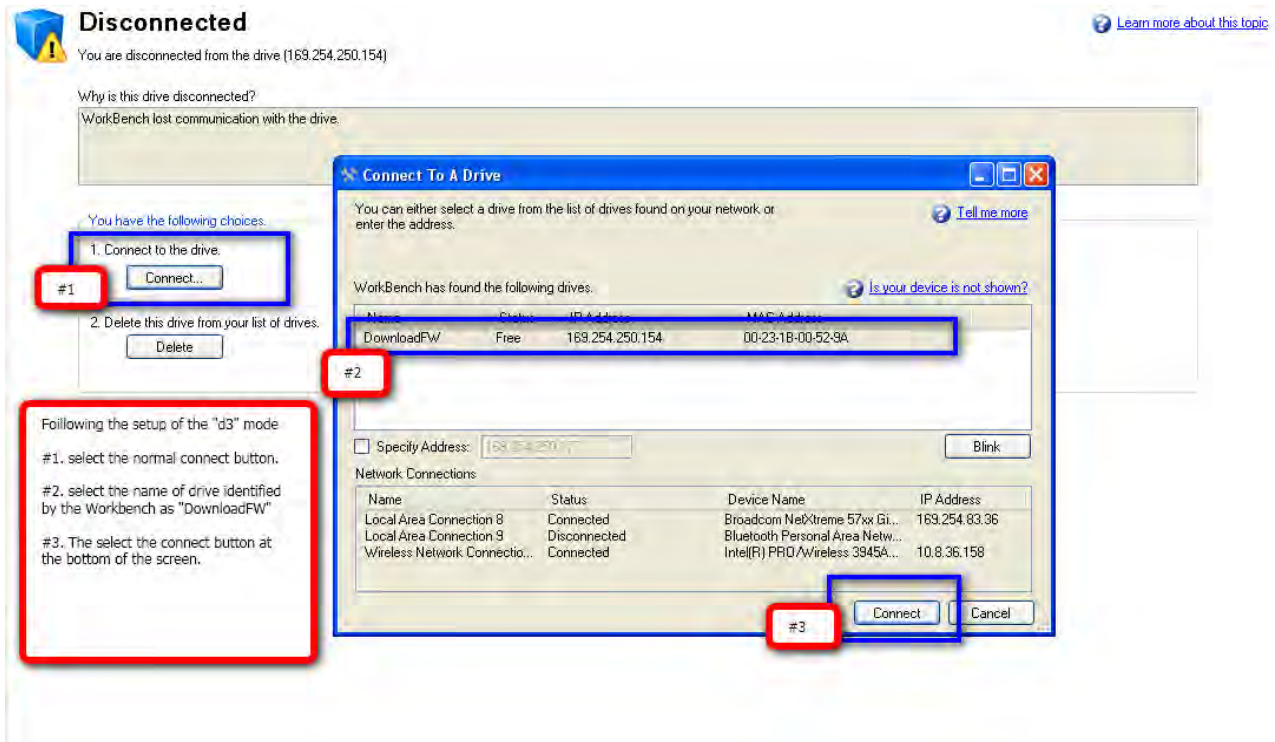
19.5 Forced download of AKD firmware.

In most cases, you can download firmware using the WorkBench without setting the hardware into a specific download state. If you are unable to connect to WorkBench for some reason, the following “forced firmware download” procedure is needed.

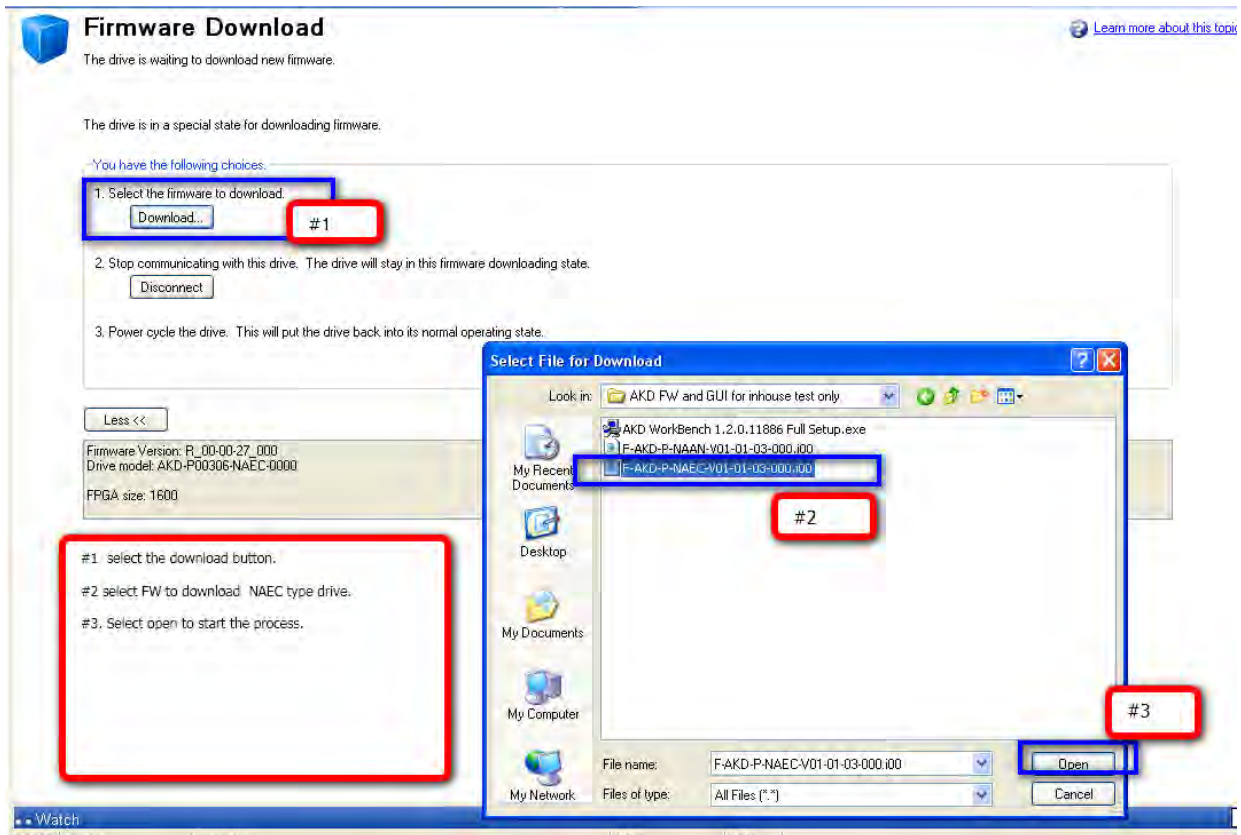
Power the AKD unit down then:

1. Depress the B1 button (display of IP Address Button) while powering up the drive. Continue to depress the button until the drive display shows "d3" (5-10 sec).
2. Once the display on the drive shows "d3" , have the EtherNet cable connected to the drive Point to Point.
3. The drive can now be connected to WorkBench in download mode.
4. Select the firmware to be downloaded; click **Open** to start the process.

WorkBench will display the **Disconnected** view. See the screenshots below for the download connection sequence.



The **Firmware Download** view appears next. See the screenshot below for the firmware selection and download sequence.



3. Next, the **Downloading Firmware** view appears and the progress bar displays the firmware download progress.

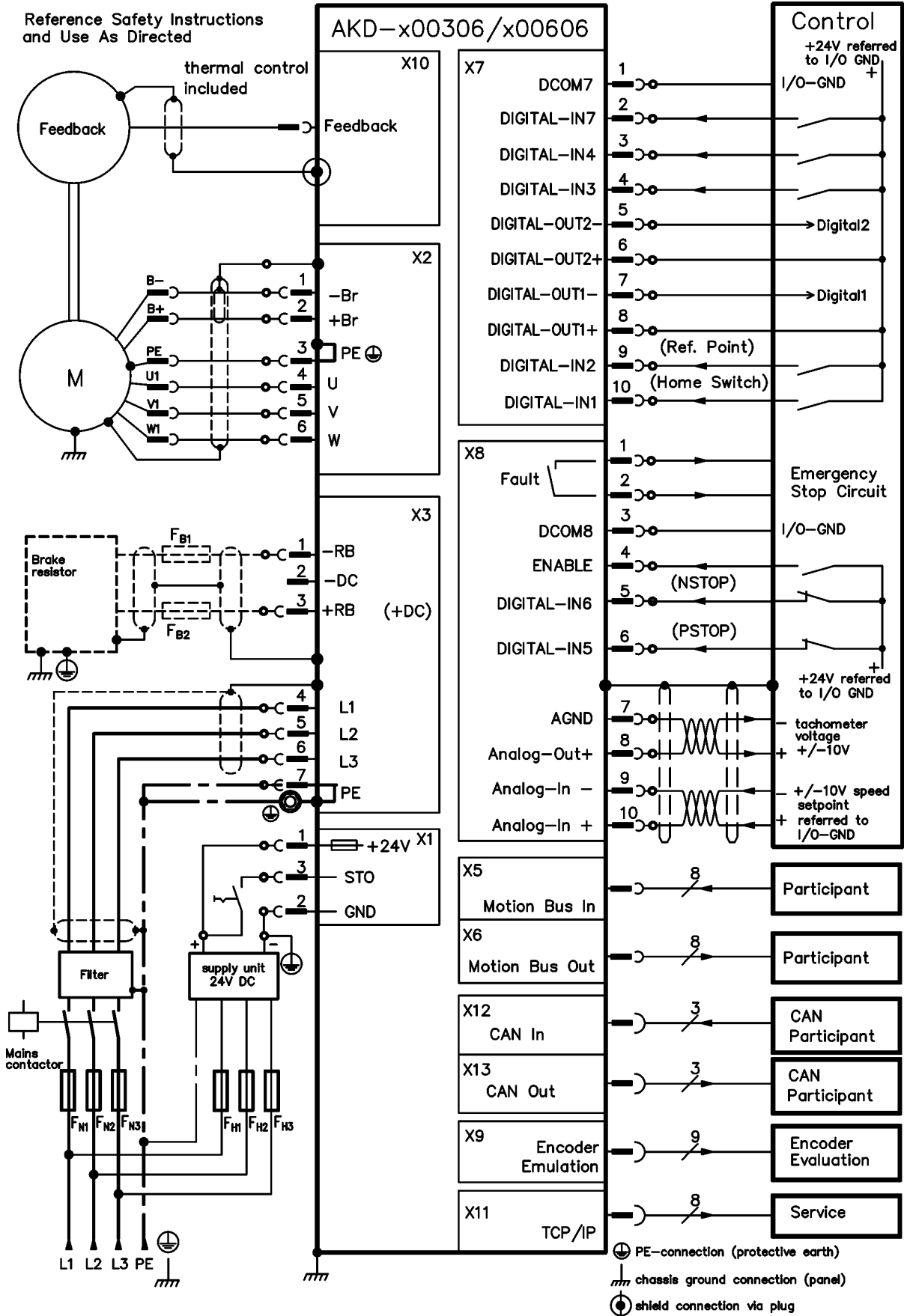
4. Once the download is complete, open the **AKD Overview** screen. The **Firmware Version** box displays the new firmware version, which confirms your successful download.

This page intentionally left blank.

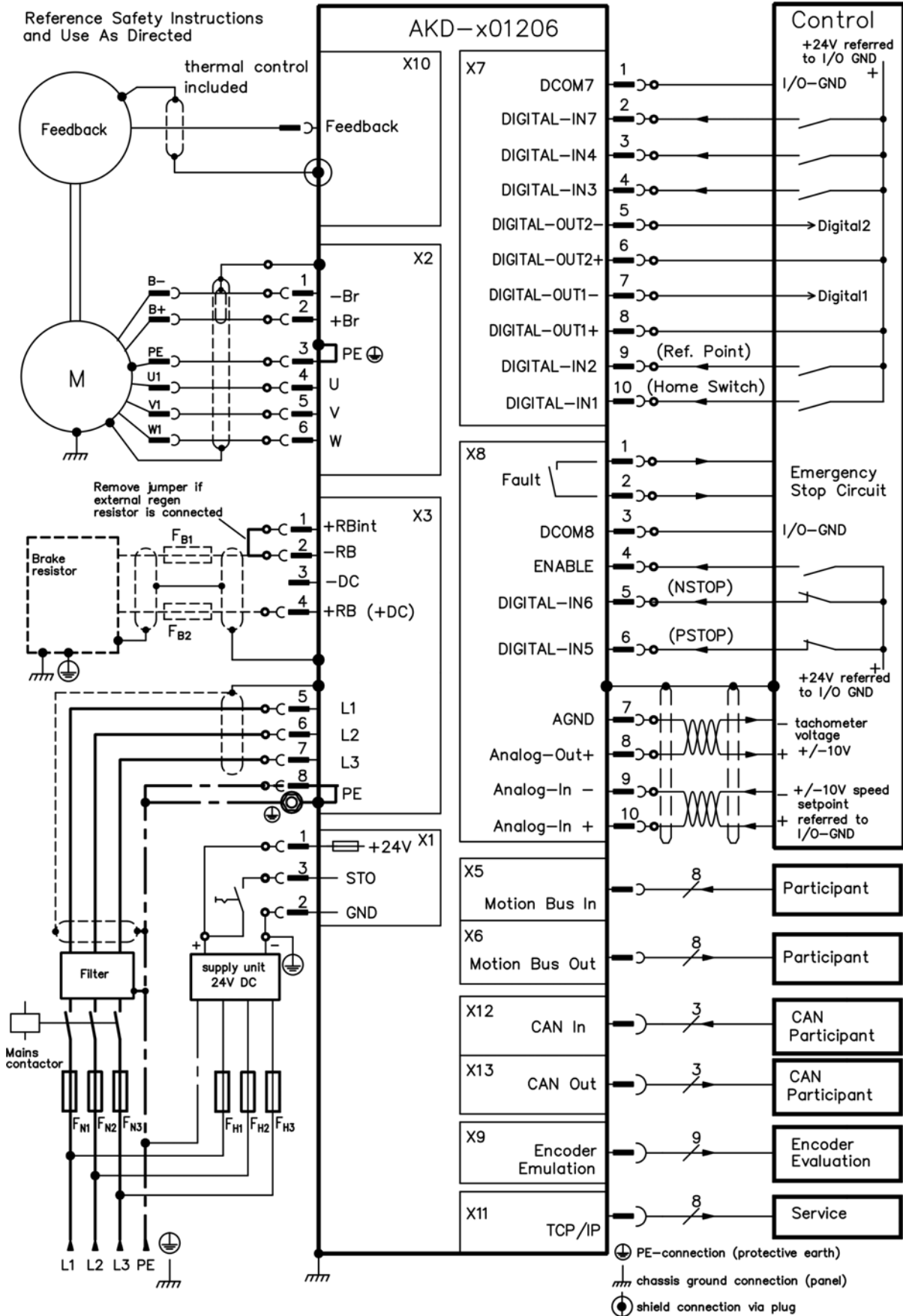
Connection Diagrams

| | | |
|-------|--|-----|
| 19.6 | Connection Diagram, AKD-x00306 to x00606..... | 264 |
| 19.7 | Connection Diagram, AKD-x01206..... | 265 |
| 19.8 | Connection Diagram, AKD-x02406 and AKD-xzzz07..... | 266 |
| 19.9 | 24 V Auxiliary Supply (X1)..... | 267 |
| 19.10 | Motor Connection..... | 268 |
| 19.11 | External Regen Resistor (X3)..... | 269 |
| 19.12 | DC Bus Link (X3)..... | 270 |
| 19.13 | Mains Supply Connection (X3, X4)..... | 271 |
| 19.14 | I/O Connection..... | 273 |
| 19.15 | Analog Output (X8)..... | 274 |
| 19.16 | Analog Input (X8)..... | 275 |
| 19.17 | Command encoder signal connection..... | 276 |
| 19.18 | Pulse / Direction signal connection..... | 278 |
| 19.19 | Up / Down signal connection..... | 278 |
| 19.20 | Feedback Connector (X10)..... | 279 |

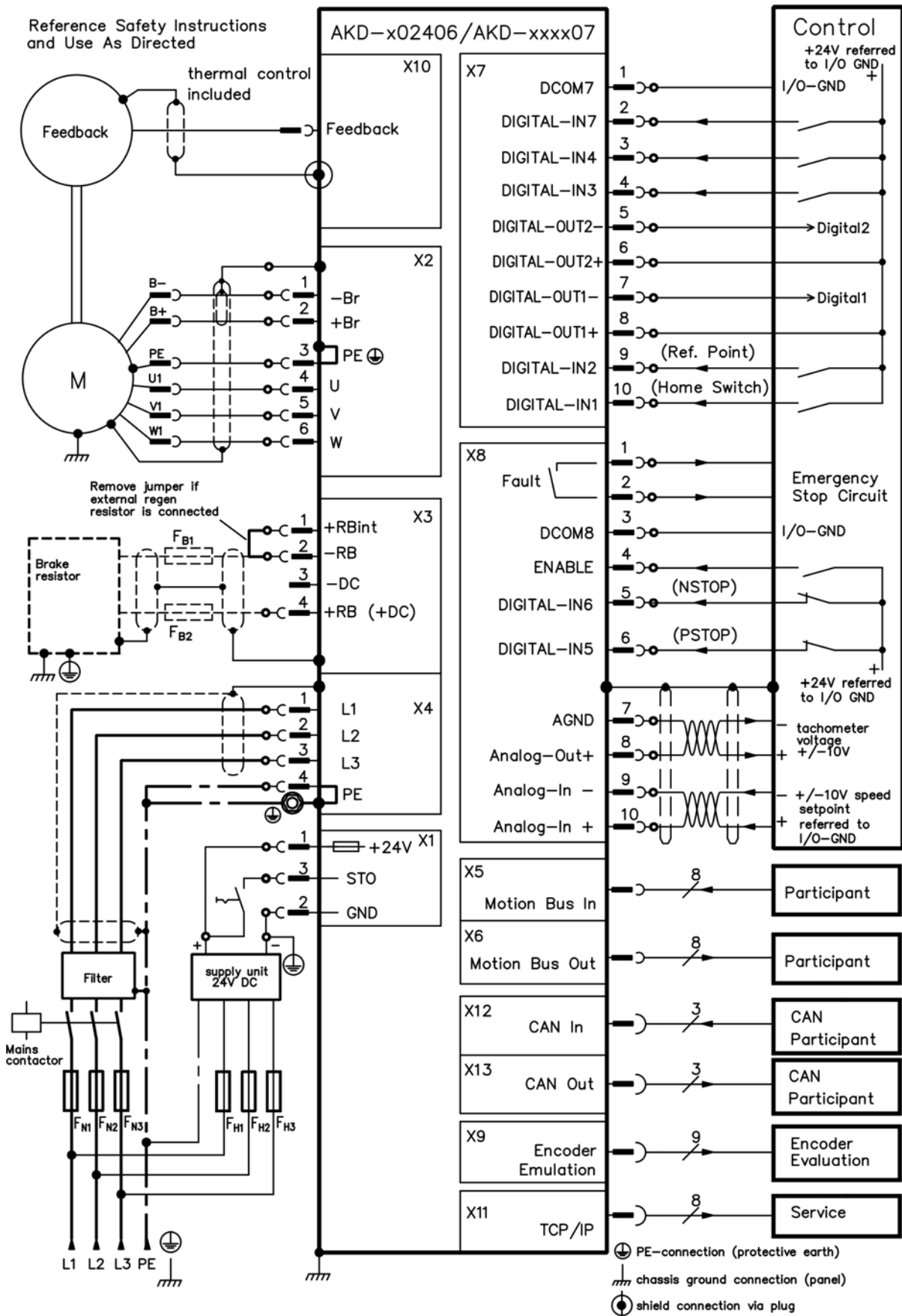
19.6 Connection Diagram, AKD-x00306 to x00606



19.7 Connection Diagram, AKD-x01206



19.8 Connection Diagram, AKD-x02406 and AKD-xzzz07



19.9 24 V Auxiliary Supply (X1)

The following diagram describes external 24 Vdc power supply, electrically isolated, for example, via an isolating transformer. The required current rating depends on the use of motor brake and option card (see page 1 or following).



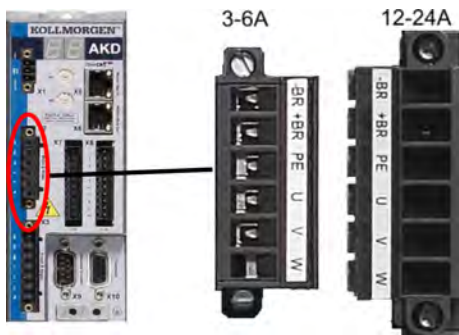
| Pin | Signal | Description |
|-----|--------|------------------------------|
| 1 | +24 | +24 Vdc Auxiliary voltage |
| 2 | GND | 24V Supply GND |
| 3 | STO | STO enable (Safe Torque Off) |

19.10 Motor Connection

Together with the motor supply cable and motor winding, the power output of the drive forms an oscillating circuit. Characteristics such as cable capacity, cable length, motor inductance, and frequency ("Electrical Data -xxxx06" (=> p. 1) or "Electrical Data -xxxx07" (=> p. 1)) determine the maximum voltage in the system.

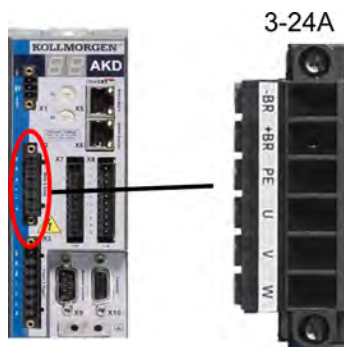
| | |
|---------------|--|
| NOTICE | <p>The dynamic voltage rise can lead to a reduction in the motor operating life and, on unsuitable motors, to flashovers in the motor winding.</p> <ul style="list-style-type: none"> • Only install motors with insulation class F (acc. to IEC60085) or above. • Only install cables that meet the requirements "Wiring" (=> p. 1). |
|---------------|--|

Connector X2 AKD-xxxx06



| Pin | Signal | Description |
|-----|--------|----------------------------------|
| 1 | -BR | Motor holding brake, negative |
| 2 | +BR | Motor holding brake, positive |
| 3 | PE | Protective earth (motor housing) |
| 4 | U | Motor phase U |
| 5 | V | Motor phase V |
| 6 | W | Motor phase W |

Connector X2 AKD-xxxx07



| Pin | Signal | Description |
|-----|--------|----------------------------------|
| 1 | -BR | Motor holding brake, negative |
| 2 | +BR | Motor holding brake, positive |
| 3 | PE | Protective earth (motor housing) |
| 4 | U | Motor phase U |
| 5 | V | Motor phase V |
| 6 | W | Motor phase W |

19.11 External Regen Resistor (X3)

For technical data on the brake circuit "Dynamic Braking" (=> p. 1). Fusing (such as fusible cut-outs) to be provided by the user see page 1.



| AKD-x00306 to AKD-x00606 (X3) | | |
|-------------------------------|--------|----------------------------------|
| Pin | Signal | Description |
| 1 | -RB | External Regen Resistor negative |
| 3 | +RB | External Regen Resistor positive |

| AKD-x01206 (X3) | | |
|-----------------|--------|---------------------------------|
| Pin | Signal | Description |
| 1 | +Rbint | Internal RegenResistor positive |
| 2 | -RB | External RegenResistor negative |
| 4 | +RB | External RegenResistor positive |

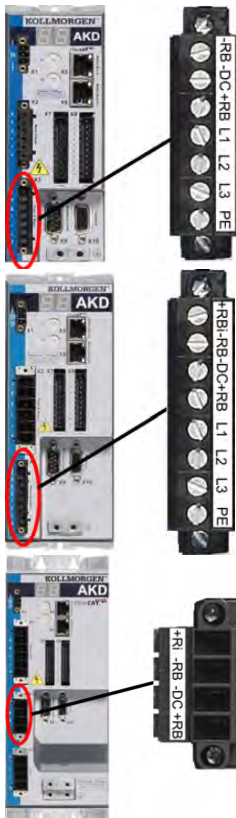
| AKD-x02406 & AKD-xzzz07 (X3) | | |
|------------------------------|--------|---------------------------------|
| Pin | Signal | Description |
| 2 | -RB | External RegenResistor negative |
| 4 | +RB | External RegenResistor positive |

19.12 DC Bus Link (X3)

The DC bus link can be connected in parallel so that the regen power is divided between all the drives that are connected to the same DC bus link circuit.

Every drive must have it's own power connection to mains voltage, even if the DC bus link is used.

NOTICE The drive can be destroyed if DC bus link voltages are different. Only drives with mains supply from the same mains (identical mains supply voltage) may be connected by the DC bus link. Use unshielded single cores (cross section see page 1) with a maximum length of 200 mm. Use shielded cables for longer lengths.



| AKD-x00306 to AKD-x00606 (X3) | | |
|-------------------------------|-----------|----------------------|
| Pin | Signal | Description |
| 2 | -DC | DC-Link Bus negative |
| 3 | +DC (+RB) | DC-Link Bus positive |

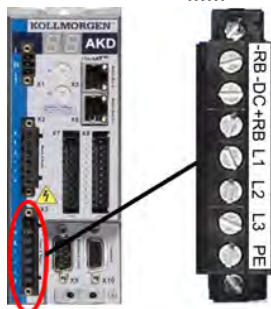
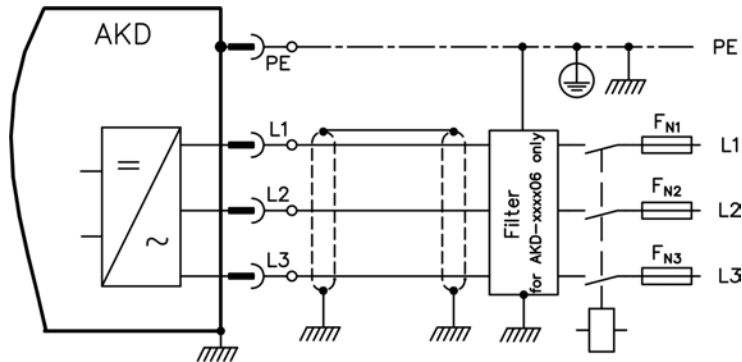
| AKD-x01206 (X3) | | |
|-----------------|-----------|----------------------|
| Pin | Signal | Description |
| 3 | -DC | DC-Link Bus negative |
| 4 | +DC (+RB) | DC-Link Bus positive |

| AKD-x02406 & AKD-xzzz07 (X3) | | |
|------------------------------|-----------|----------------------|
| Pin | Signal | Description |
| 3 | -DC | DC-Link Bus negative |
| 4 | +DC (+RB) | DC-Link Bus positive |

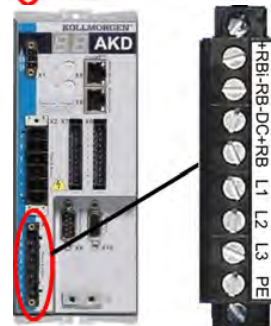
19.13 Mains Supply Connection (X3, X4)

19.13.1 Three Phase connection (all AKD types)

- Directly to 3-phase supply network, supply networks see page 1
- Filtering for AKD-xzzz06 to be provided by the user.
- Fusing (such as fusible cut-outs) to be provided by the user see page 1.



| AKD-x00306 to AKD-x00606 (X3) | | |
|-------------------------------|--------|------------------|
| Pin | Signal | Description |
| 4 | L1 | Line 1 |
| 5 | L2 | Line 2 |
| 6 | L3 | Line 3 |
| 7 | PE | Protective Earth |



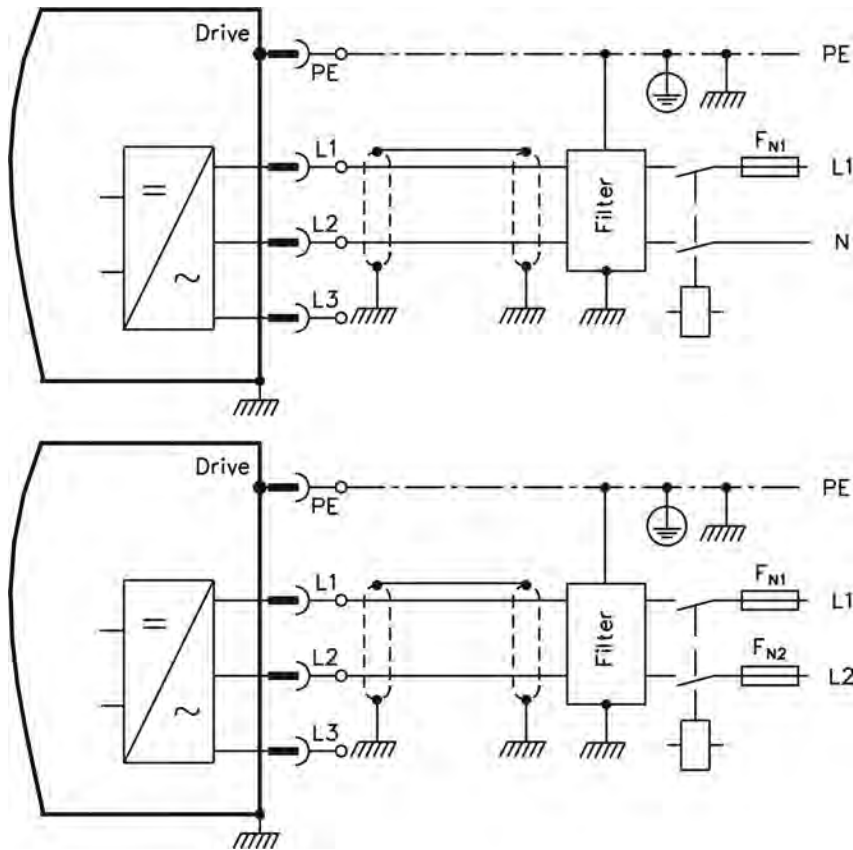
| AKD-x01206 (X3) | | |
|-----------------|--------|------------------|
| Pin | Signal | Description |
| 5 | L1 | Line 1 |
| 6 | L2 | Line 2 |
| 7 | L3 | Line 3 |
| 8 | PE | Protective Earth |



| AKD-x02406 & AKD-xzzz07 (X4) | | |
|------------------------------|--------|------------------|
| Pin | Signal | Description |
| 1 | L1 | Line 1 |
| 2 | L2 | Line 2 |
| 3 | L3 | Line 3 |
| 4 | PE | Protective Earth |

19.13.2 Single phase connection (AKD-xzzz06 only)

- Directly to single-phase supply network (120 V_{-10%} to 240 V^{+10%}) with neutral line or Directly to two-phase supply network (120 V_{-10%} to 240 V^{+10%}) without neutral line
- Supply networks see page 1
- Leave L3 open circuit
- Filtering to be provided by the user.
- Fusing (such as fusible cut-outs) to be provided by the user see page 1.

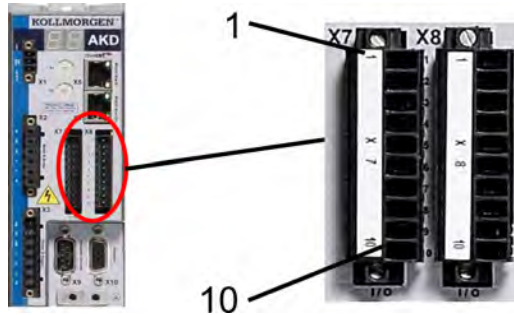


| AKD-x00306 to AKD-x00606 (X3) | | |
|-------------------------------|--------|-------------------|
| Pin | Signal | Description |
| 4 | L1 | Line 1 |
| 5 | L2 (N) | Neutral or Line 2 |
| 7 | PE | Protective Earth |

| AKD-x01206 (X3) | | |
|-----------------|--------|-------------------|
| Pin | Signal | Description |
| 5 | L1 | Line 1 |
| 6 | L2 (N) | Neutral or Line 2 |
| 8 | PE | Protective Earth |

19.14 I/O Connection

All standard digital and analog I/O signals are connected to X7 and X8.



19.14.1 I/O Connectors (X7 and X8)

| Conn. | Pin | Signal | Abbreviation | Function | Specials |
|-------|-----|--------------------|--------------------|--|------------------|
| X7 | 1 | Digital Common X7 | DCOM7 | Common line for X7 pins 2, 3, 4, 9, 10 | - |
| X7 | 2 | Digital Input 7 | DIGITAL-IN 7 | Programmable | - |
| X7 | 3 | Digital Input 4 | DIGITAL-IN 4 | Programmable | - |
| X7 | 4 | Digital Input 3 | DIGITAL-IN 3 | Programmable | - |
| X7 | 5 | Digital Output 2- | DIGITAL-OUT2- | Programmable | - |
| X7 | 6 | Digital Output 2+ | DIGITAL-OUT2+ | Programmable | - |
| X7 | 7 | Digital Output 1- | DIGITAL-OUT1- | Programmable | - |
| X7 | 8 | Digital Output 1+ | DIGITAL-OUT1+ | Programmable | - |
| X7 | 9 | Digital Input 2 | DIGITAL-IN 2 | Programmable | high speed |
| X7 | 10 | Digital Input 1 | DIGITAL-IN 1 | Programmable | high speed |
| X8 | 1 | Fault Relay Output | Fault Relay Output | Fault Relay Output | - |
| X8 | 2 | Fault Relay Output | Fault Relay Output | Fault Relay Output | - |
| X8 | 3 | Digital Common X8 | DCOM8 | Common line for X8 pins 4, 5, 6 | - |
| X8 | 4 | Digital Input 8 | DIGITAL-IN 8 | Output stage enable | not programmable |
| X8 | 5 | Digital Input 6 | DIGITAL-IN 6 | Programmable | - |
| X8 | 6 | Digital Input 5 | DIGITAL-IN 5 | Programmable | - |
| X8 | 7 | Analog Ground | AGND | Analog GND | - |
| X8 | 8 | Analog Output + | Analog-Out | Actual velocity voltage | - |
| X8 | 9 | Analog Input - | Analog-In- | Velocity set point | - |
| X8 | 10 | Analog Input + | Analog-In+ | | - |

Digital common lines for X7 and X8 are not common to each other.

The DCOMx lines should be connected to the 0V of the I/O supply when using sensors of type "Source" with digital inputs.

The DCOMx lines should be connected to the 24V of the I/O supply when using sensors of type "Sink" with digital inputs.

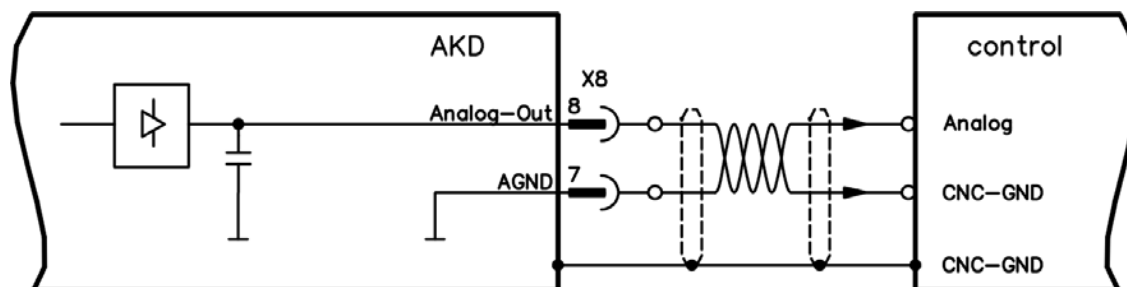
19.15 Analog Output (X8)

Terminals X8/7-8 can be used to output converted analog values of digital measurements recorded in the drive. A list of the pre-programmed functions is included in the WorkBench setup software.

Technical characteristics

- Output voltage range referring to AGND: ± 10 V
- Resolution: 16 Bit and fully monotonic
- Unadjusted offset: < 50 mV
- Offset drift typ: $250 \mu\text{V}/^\circ\text{C}$
- Gain or slope tolerance: $\pm 3\%$
- Nonlinearity: $< 0.1\%$ of full scale or 10 mV
- Output impedance: 110 ohms
- Specification complies with IEC 61131-2 Table 11
- -3 dB Bandwidth: > 8 kHz
- Maximum output current: 20 mA
- Capacitive load: any value but response speed limited by max Iout and by Rout
- Protected for short circuit to AGND

Analog Output Wiring Diagram



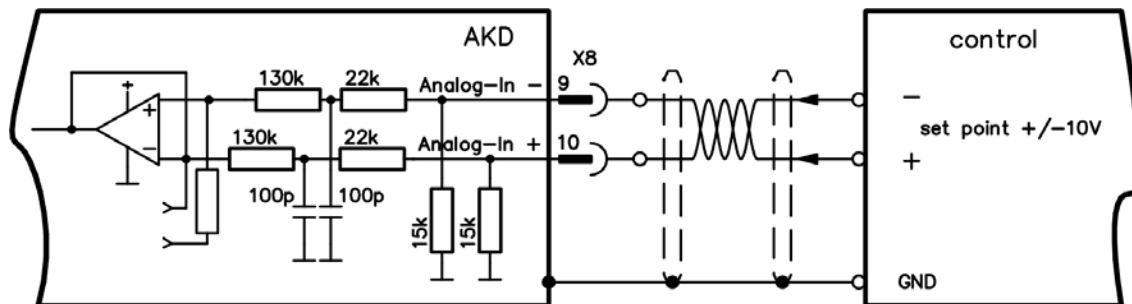
19.16 Analog Input (X8)

The drive is fitted with one differential input for analog torque, velocity, or position control.

Technical characteristics

- Differential input voltage range: ± 12.5 V
- Maximum input voltage referring to I/O Return: -12.5, +16.0 V
- Resolution: 16 Bit and fully monotonic
- Unadjusted offset: < 50 mV
- Offset drift typ: $250 \text{ V} / ^\circ \text{C}$
- Gain or slope tolerance: $\pm 3\%$
- Nonlinearity: < 0.1% of full scale or 12.5 mV
- Common Mode Rejection Ratio: > 30 dB at 60 Hz
- Input impedance: > 13k Ohms
- Signal to noise ratio referred to full scale:
 - AIN.CUTOFF = 3000 Hz: 14 bit
 - AIN.CUTOFF = 800 Hz: 16 bit

Analog Input Wiring Diagram



Application examples for set point input Analog-In:

- adjustable external current limit
- reduced-sensitivity input for setting-up/jog operation
- pre-control/override

Defining the direction of rotation

Standard setting: clockwise rotation of the motor shaft (looking at the shaft end) affected by positive voltage between terminal X8/10 (+) and terminal X8/9 (-)

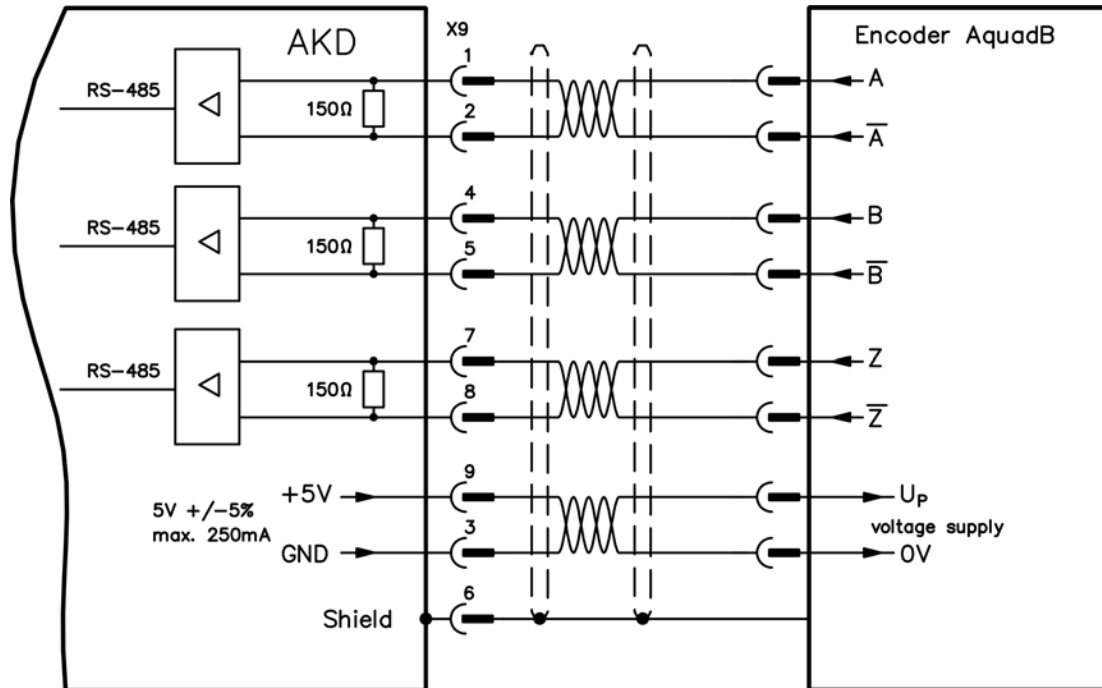
To reverse the direction of rotation, swap the connections to terminals X8/9-10, or change the DRV.DIR parameter in the "Feedback" screen page.

19.17 Command encoder signal connection

19.17.1 Incremental encoder input 5 V (X9)

A 5 V A quad B encoder, or the encoder emulation output of another drive can be connected to this input and used as a commander encoder, dual loop feedback, gearing or camming input. Don't use for primary motor feedback connection!

Connection Diagram



19.17.2 Incremental encoder input 24 V (X7)

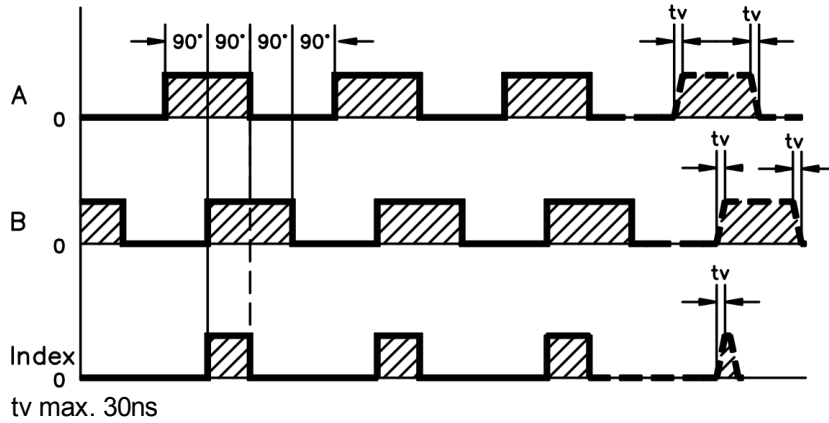
A 24 V A quad B encoder can be connected to the digital inputs 1 and 2 and used as a commander encoder, dual loop feedback, gearing or camming input. Don't use for primary motor feedback connection!

Connection Diagram

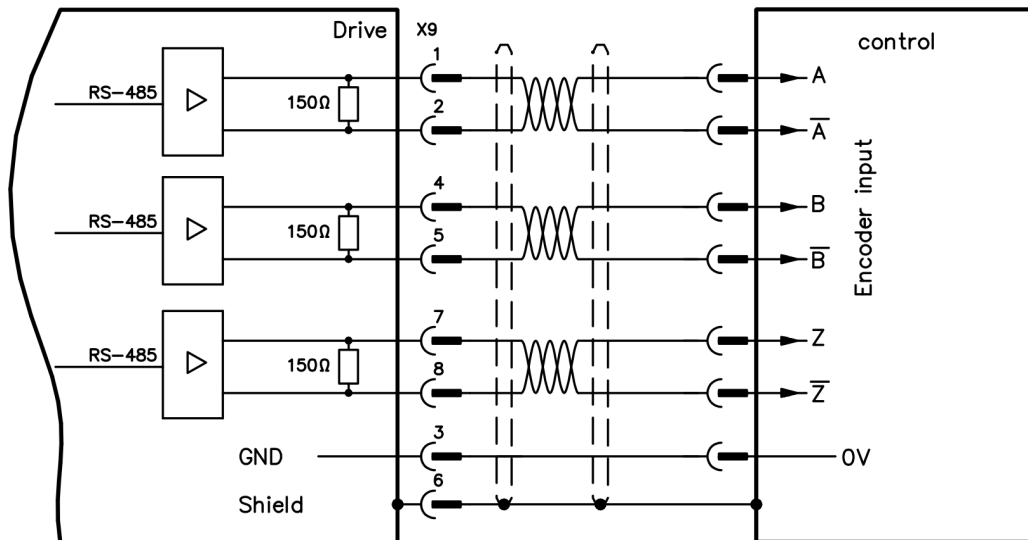
19.17.3 Emulated Encoder Output (EEO) - A quad B (X9)

The drive calculates the motor shaft position from the cyclic- absolute signals of the primary feedback, generating incremental-encoder compatible pulses from this information. Pulse outputs on the SubD connector X9 are 3 signals, A, B and Index, with 90° phase difference (i.e. in quadrature, hence the alternative term “A quad B” output), with a zero pulse.

The resolution (before multiplication) can be set by the DRV.EMUERES parameter. Use the DRV.EMUEZOFFSET parameter to adjust + save the zero pulse position within one mechanical turn. The drivers operate from an internal supply voltage.



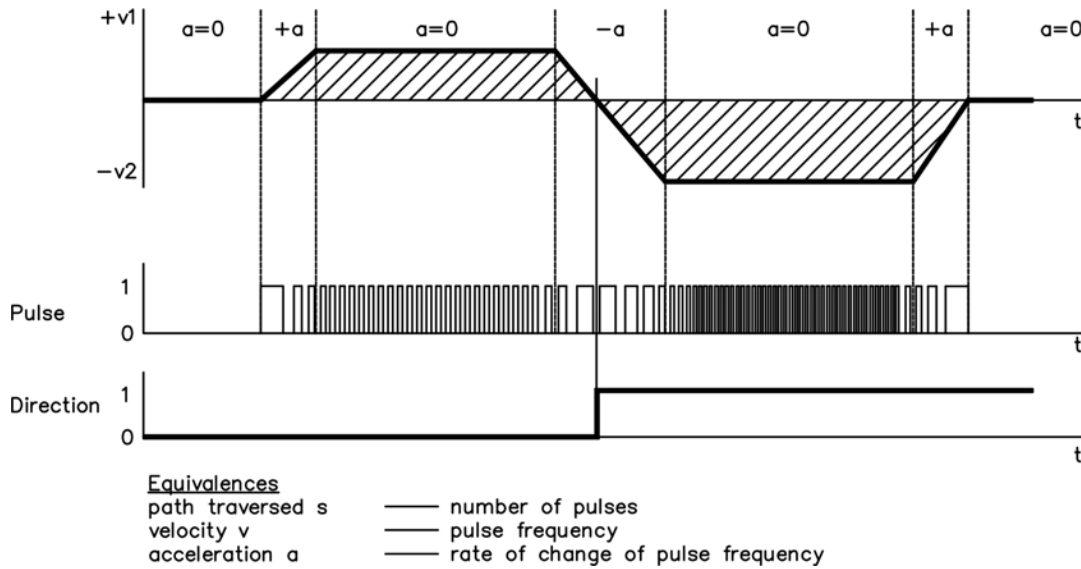
Connection Diagram



19.18 Pulse / Direction signal connection

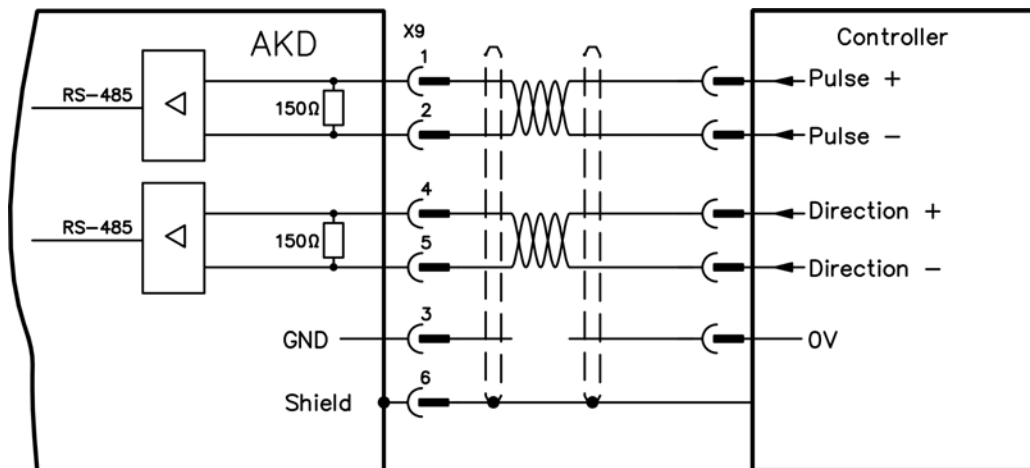
The drive can be connected to a stepper-motor controller. Set parameters for the drive with WorkBench. The number of pulses can be adjusted, so that the drive can be adapted to match any stepper controller.

Speed profile and signal diagram



19.18.1 Pulse / Direction input 5 V (X9)

Connection to 5 V signal level stepper-motor controllers.



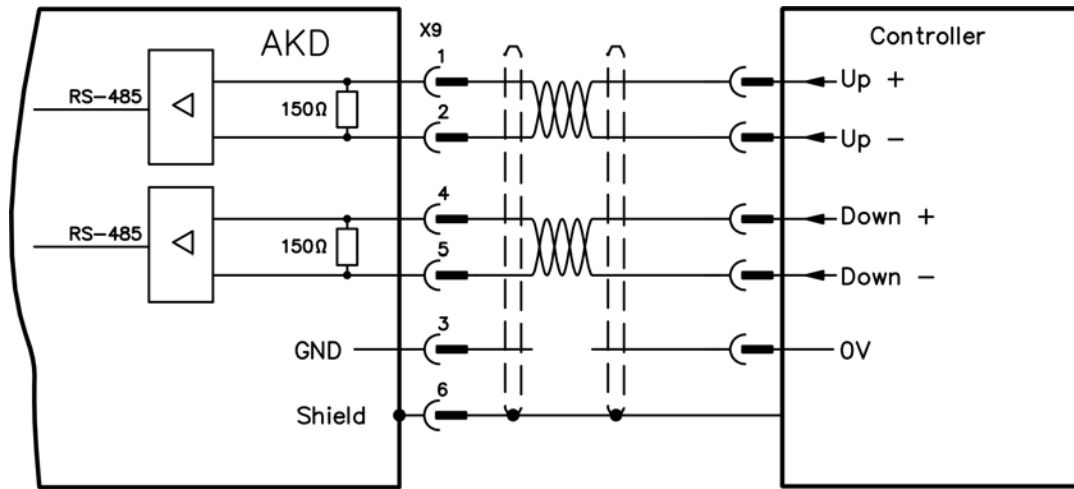
19.18.2 Pulse / Direction input 24 V (X7)

Connection to 24 V signal level stepper-motor controllers.

19.19 Up / Down signal connection

19.19.1 Up / Down input 5 V (X9)

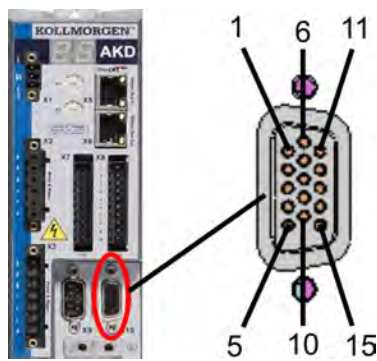
The drive can be connected to a third-party controller which delivers 5 V up-down signals



19.19.2 Up / Down input 24 V (X7)

The drive can be connected to a third-party controller which delivers 24 V up-down signals.

19.20 Feedback Connector (X10)

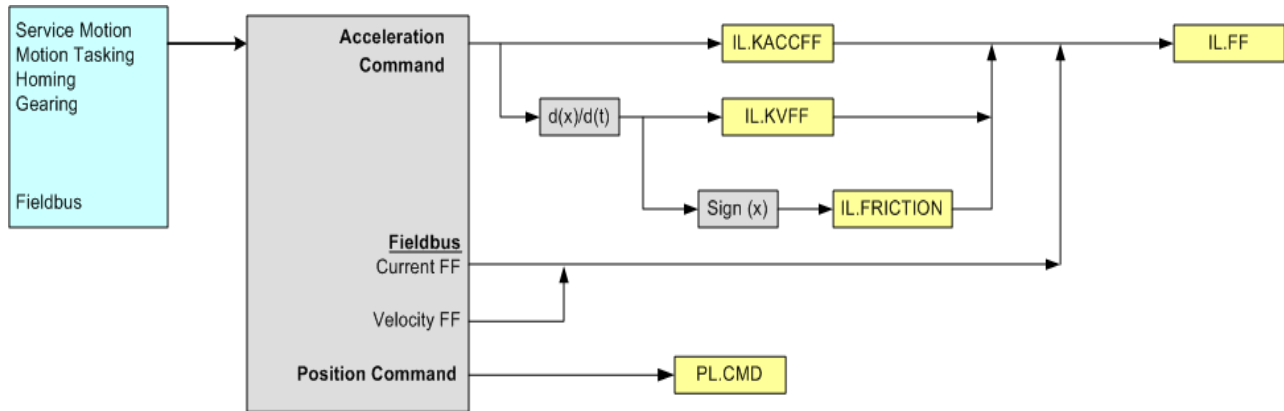


| Pin | SFD | Resolver | BiSS (analog) | EnDAT 2.1 | EnDAT 2.2 | Hiper-face | Sine Encoder +Hall | Incremental Encoder +Hall |
|-----|--------|----------------------------|---------------|-----------|-----------|------------|--------------------|---------------------------|
| 1 | - | - | - | - | - | - | Hall U | Hall U |
| 2 | - | - | CLOCK+ | CLOCK+ | CLOCK+ | - | Hall V | Hall V |
| 3 | - | - | CLOCK- | CLOCK- | CLOCK- | - | Hall W | Hall W |
| 4 | SENSE+ | - | SENSE+ | SENSE+ | SENSE+ | SENSE+ | SENSE+ | SENSE+ |
| 5 | SENSE- | - | SENSE- | SENSE- | SENSE- | SENSE- | SENSE- | SENSE- |
| 6 | COM+ | R1 Ref+ | DATA+ | DATA+ | DATA+ | DATA+ | Zero+ | Zero+ |
| 7 | COM- | R2 Ref- | DATA- | DATA- | DATA- | DATA- | Zero- | Zero- |
| 8 | - | Thermal control (PTC) | | | | | | |
| 9 | - | Thermal control (PTC, GND) | | | | | | |
| 10 | +5 V | - | +5 V | +5 V | +5 V | +8 to +9 V | +5 V | +5 V |
| 11 | 0 V | - | 0 V | 0 V | 0 V | 0 V | 0 V | 0 V |
| 12 | - | S1 SIN+ | A+ | A+ | - | SIN+ | A+ | A+ |
| 13 | - | S3 SIN- | A- | A- | - | SIN- | A- | A- |
| 14 | - | S2 COS+ | B+ | B+ | - | COS+ | B+ | B+ |
| 15 | - | S4 COS- | B- | B- | - | COS- | B- | B- |

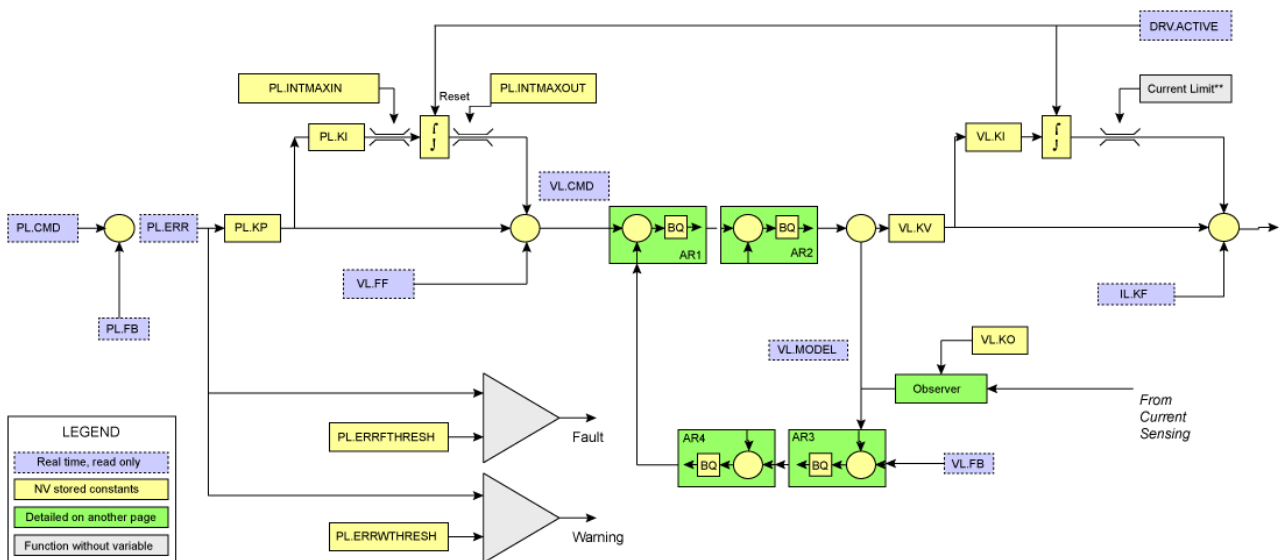
Block Diagrams

| | | |
|--------------|--|------------|
| 19.21 | Block Diagram for Current Loop..... | 282 |
| 19.22 | Block Diagram for Position/Velocity Loop..... | 282 |

19.21 Block Diagram for Current Loop



19.22 Block Diagram for Position/Velocity Loop



Appendix A

Fieldbus Manuals

- *AKD CANopen Manual*
- *AKD EtherCAT Manual*
- *AKD Modbus Manual*

This page intentionally left blank.

20 Modbus

20.1 Overview

Modbus is a simple communication protocol often used for reporting data from an industrial device to an HMI or PLC. Modbus TCP extends the protocol to TCP/IP networks by embedding the same Protocol Data Unit within TCP/IP packets.

The AKD supports a Modbus TCP service channel for up to 3 simultaneous client connections, alongside the Workbench configuration software.

Most drive parameters are supported over Modbus TCP, with the exception of commands which use character strings. For information about the Modbus protocol, please see <http://www.Modbus.org/specs.php>.

20.2 Installation and Setup

Modbus TCP is provided over the service port on the top of the drive (X11 connector), the same as for Workbench. Connect the drive and a client device such as an HMI to a working Ethernet network. For ease of testing and configuration, connect a PC running Workbench to the same network.

After booting, the drive will flash its Ethernet IP address on the front display – the drive can be accessed at this address for Modbus on port 502. Workbench uses the same address, but a different port number.

Once the devices are connected, the client device can open a connection to the AKD using these settings:

- IP Address: read from drive display or Workbench connect screen
- Port: 502
- Add Modbus CRC code: No

20.3 Overview of Messaging

All parameters with an internal data width of 32-bit and lower are mapped to Modbus as 32-bit (2 register) values. The contents of both registers must be read or written in the same message, by setting the Starting Address to the lowest index of the parameter and the Quantity of Registers to two.

All parameters with an internal data width of 64-bit are mapped to Modbus as 64-bit (4 register) values. The contents of all four registers must be read or written in the same message, by setting the Starting Address to the lowest index of the parameter and the Quantity of Registers to four.

Parameters are scaled as in the Terminal window of Workbench. But note that floating point values are scaled by an additional factor of 1000 to retain precision.

Drive commands which do not accept or return a value (such as DRV.EN) are executed by writing to the parameter. The data value is ignored. Read attempts will result in an error response.

20.4 Supported Functions

Two Modbus functions are currently supported:

- Read Holding Registers. Function code = 0x03 (3)
- Write Multiple Registers. Function code = 0x10 (16)

To allow only 32bit and 64bit access, the quantity of registers is limited to 2 registers for 32bit variables and 4 registers for 64bit variables. Other quantities shall return an exception response.

20.5 Read Holding Registers (0x03)

This function code is used to read all registers of one drive parameter.

Request

| | | |
|-----------------------|---------|--|
| Function Code | 1 Byte | 0x03 |
| Starting Address | 2 Bytes | 0x0000-0xFFFF (see Modbus Parameter Table) |
| Quantity of Registers | 2 Bytes | 2 (32-bit values) or 4 (64-bit values) |

Normal Response

| | | |
|----------------|--------------|--------|
| Function Code | 1 Byte | 0x03 |
| Byte Count | 1 Byte | 2 x N* |
| Register Value | N* x 2 Bytes | Data |

*N = Quantity of Registers

Error Response

| | | |
|----------------|--------|---------------------|
| Function Code | 1 Byte | 0x83 |
| Exception Code | 1 Byte | See exception codes |

Following is an example of reading the position loop feedback PL.FB at index 588 (0x024C), with an actual value of 0x1 2A05 F200.

Request

| | |
|--------------------------|----|
| Function | 03 |
| Starting Address Hi | 02 |
| Starting Address Lo | 4C |
| Quantity of Registers Hi | 00 |
| Quantity of Registers Lo | 04 |

Response

| | |
|-----------------|----|
| Function | 03 |
| Byte Count | 08 |
| Register 588 Hi | 00 |
| Register 588 Lo | 00 |
| Register 589 Hi | 00 |
| Register 589 Lo | 01 |
| Register 590 Hi | 2A |
| Register 590 Lo | 05 |
| Register 591 Hi | F2 |
| Register 591 Lo | 00 |

20.6 Write Multiple Registers (0x10)

This function code is used to write all registers of one drive parameter.

Request

| | | |
|-----------------------|--------------|---|
| Function Code | 1 Byte | 0x10 |
| Starting Address | 2 Bytes | 0x0000-0xFFFF (see Modbus index in parameter lists) |
| Quantity of Registers | 2 Bytes | 2 (32-bit values) or 4 (64-bit values) |
| Byte Count | 1 Byte | 2 x N* |
| Register Value | N* x 2 Bytes | Data |

*N = Quantity of Registers

Normal Response

| | | |
|-----------------------|---------|---------------|
| Function Code | 1 Byte | 0x10 |
| Starting Address | 2 Bytes | 0x0000-0xFFFF |
| Quantity of Registers | 2 Bytes | 2 or 4 |

Error Response

| | | |
|----------------|--------|---------------------|
| Function Code | 1 Byte | 0x90 |
| Exception Code | 1 Byte | See exception codes |

Following is an example of setting the mode of digital input 1 (DIN1.MODE) at index 122 (0x007A) to a value of 3.

Request

| | |
|--------------------------|----|
| Function | 10 |
| Starting Address Hi | 00 |
| Starting Address Lo | 7A |
| Quantity of Registers Hi | 00 |
| Quantity of Registers Lo | 02 |
| Byte Count | 04 |
| Register 122 Hi | 00 |
| Register 122 Lo | 00 |
| Register 123 Hi | 00 |
| Register 123 Lo | 03 |

Response

| | |
|--------------------------|----|
| Function | 10 |
| Starting Address Hi | 00 |
| Starting Address Lo | 7A |
| Quantity of Registers Hi | 00 |
| Quantity of Registers Lo | 02 |

20.7 Exception Response Codes

Standard Codes

| Description | Exception Code |
|---|----------------|
| Illegal Function | 1 |
| Illegal data address | 2 |
| Illegal Data Value | 3 |
| Slave Device Failure | 4 |
| Acknowledge | 5 |
| Slave Device Busy | 6 |
| Memory Parity Error | 8 |
| Gateway Path Unavailable | 10 |
| Gateway Target Device Failed To Respond | 11 |

Manufacturer Codes

| Description | Exception Code |
|---|----------------|
| Illegal Block Size (must be 2 registers for 32-bit and 4 registers for 64-bit parameters) | 32 |
| Unknown Parameter (bad address) | 33 |
| Error Processing Command (eg. Data value is greater than maximum) | 34 |
| Byte Count field is invalid in request | 35 |

20.8 Modbus

20.8.1 Overview

Modbus is a simple communication protocol often used for reporting data from an industrial device to an HMI or PLC. Modbus TCP extends the protocol to TCP/IP networks by embedding the same Protocol Data Unit within TCP/IP packets. The AKD supports a Modbus TCP service channel for connections with up to three masters.

Most drive parameters are supported over Modbus TCP (see 20.10 Modbus Parameter Table, with the exception of commands which output character strings. For information about the Modbus protocol, please see:

<http://www.Modbus.org/specs.php>.

20.8.2 Modbus Installation and Setup

Modbus TCP is provided over the service port on the top of the drive (X11 connector, the connector used for WorkBench). Connect the drive and a device such as an HMI to a working Ethernet network. For ease of testing and configuration, connect a PC running WorkBench to the same network.

After booting, the drive will flash its Ethernet IP address on the front display. The drive can be accessed at this address for Modbus on port 502. WorkBench uses the same address, but a different port number.

Once the devices are connected, the connected device can open a connection to the AKD using these settings:

- IP Address: read from drive display or Workbench connect screen
- Port: 502
- Add Modbus CRC code: No

20.8.3 Overview of Messaging

All parameters with an internal data width of 32-bit and lower are mapped to Modbus as 32-bit (2 register) values. The contents of both registers must be read or written in the same message, by setting the Starting Address to the lowest index of the parameter and the Quantity of Registers to two.

All parameters with an internal data width of 64-bit are mapped to Modbus as 64-bit (4 register) values. The contents of all four registers must be read or written in the same message by setting the Starting Address to the lowest index of the parameter and the Quantity of Registers to four.

Parameters are scaled as in the Terminal window of WorkBench, but note that floating point values are scaled by an additional factor of 1,000 to retain precision. Drive commands which do not accept or return a value (such as DRV.EN) are executed by writing to the parameter. The data value is ignored and read attempts will result in a value of zero.

20.8.4 Supported Functions

Two Modbus functions are currently supported:

- Read Holding Registers. Function code = 0x03 (3).
- Write Multiple Registers. Function code = 0x10 (16).

To allow only 32-bit and 64-bit access, the quantity of registers is limited to two registers for 32-bit variables and four registers for 64-bit variables. You can also read/write to a single register for certain parameters. Other quantities return an exception response.

20.8.5 Read Holding Registers (0x03)

This function code is used to read all registers of one drive parameter.

Request

| | | |
|------------------------------|---------|--|
| Function Code | 1 Byte | 0x03 |
| Starting Address | 2 Bytes | 0x0000-0xFFFF (see 20.10 Modbus Parameter Table) |
| Quantity of Registers | 2 Bytes | Two 32-bit values or four 64-bit values |

Normal Response

| | | |
|-----------------------|--------------|--------|
| Function Code | 1 Byte | 0x03 |
| Byte Count | 1 Byte | 2 x N* |
| Register Value | N* x 2 Bytes | Data |

*N = Quantity of Registers

Error Response

| | | |
|-----------------------|--------|------------------------------|
| Function Code | 1 Byte | 0x83 |
| Exception Code | 1 Byte | See Exception Response Codes |

Following is an example of reading the position loop feedback PL.FB at index 588 (0x024C), with an actual value of 0x1 2A05 F200.

Request

| | |
|------------------------------|-------|
| Function | 03 |
| Starting Address | 02 4C |
| Quantity of Registers | 00 04 |

Response

| | |
|---------------------|-------|
| Function | 03 |
| Byte Count | 08 |
| Register 588 | 00 00 |
| Register 589 | 00 01 |
| Register 590 | 2A 05 |
| Register 591 | F2 00 |

20.8.6 Write Multiple Registers (0x10)

This function code is used to write all registers of one drive parameter.

Request

| | | |
|------------------------------|--------------|---|
| Function Code | 1 Byte | 0x10 |
| Starting Address | 2 Bytes | 0x0000-0xFFFF (see Modbus index in parameter lists) |
| Quantity of Registers | 2 Bytes | 2 (32-bit values) or 4 (64-bit values) |
| Byte Count | 1 Byte | 2 x N* |
| Register Value | N* x 2 Bytes | Data |

*N = Quantity of Registers

Normal Response

| | | |
|------------------------------|---------|---------------|
| Function Code | 1 Byte | 0x10 |
| Starting Address | 2 Bytes | 0x0000-0xFFFF |
| Quantity of Registers | 2 Bytes | 2 or 4 |

Error Response

| | | |
|-----------------------|--------|------------------------------|
| Function Code | 1 Byte | 0x90 |
| Exception Code | 1 Byte | See Exception Response Codes |

Following is an example of setting the mode of digital input 1 (DIN1.MODE) at index 122 (0x007A) to a value of 3.

Request

| | |
|------------------------------|-------|
| Function | 10 |
| Starting Address | 00 7A |
| Quantity of Registers | 00 02 |
| Byte Count | 04 |
| Register 122 | 00 00 |
| Register 123 | 00 03 |

Response

| | |
|------------------------------|-------|
| Function | 10 |
| Starting Address | 00 7A |
| Quantity of Registers | 00 02 |

20.8.7 Exception Response Codes

Standard Codes

| Description | Exception Code |
|------------------|----------------|
| Illegal Function | 1 |

| Description | Exception Code |
|---|----------------|
| Illegal data address | 2 |
| Illegal Data Value | 3 |
| Slave Device Failure | 4 |
| Acknowledge | 5 |
| Slave Device Busy | 6 |
| Memory Parity Error | 8 |
| Gateway Path Unavailable | 10 |
| Gateway Target Device Failed To Respond | 11 |

Manufacturer Codes

| Description | Exception Code |
|--|----------------|
| Illegal Block Size (must be two registers for 32-bit and four registers for 64-bit parameters) | 32 |
| Unknown Parameter (bad address) | 33 |
| Error Processing Command (such as "Data value is greater than maximum") | 34 |
| Byte Count field is invalid in request | 35 |

20.9 Modbus Dynamic Mapping

Modbus dynamic mapping allows you to map any of the fixed register addresses to a new register address. By re-ordering the sequence block, read/write access of the re-mapped parameters becomes possible.

In general, all parameters are mapped as 32-bit values and occupy at least two Modbus registers. Parameters with a 64-bit data width occupy four Modbus registers. All 64-bit parameters are also mapped as 32-bit value (two registers) starting at register address 2000 (see 20.11 Modbus 64-bit Parameters to 32-bit Mapping).

20.9.1 Configuring Dynamic Mapping

The start address for dynamically mapped registers is 8192 (0x2000).

Mapping works as follows:

1. Enable dynamic mapping by writing a 1 (one) to register address 4096 (0x1000). Writing a 1 enables the configuration mode of dynamic mapping. Writing a zero puts it into "run" mode, in which you can use the mapped parameter.
2. The mapping is now register oriented. Next, write the valid fixed register address of the parameter you would like to map to an address starting at 8192. To map a 32-bit parameter, you must map the corresponding two registers.

Example

AOUT.VSCALE

Write 36 (hi-word) to address 8192 and 37 (lo-word) to address 8193.

The register oriented mapping allows you to optimize the data width for block transfers. To map a parameter with a data width of 16-bit, only the lo-word register needs to be mapped.

3. Example

MODBUS.DIO

Write 941 to address 8194

MODBUS.DIO fixed addresses are 940 (hi-word) and 941 (lo-word). Only the lo-word is of interest (16-bit value), because only the lo-word contains the actual value.

4. Disable dynamic mapping by writing a 0 (zero) to register address 4096 (0x1000). Disabling dynamic mapping puts it in a mode where you can now access the value of the mapped parameter and use the mapped register.

The fixed mapping starting at address 0 is still valid.

An example of the entire mapping process is included below:

Example

To re-map MODBUS.DIO (16-bit), MODBUS.DRVSTAT (16-bit), and DRV.MOTIONSTAT (32-bit):

1. Enable dynamic mapping.
2. Block write transfer to address 0x2000 (8192).
Values: 941, 945, 268, 269
3. Modbus telegram:
Function: 0x10; Address: 0x20, 0x00; Number of Registers: 0x00, 0x04; Data Length in Byte: 0x08; Data: 0x03, 0xAD, 0x03, 0xB1, 0x01, 0x0C, 0x01, 0x0d
4. Disable dynamic mapping.
5. Execute DRV.NVSAVE to store the mapping table non-volatile.
6. The re-mapped addresses are:
0x2000 : MODBUS.DIO
0x2001 : MODBUS.DRVSTAT
0x2002 : DRV.MOTIONSTAT (hi-word)
0x2003 : DRV.MOTIONSTAT (lo-word)

20.9.2 Saving and Resetting Dynamic Mapping

Use the following parameters to save and reset dynamic mapping:

- DRV.NVSAVE. Save parameters AND the dynamic mapping table non-volatile.
- DRV.RSTVAR. Set parameters to default value AND clear the dynamic mapping table.

20.9.3 Scaling parameters

You can scale parameters accessed through Modbus. This scaling is independent from the units of UNIT.PIN and UNIT.POUT. Instead, the user units are defined by MODBUS parameters:

- MODBUS.PIN (default = 1)
- MODBUS.POUT (default = 1)
- MODBUS.PSCALE (default = 20)

With the default settings shown above, the units are as follows:

- Position [counts (PSCALE bit per rev)]
- Velocity [counts/sec]
- Acceleration / Deceleration [counts/sec²]
- Current (Torque) [mA]

20.9.4 Modbus specific registers (Parameters)

The following parameters provide a shortcut to digital I/O, status information, and commands.

| Parameter | Bit |
|----------------|---|
| MODBUS.DIO | Bit 0 to 6: DIN.STATES Bit 16 and 17: DOUT.STATES |
| MODBUS.DRVSTAT | Parameters collected: Bit 0: DRV.ACTIVE (drive active) Bit 1: STO.STATUS (STO status) Bit 2: HWLS.POSSTATE (positive HW limit) Bit 3: HWLS.NEGSTATE (negative HW limit) Bit 4: SWLS.STATE (positive SW limit) Bit 5: SWLS.STATE (negative SW limit) |
| MODBUS.DRV | Bit 0: DRV.STOP (write 1 to execute) Bit 1: DRV.EN (write 1 to enable drive) and DRV.DIS (write 0 to disable drive) |
| MODBUS.HOME | Bit 0: HOME.MOVE (write 1 to execute) Bit 1: HOME.SET (write 1 to execute) |
| MODBUS.MOTOR | Bit 0: MOTOR.BRAKE Bit 1: MOTOR.BRAKERLS |
| MODBUS.MT | Bit 0: MT.CLEAR number from MT.NUM Bit 1: MT.CONTINUE Bit 2: MT.LOAD number from MT.NUM Bit 3: MT.SET Bit 4: MT.MOVE (number from MT.NUM) Note: when both bit 2 and 4 are set then MT.SET is executed first and MT.MOVE second to start the motion task. |
| MODBUS.SM | Bit 0: One direction (sets SM.MODE to either 0 or 1) Bit 1: Start move Edge-triggered: <ul style="list-style-type: none"> • 0 → 1 : Start motion (execution of SM.MOVE) • 1 → 0 : Stop motion (execution of DRV.STOP) |

20.9.5 32-bit versus 16-bit Values

Modbus mirrors all parameters either as 32-bit or 64-bit values. The minimum data width of 32-bit supports backward compatibility, which means that if the internal data width changes, this change has no effect on the register address of consecutive parameters. Dynamic mapping allows you to map only the registers that are relevant to the application and thus reduces communication overhead.

20.9.6 Mapping of 64-bit Parameters to 32-bit Parameters

All 64-bit parameters are mapped as 32-bit parameters starting at register address 2000. Only the 32-bit lo-word is mapped with the same scaling as the 64-bit parameter. Chapter 7 contains the general mapping table (chapter 7.2) and the 64-bit to 32-bit mapping table (chapter 7.3).

20.9.7 Fault Registers

Fault registers MODBUS.FAULT1 (register address 954) to MODBUS.FAULT10 (register address 972) contain the fault state of the drive.

You can obtain the current fault state as follows:

1. Read MODBUS.FAULT1.
2. If MODBUS.FAULT1 is zero, then the drive is fault free.
3. A nonzero value of MODBUS.FAULT1 is a fault number.
4. If MODBUS.FAULT1 is nonzero, the following fault registers (MODBUS.FAULT2 to MODBUS.FAULT10) contain possible further fault numbers.
5. A value of zero indicates no further faults

20.9.8 Mapping Table

The tables below show Modbus addresses and attributes. The attributes are described as follows:

| Attribute | Description |
|-----------------------|--|
| 64-bit Parameter | Parameter is internally a 64-bit parameter. |
| 8-bit, 16-bit, 32-bit | Internal data size, mapped as 32-bit (2 registers). |
| 64-bit | Internal data size, mapped as 64-bit (4 registers). |
| Low 32 bit word | Internally a 64-bit value, only the low 32-bit word is mapped (2 registers). |
| Signed | Sign bit is significant (negative/positive values are accepted). |
| Command | Executes a command. |
| Command, data width | Executes a command with numerical argument (for example, MT.NUM). |

o

20.10 Modbus Parameter Table

| Parameter | Modbus Register Address | 64-bit | Attribute |
|------------------|-------------------------|--------|--------------------|
| AIN.CUTOFF | 0 | | 32-bit |
| AIN.DEADBAND | 2 | | 16-bit |
| AIN.ISCALE | 4 | | 32-bit |
| AIN.OFFSET | 6 | | 16-bit, signed |
| AIN.PSCALE | 8 | 64-bit | 64-bit |
| AIN.VALUE | 12 | | 16-bit |
| AIN.VSCALE | 14 | | 32-bit |
| AIN.ZERO | 16 | | Command |
| AOUT.ISCALE | 18 | | 32-bit |
| AOUT.MODE | 20 | | 16-bit |
| AOUT.OFFSET | 22 | | 16-bit, signed |
| AOUT.PSCALE | 24 | 64-bit | 64-bit |
| AOUT.VALUE | 28 | 64-bit | 64-bit, signed |
| AOUT.VALUEU | 32 | 64-bit | 64-bit, signed |
| AOUT.VSCALE | 36 | | 32-bit |
| BODE.EXCITEGAP | 38 | | 8-bit |
| BODE.FREQ | 40 | | 32-bit |
| BODE.IAMP | 42 | | 16-bit, signed |
| BODE.INJECTPOINT | 44 | | 8-bit |
| BODE.MODE | 46 | | 8-bit |
| BODE.MODETIMER | 48 | | 32-bit |
| BODE.PRBDDEPTH | 50 | | 8-bit |
| BODE.VAMP | 52 | 64-bit | low 32-bit, signed |
| CAP0.EDGE | 54 | | 8-bit |
| CAP0.EN | 56 | | 8-bit |
| CAP0.EVENT | 58 | | 8-bit |
| CAP0.FILTER | 60 | | 8-bit |
| CAP0.MODE | 62 | | 8-bit |
| CAP0.PLFB | 64 | 64-bit | 64-bit, signed |
| CAP0.PREEDGE | 68 | | 8-bit |
| CAP0.PREFILTER | 70 | | 8-bit |
| CAP0.PRESELECT | 72 | | 8-bit |
| CAP0.STATE | 74 | | 8-bit |
| CAP0.T | 76 | | 32-bit |
| CAP0.TRIGGER | 78 | | 8-bit |
| CAP1.EDGE | 80 | | 8-bit |
| CAP1.EN | 82 | | 8-bit |
| CAP1.EVENT | 84 | | 8-bit |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|----------------|-------------------------|--------|-----------------|
| CAP1.FILTER | 86 | | 8-bit |
| CAP1.MODE | 88 | | 8-bit |
| CAP1.PLFB | 90 | 64-bit | 64-bit, signed |
| CAP1.PREEDGE | 94 | | 8-bit |
| CAP1.PREFILTER | 96 | | 8-bit |
| CAP1.PRESELECT | 98 | | 8-bit |
| CAP1.STATE | 100 | | 8-bit |
| CAP1.T | 102 | | 32-bit |
| CAP1.TRIGGER | 104 | | 8-bit |
| CS.DEC | 106 | 64-bit | 64bit |
| CS.STATE | 110 | | 8-bit |
| CS.TO | 112 | | 32-bit |
| CS.VTHRESH | 114 | 64-bit | low 32-bit word |
| DIN.ROTARY | 116 | | 8-bit |
| DIN.STATES | 118 | | 8-bit |
| DIN1.INV | 120 | | 8-bit |
| DIN1.MODE | 122 | | 16-bit |
| DIN1.PARAM | 124 | 64-bit | 64-bit, signed |
| DIN1.STATE | 128 | | 8-bit |
| DIN2.INV | 130 | | 8-bit |
| DIN2.MODE | 132 | | 16-bit |
| DIN2.PARAM | 134 | 64-bit | 64-bit, signed |
| DIN2.STATE | 138 | | 8-bit |
| DIN3.INV | 140 | | 8-bit |
| DIN3.MODE | 142 | | 16-bit |
| DIN3.PARAM | 144 | 64-bit | 64-bit, signed |
| DIN3.STATE | 148 | | 8-bit |
| DIN4.INV | 150 | | 8-bit |
| DIN4.MODE | 152 | | 16-bit |
| DIN4.PARAM | 154 | 64-bit | 64-bit, signed |
| DIN4.STATE | 158 | | 8-bit |
| DIN5.INV | 160 | | 8-bit |
| DIN5.MODE | 162 | | 16-bit |
| DIN5.PARAM | 164 | 64-bit | 64-bit, signed |
| DIN5.STATE | 168 | | 8-bit |
| DIN6.INV | 170 | | 8-bit |
| DIN6.MODE | 172 | | 16-bit |
| DIN6.PARAM | 174 | 64-bit | 64-bit, signed |
| DIN6.STATE | 178 | | 8-bit |
| DIN7.INV | 180 | | 8-bit |
| DIN7.MODE | 182 | | 16-bit |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|------------------|-------------------------|--------|----------------|
| DIN7.PARAM | 184 | 64-bit | 64-bit, signed |
| DIN7.STATE | 188 | | 8-bit |
| DOU7.CTRL | 190 | | 8-bit |
| DOU7.RELAYMODE | 192 | | 8-bit |
| DOU7.STATES | 194 | | 8-bit |
| DOU1.MODE | 196 | | 8-bit |
| DOU1.PARAM | 198 | 64-bit | 64-bit, signed |
| DOU1.STATE | 202 | | 8-bit |
| DOU1.STATEU | 204 | | 8-bit |
| DOU2.MODE | 206 | | 8-bit |
| DOU2.PARAM | 208 | 64-bit | 64-bit, signed |
| DOU2.STATE | 212 | | 8-bit |
| DOU2.STATEU | 214 | | 8-bit |
| DRV.ACC | 216 | 64-bit | 64-bit |
| DRV.ACTIVE | 220 | | 8-bit |
| DRV.CLRFAULTHIST | 222 | | Command |
| DRV.CLRFAULTS | 224 | | Command |
| DRV.CMDSOURCE | 226 | | 8-bit |
| DRV.DBILIMIT | 228 | | 16-bit |
| DRV.DEC | 230 | 64-bit | 64-bit |
| DRV.DIR | 234 | | 8-bit |
| DRV.DIS | 236 | | Command |
| DRV.DISMODE | 238 | | 8-bit |
| DRV.DISSOURCES | 240 | | 16-bit |
| DRV.DISTO | 242 | | 32-bit |
| DRV.EMUEDIR | 244 | | 8-bit |
| DRV.EMUEMODE | 246 | | 16-bit |
| DRV.EMUEMTURN | 248 | | 32-bit |
| DRV.EMUERES | 250 | | 32-bit |
| DRV.EMUEZOFFSET | 252 | | 16-bit |
| DRV.EN | 254 | | Command |
| DRV.ENDEFAULT | 256 | | 8-bit |
| DRV.HANDWHEEL | 258 | | 32-bit |
| DRV.HWENMODE | 260 | | 8-bit |
| DRV.ICONT | 262 | | 16-bit Signed, |
| DRV.IPEAK | 264 | | 16-bit Signed, |
| DRV.IZERO | 266 | | 16-bit |
| DRV.MOTIONSTAT | 268 | | 32-bit |
| DRV.OPMODE | 270 | | 8-bit |
| DRV.RSTVAR | 272 | | Command |
| DRV.STOP | 274 | | Command |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|-------------------|-------------------------|--------|----------------|
| DRV.TYPE | 276 | | 8-bit |
| DRV.ZERO | 278 | | 8-bit |
| FB1.BISSBITS | 280 | | 8-bit |
| FB1.ENCREC | 282 | | 32-bit |
| FB1.IDENTIFIED | 284 | | 8-bit |
| FB1.INITSIGNED | 286 | | 8-bit, signed |
| FB1.MECHPOS | 288 | | 32-bit |
| FB1.OFFSET | 290 | 64-bit | 64-bit, signed |
| FB1.ORIGIN | 294 | 64-bit | 64-bit |
| FB1.PFIND | 298 | | 8-bit |
| FB1.PFINDCMDU | 300 | | 16-bit |
| FB1.POLES | 302 | | 16-bit |
| FB1.PSCALE | 304 | | 8-bit |
| FB1.RESKTR | 306 | | 16-bit |
| FB1.RESREFPHASE | 308 | | 32-bit, signed |
| FB1.SELECT | 310 | | 8-bit, signed |
| FB1.TRACKINGCAL | 312 | | 8-bit |
| FBUS.PARAM01 | 314 | | 32-bit |
| FBUS.PARAM02 | 316 | | 32-bit |
| FBUS.PARAM03 | 318 | | 32-bit |
| FBUS.PARAM04 | 320 | | 32-bit |
| FBUS.PARAM05 | 322 | | 32-bit |
| FBUS.PARAM06 | 324 | | 32-bit |
| FBUS.PARAM07 | 326 | | 32-bit |
| FBUS.PARAM08 | 328 | | 32-bit |
| FBUS.PARAM09 | 330 | | 32-bit |
| FBUS.PARAM10 | 332 | | 32-bit |
| FBUS.PARAM11 | 334 | | 32-bit |
| FBUS.PARAM12 | 336 | | 32-bit |
| FBUS.PARAM13 | 338 | | 32-bit |
| FBUS.PARAM14 | 340 | | 32-bit |
| FBUS.PARAM15 | 342 | | 32-bit |
| FBUS.PARAM16 | 344 | | 32-bit |
| FBUS.PARAM17 | 346 | | 32-bit |
| FBUS.PARAM18 | 348 | | 32-bit |
| FBUS.PARAM19 | 350 | | 32-bit |
| FBUS.PARAM20 | 352 | | 32-bit |
| FBUS.PLLTHRESH | 354 | | 16-bit |
| FBUS.SAMPLEPERIOD | 356 | | 8-bit |
| FBUS.SYNCACT | 358 | | 32-bit |
| FBUS.SYNCDIST | 360 | | 32-bit |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|-----------------|-------------------------|--------|-----------------|
| FBUS.SYNCWND | 362 | | 32-bit |
| FBUS.TYPE | 364 | | 8-bit |
| GEAR.ACCMAX | 366 | 64-bit | 64-bit |
| GEAR.DECMAX | 370 | 64-bit | 64-bit |
| GEAR.IN | 374 | | 16-bit |
| GEAR.MODE | 376 | | 16-bit |
| GEAR.MOVE | 378 | | Command |
| GEAR.OUT | 380 | | 16-bit, signed |
| GEAR.VMAX | 382 | 64-bit | low 32bit word, |
| HOME.ACC | 384 | 64-bit | 64-bit |
| HOME.AUTOMOVE | 388 | | 8-bit |
| HOME.DEC | 390 | 64-bit | 64-bit |
| HOME.DIR | 394 | | 16-bit |
| HOME.DIST | 396 | 64-bit | 64-bit, signed |
| HOME.FEEDRATE | 400 | | 16-bit |
| HOME.IPEAK | 402 | 64-bit | 64-bit, signed |
| HOME.MODE | 406 | | 16-bit |
| HOME.MOVE | 408 | | Command |
| HOME.P | 410 | 64-bit | 64-bit, signed |
| HOME.PERRTHRESH | 414 | 64-bit | 64-bit, signed |
| HOME.SET | 418 | | Command |
| HOME.V | 420 | 64-bit | Low 32-bit word |
| HWLS.NEGSTATE | 422 | | 8-bit |
| HWLS.POSSTATE | 424 | | 8-bit |
| IL.BUSFF | 426 | | 16-bit, signed |
| IL.CMD | 428 | | 16-bit, signed |
| IL.CMDU | 430 | | 16-bit, signed |
| IL.FB | 432 | | 16-bit, signed |
| IL.FF | 434 | | 16-bit |
| IL.FOLDFTHRESH | 436 | | 16-bit |
| IL.FOLDFTHRESHU | 438 | | 32-bit, signed |
| IL.FOLDWTHRESH | 440 | | 32-bit, signed |
| IL.FRICTION | 442 | | 32-bit |
| IL.IFOLD | 444 | | 32-bit |
| IL.IUFB | 446 | | 16-bit, signed |
| IL.IVFB | 448 | | 16-bit, signed |
| IL.KACFF | 450 | | 32-bit, signed |
| IL.KBUSFF | 452 | | 32-bit |
| IL.KP | 454 | | 16-bit |
| IL.KPDRATIO | 456 | | 32-bit |
| IL.KVFF | 458 | | 32-bit, signed |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|-------------------|-------------------------|--------|----------------|
| IL.LIMITN | 460 | | 16-bit, signed |
| IL.LIMITP | 462 | | 16-bit, signed |
| IL.MFOLDD | 464 | | 32-bit |
| IL.MFOLDR | 466 | | 32-bit |
| IL.MFOLDT | 468 | | 32-bit |
| IL.MIFOLD | 470 | | 32-bit |
| IL.OFFSET | 472 | | 32-bit, signed |
| IL.VCMD | 474 | | 16-bit, signed |
| IL.VUFB | 476 | | 16-bit, signed |
| IL.VVFB | 478 | | 16-bit, signed |
| MOTOR.AUTOSET | 480 | | 8-bit |
| MOTOR.BRAKE | 482 | | 8-bit |
| MOTOR.BRAKERLS | 484 | | 8-bit |
| MOTOR.CTF0 | 486 | | 32-bit |
| MOTOR.ICONT | 488 | | 32-bit |
| MOTOR.IDDATAVALID | 490 | | 8-bit |
| MOTOR.INERTIA | 492 | | 32-bit |
| MOTOR.IPEAK | 494 | | 32-bit |
| MOTOR.KT | 496 | | 32-bit |
| MOTOR.LQLL | 498 | | 32-bit |
| MOTOR.PHASE | 500 | | 16-bit |
| MOTOR.PITCH | 502 | | 32-bit |
| MOTOR.POLES | 504 | | 16-bit |
| MOTOR.R | 506 | | 32-bit |
| MOTOR.RTYPE | 508 | | 8-bit |
| MOTOR.TBRAKEAPP | 510 | | 16-bit |
| MOTOR.TBRAKERLS | 512 | | 16-bit |
| MOTOR.TBRAKETO | 990 | | 16-bit |
| MOTOR.TEMP | 514 | | 32-bit |
| MOTOR.TEMPFAULT | 516 | | 32-bit |
| MOTOR.TEMPWARN | 518 | | 32-bit |
| MOTOR.TYPE | 520 | | 8-bit |
| MOTOR.VMAX | 522 | | 16-bit |
| MOTOR.VOLTMAX | 524 | | 16-bit |
| MOTOR.VOLTMIN | 998 | | 16-bit |
| MOTOR.VOLTRATED | 992 | | 16-bit |
| MOTOR.VRATED | 994 | Yes | 64-bit, signed |
| MT.ACC | 526 | 64-bit | 64-bit |
| MT.CLEAR | 530 | | 16-bit, signed |
| MT.CNTL | 532 | | 32-bit |
| MT.CONTINUE | 534 | | Command |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|---------------|-------------------------|--------|-------------------------|
| MT.DEC | 536 | 64-bit | 64-bit |
| MT.EMERGMT | 540 | | 16-bit, signed |
| MT.LOAD | 542 | | Command |
| MT.MOVE | 544 | | Command 16-bit |
| MT.MTNEXT | 546 | | 8-bit |
| MT.NUM | 548 | | 8-bit |
| MT.P | 550 | 64-bit | 64-bit, signed |
| MT.SET | 554 | | Command 8-bit |
| MT.TNEXT | 556 | | 16-bit |
| MT.TNUM | 558 | | 8-bit |
| MT.TPOSWND | 560 | 64-bit | 64-bit, signed |
| MT.TVELWND | 564 | | 32-bit |
| MT.V | 566 | 64-bit | low 32-bit word |
| MT.VCMD | 568 | 64-bit | low 32-bit word, signed |
| PL.CMD | 570 | 64-bit | 64-bit |
| PL.ERR | 574 | 64-bit | 64-bit |
| PL.ERRMODE | 578 | | 8-bit |
| PL.ERRFTHRESH | 580 | 64-bit | 64-bit |
| PL.ERRWTHRESH | 584 | 64-bit | 64-bit |
| PL.FB | 588 | 64-bit | 64-bit, signed |
| PL.FBSOURCE | 592 | | 8-bit |
| PL.INTINMAX | 594 | 64-bit | 64-bit |
| PL.INTOUTMAX | 598 | 64-bit | 64-bit |
| PL.KI | 602 | | 32-bit |
| PL.KP | 604 | | 32-bit |
| PL.MODP1 | 606 | 64-bit | 64-bit, signed |
| PL.MODP2 | 610 | 64-bit | 64-bit, signed |
| PL.MODPDIR | 614 | | 8-bit |
| PL.MODPEN | 616 | | 8-bit |
| PLS.EN | 618 | | 16-bit |
| PLS.MODE | 620 | | 16-bit |
| PLS.P1 | 622 | 64-bit | 64-bit, signed |
| PLS.P2 | 626 | 64-bit | 64-bit, signed |
| PLS.P3 | 630 | 64-bit | 64-bit, signed |
| PLS.P4 | 634 | 64-bit | 64-bit, signed |
| PLS.P5 | 638 | 64-bit | 64-bit, signed |
| PLS.P6 | 642 | 64-bit | 64-bit, signed |
| PLS.P7 | 646 | 64-bit | 64-bit, signed |
| PLS.P8 | 650 | 64-bit | 64-bit, signed |
| PLS.RESET | 654 | | 16-bit |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|---------------|-------------------------|--------|----------------|
| PLS.STATE | 656 | | 16-bit |
| PLS.T1 | 658 | | 16-bit |
| PLS.T2 | 660 | | 16-bit |
| PLS.T3 | 662 | | 16-bit |
| PLS.T4 | 664 | | 16-bit |
| PLS.T5 | 666 | | 16-bit |
| PLS.T6 | 668 | | 16-bit |
| PLS.T7 | 670 | | 16-bit |
| PLS.T8 | 672 | | 16-bit |
| PLS.UNITS | 674 | | 8-bit |
| PLS.WIDTH1 | 676 | 64-bit | 64-bit, signed |
| PLS.WIDTH2 | 680 | 64-bit | 64-bit, signed |
| PLS.WIDTH3 | 684 | 64-bit | 64-bit, signed |
| PLS.WIDTH4 | 688 | 64-bit | 64-bit, signed |
| PLS.WIDTH5 | 692 | 64-bit | 64-bit, signed |
| PLS.WIDTH6 | 696 | 64-bit | 64-bit, signed |
| PLS.WIDTH7 | 700 | 64-bit | 64-bit, signed |
| PLS.WIDTH8 | 704 | 64-bit | 64-bit, signed |
| REC.ACTIVE | 708 | | 8-bit |
| REC.DONE | 710 | | 8-bit |
| REC.GAP | 712 | | 16-bit |
| REC.NUMPOINTS | 714 | | 16-bit |
| REC.OFF | 716 | | Command |
| REC.STOPTYPE | 718 | | 8-bit |
| REC.TRIG | 720 | | Command |
| REC.TRIGPOS | 722 | | 8-bit |
| REC.TRIGSLOPE | 726 | | |
| REC.TRIGTYPE | 728 | | 8-bit |
| REC.TRIGVAL | 730 | 64-bit | 8-bit |
| REGEN.POWER | 734 | 64-bit | 64-bitSigned, |
| REGEN.REXT | 738 | | 64-bit |
| REGEN.TEXT | 740 | | 8-bit |
| REGEN.TYPE | 742 | | 32-bit |
| REGEN.WATTEXT | 744 | | 8-bitSigned, |
| SM.I1 | 746 | | 16-bit |
| SM.I2 | 748 | | 16-bitSigned, |
| SM.MODE | 750 | | 16-bitSigned, |
| SM.MOVE | 752 | | 16-bit |
| SM.T1 | 754 | | Command |
| SM.T2 | 756 | | 16-bit |
| SM.V1 | 758 | 64-bit | 16-bit |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|----------------|-------------------------|--------|-------------------------|
| SM.V2 | 760 | 64-bit | low 32-bit word, signed |
| STO.STATE | 762 | | low 32-bit word, signed |
| SWLS.EN | 764 | | 8-bit |
| SWLS.LIMIT0 | 766 | 64-bit | 16-bit |
| SWLS.LIMIT1 | 770 | 64-bit | 64-bitSigned, |
| SWLS.STATE | 774 | | 64-bitSigned, |
| UNIT.ACCLINEAR | 776 | | 16-bit |
| UNIT.ACCROTARY | 778 | | 8-bit |
| UNIT.PIN | 780 | | 8-bit |
| UNIT.PLINEAR | 782 | | 32-bit |
| UNIT.POUT | 784 | | 8-bit |
| UNIT.PROTARY | 786 | | 32-bit |
| UNIT.VLINEAR | 788 | | 8-bit |
| UNIT.VROTARY | 790 | | 8-bit |
| VBUS.CALGAIN | 792 | | 8-bit |
| VBUS.OVFTHRESH | 794 | | 32-bit |
| VBUS.OVWTHRESH | 796 | | 16-bit |
| VBUS.RMSLIMIT | 798 | | 16-bit |
| VBUS.UVFTHRESH | 800 | | 8-bit |
| VBUS.UVMODE | 802 | | 16-bit |
| VBUS.UVWTHRESH | 804 | | 8-bit |
| VBUS.VALUE | 806 | | 16-bit |
| VL.ARPF1 | 808 | | 32-bit |
| VL.ARPF2 | 810 | | 32-bit |
| VL.ARPF3 | 812 | | 32-bit |
| VL.ARPF4 | 814 | | 32-bit |
| VL.ARPQ1 | 816 | | 32-bit |
| VL.ARPQ2 | 818 | | 32-bit |
| VL.ARPQ3 | 820 | | 32-bit |
| VL.ARPQ4 | 822 | | 32-bit |
| VL.ARTYPE1 | 824 | | 8-bit |
| VL.ARTYPE2 | 826 | | 8-bit |
| VL.ARTYPE3 | 828 | | 8-bit |
| VL.ARTYPE4 | 830 | | 8-bit |
| VL.ARF1 | 832 | | 32-bit |
| VL.ARF2 | 834 | | 32-bit |
| VL.ARF3 | 836 | | 32-bit |
| VL.ARF4 | 838 | | 32-bit |
| VL.ARFQ1 | 840 | | 32-bit |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|-------------|-------------------------|--------|-------------------------|
| VL.ARZQ2 | 842 | | 32-bit |
| VL.ARZQ3 | 844 | | 32-bit |
| VL.ARZQ4 | 846 | | 32-bit |
| VL.BUSFF | 848 | | 32-bit, signed |
| VL.CMD | 850 | 64-bit | low 32-bit word, signed |
| VL.CMDU | 852 | 64-bit | low 32-bit word, signed |
| VL.ERR | 854 | | 32-bit, signed |
| VL.FB | 856 | 64-bit | low 32-bit word, signed |
| VL.FBFILTER | 858 | 64-bit | low 32-bit word, signed |
| VL.FBSOURCE | 860 | | 8-bit |
| VL.FF | 862 | | 32-bit, signed |
| VL.GENMODE | 864 | | 16-bit |
| VL.KBUSFF | 866 | | 32-bit |
| VL.KI | 868 | | 32-bit |
| VL.KO | 870 | | 32-bit |
| VL.KP | 872 | | 32-bit |
| VL.KVFF | 874 | | 32-bit |
| VL.LIMITN | 876 | 64-bit | low 32-bit word, signed |
| VL.LIMITP | 878 | 64-bit | low 32-bit word |
| VL.LMJR | 880 | | 32-bit |
| VL.MODEL | 882 | | 32-bit, signed |
| VL.OBSBW | 884 | | 32-bit |
| VL.OBSMODE | 886 | | 32-bit |
| VL.THRESH | 888 | 64-bit | low 32-bit word, signed |
| WS.ARM | 890 | | Command |
| WS.DISTMIN | 892 | 64-bit | 64-bit, signed |
| WS.DISTMIN | 896 | 64-bit | 64-bit, signed |
| WS.IMAX | 900 | | 16-bit, signed |
| WS.MODE | 902 | | 8-bit |
| WS.NUMLOOPS | 904 | | 8-bit |
| WS.STATE | 906 | | 8-bit |
| WS.T | 908 | | 16-bit |
| WS.TDELAY1 | 910 | | 16-bit |
| WS.TDELAY2 | 912 | | 16-bit |
| WS.TDELAY3 | 914 | | 16-bit |
| WS.VTHRESH | 916 | 64-bit | low 32-bit word, signed |

| Parameter | Modbus Register Address | 64-bit | Attribute |
|--------------------------|-------------------------|--------|-----------|
| DIN1.FILTER | 918 | | 16-bit |
| DIN2.FILTER | 920 | | 16-bit |
| DIN3.FILTER | 922 | | 16-bit |
| DIN4.FILTER | 924 | | 16-bit |
| DIN5.FILTER | 926 | | 16-bit |
| DIN6.FILTER | 928 | | 16-bit |
| DIN7.FILTER | 930 | | 16-bit |
| FB1.HALLSTATEU | 932 | | |
| FB1.HALLSTATEV | 934 | | |
| FB1.HALLSTATEW | 936 | | |
| DRV.NVSAVE | 938 | | Command |
| MODBUS.DIO | 940 | | 32-bit |
| MODBUS.DRV | 942 | | 32-bit |
| MODBUS.DRVSTAT | 944 | | 32-bit |
| MODBUS.HOME | 946 | | 32-bit |
| MODBUS.MOTOR | 948 | | 32-bit |
| MODBUS.MT | 950 | | 32-bit |
| MODBUS.SM | 952 | | 32-bit |
| MODBUS.FAULT1 | 954 | | 16-bit |
| MODBUS.FAULT2 | 956 | | 16-bit |
| MODBUS.FAULT3 | 958 | | 16-bit |
| MODBUS.FAULT4 | 960 | | 16-bit |
| MODBUS.FAULT5 | 962 | | 16-bit |
| MODBUS.FAULT6 | 964 | | 16-bit |
| MODBUS.FAULT7 | 966 | | 16-bit |
| MODBUS.FAULT8 | 968 | | 16-bit |
| MODBUS.FAULT9 | 970 | | 16-bit |
| MODBUS.FAULT10 | 972 | | 16-bit |
| MODBUS.PIN | 974 | | 32-bit |
| MODBUS.POUT | 976 | | 32-bit |
| MODBUS.PSCALE | 978 | | 16-bit |
| MODBUS.UNITLABEL | 980 | | |
| MOTOR.HFPHASEREAD | 982 | | 16-bit |
| FB2.ENCRES | 984 | | 32-bit |
| FB2.MODE | 986 | | 16-bit |
| "FB2.SOURCE" (=> p. 440) | 988 | | 16-bit |

20.11 Modbus 64-bit Parameters to 32-bit Mapping

| Parameter | Address | 64-bit | Attributes |
|--------------------|---------|--------|-------------------------|
| AIN.PSCALE_32 | 2000 | Yes | low 32-bit word |
| AOUT.PSCALE_32 | 2002 | Yes | low 32-bit word |
| AOUT.VALUE_32 | 2004 | Yes | low 32-bit word, signed |
| AOUT.VALUEU_32 | 2006 | Yes | low 32-bit word, signed |
| CAP0.PLFB_32 | 2008 | Yes | low 32-bit word, signed |
| CAP1.PLFB_32 | 2010 | Yes | low 32-bit word, signed |
| CS.DEC_32 | 2012 | Yes | low 32-bit word |
| DIN1.PARAM_32 | 2014 | Yes | low 32-bit word, signed |
| DIN2.PARAM_32 | 2016 | Yes | low 32-bit word, signed |
| DIN3.PARAM_32 | 2018 | Yes | low 32-bit word, signed |
| DIN4.PARAM_32 | 2020 | Yes | low 32-bit word, signed |
| DIN5.PARAM_32 | 2022 | Yes | low 32-bit word, signed |
| DIN6.PARAM_32 | 2024 | Yes | low 32-bit word, signed |
| DIN7.PARAM_32 | 2026 | Yes | low 32-bit word, signed |
| DOUT1.PARAM_32 | 2028 | Yes | low 32-bit word, signed |
| DOUT2.PARAM_32 | 2030 | Yes | low 32-bit word, signed |
| DRV.ACC_32 | 2032 | Yes | low 32-bit word |
| DRV.DEC_32 | 2034 | Yes | low 32-bit word |
| FB1.OFFSET_32 | 2036 | Yes | low 32-bit word, signed |
| FB1.ORIGIN_32 | 2038 | Yes | low 32-bit word |
| GEAR.ACCMAX_32 | 2040 | Yes | low 32-bit word |
| GEAR.DECMAX_32 | 2042 | Yes | low 32-bit word |
| HOME.ACC_32 | 2044 | Yes | low 32-bit word |
| HOME.DEC_32 | 2046 | Yes | low 32-bit word |
| HOME.DIST_32 | 2048 | Yes | low 32-bit word, signed |
| HOME.IPEAK_32 | 2050 | Yes | low 32-bit word, signed |
| HOME.P_32 | 2052 | Yes | low 32-bit word, signed |
| HOME.PERRTHRESH_32 | 2054 | Yes | low 32-bit word, signed |
| MOTOR.VRATED_32 | 2126 | Yes | low 32-bit word, signed |
| MT.ACC_32 | 2056 | Yes | low 32-bit word |
| MT.DEC_32 | 2058 | Yes | low 32-bit word |
| MT.P_32 | 2060 | Yes | low 32-bit word, signed |
| MT.TPOSWND_32 | 2062 | Yes | low 32-bit word, signed |
| PL.CMD_32 | 2064 | Yes | low 32-bit word |
| PL.ERR_32 | 2066 | Yes | low 32-bit word |
| PL.ERRFTHRESH_32 | 2068 | Yes | low 32-bit word |
| PL.ERRWTHRESH_32 | 2070 | Yes | low 32-bit word |
| PL.FB_32 | 2072 | Yes | low 32-bit word, signed |
| PL.INTINMAX_32 | 2074 | Yes | low 32-bit word |
| PL.INTOUTMAX_32 | 2076 | Yes | low 32-bit word |
| PL.MODP1_32 | 2078 | Yes | low 32-bit word, signed |
| PL.MODP2_32 | 2080 | Yes | low 32-bit word, signed |
| PLS.P1_32 | 2082 | Yes | low 32-bit word, signed |

| Parameter | Address | 64-bit | Attributes |
|-----------------|---------|--------|-------------------------|
| PLS.P2_32 | 2084 | Yes | low 32-bit word, signed |
| PLS.P3_32 | 2086 | Yes | low 32-bit word, signed |
| PLS.P4_32 | 2088 | Yes | low 32-bit word, signed |
| PLS.P5_32 | 2090 | Yes | low 32-bit word, signed |
| PLS.P6_32 | 2092 | Yes | low 32-bit word, signed |
| PLS.P7_32 | 2094 | Yes | low 32-bit word, signed |
| PLS.P8_32 | 2096 | Yes | low 32-bit word, signed |
| PLS.WIDTH1_32 | 2098 | Yes | low 32-bit word, signed |
| PLS.WIDTH2_32 | 2100 | Yes | low 32-bit word, signed |
| PLS.WIDTH3_32 | 2102 | Yes | low 32-bit word, signed |
| PLS.WIDTH4_32 | 2104 | Yes | low 32-bit word, signed |
| PLS.WIDTH5_32 | 2106 | Yes | low 32-bit word, signed |
| PLS.WIDTH6_32 | 2108 | Yes | low 32-bit word, signed |
| PLS.WIDTH7_32 | 2110 | Yes | low 32-bit word, signed |
| PLS.WIDTH8_32 | 2112 | Yes | low 32-bit word, signed |
| REC.TRIGVAL_32 | 2114 | Yes | low 32-bit word, signed |
| REGEN.POWER_32 | 2116 | Yes | low 32-bit word |
| SWLS.LIMIT0_32 | 2118 | Yes | low 32-bit word, signed |
| SWLS.LIMIT1_32 | 2120 | Yes | low 32-bit word, signed |
| WS.DISTMAX_32 | 2122 | Yes | low 32-bit word, signed |
| WS.DISTMIN_32 | 2124 | Yes | low 32-bit word, signed |
| MOTOR.VRATED_32 | 2126 | Yes | low 32-bit word, signed |

This page intentionally left blank.

21 Ethercat Manual

AKD™

EtherCAT Communication



Edition: Revision C, April 2011

Valid for Hardware Revision C

Part Number 903-200005-00

Original Documentation

EtherCAT®

Keep all manuals as a product component during the life span of the product. Pass all manuals to future users and owners of the product.

KOLLMORGEN®

Record of Document Revisions:

| Revision | Remarks |
|------------|---|
| -, 11/2009 | Beta launch version |
| -, 12/2009 | Minor formatting changes |
| A, 07/2010 | FBUS.PARAM04 added, part number added, page format, release information |
| B, 10/2010 | Setup for KAS added |
| C, 01/2011 | HW Rev. C |
| D, 04/2011 | WoE, corrections |

Hardware Revision (HR)

| Hardware Revision | Firmware | WorkBench | Remarks |
|-------------------|------------------|---------------|----------------|
| A | M_01-03-zz-zzz | 1.3.0.zzzzz | Start revision |
| C | ≥ M_01-03-00-011 | ≥ 1.3.0.zzzzz | STO certified |
| | | | |

Ethernet/IP is a registered trademark of ODVA, Inc.

Ethernet/IP Communication Stack: copyright (c) 2009, Rockwell Automation

EnDat is a registered trademark of Dr. Johannes Heidenhain GmbH

EtherCAT is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH”

HIPERFACE is a registered trademark of Max Stegmann GmbH

WINDOWS is a registered trademark of Microsoft Corporation

AKD is a registered trademark of Kollmorgen Corporation

Current patents:

US Patent 5,646,496 (used in control card R/D and 1 Vp-p feedback interface)

US Patent 5,162,798 (used in control card R/D)

US Patent 6,118,241 (used in control card simple dynamic braking)

Technical changes which improve the performance of the device may be made without prior notice!

Printed in the United States of America

This document is the intellectual property of Kollmorgen. All rights reserved. No part of this work may be reproduced in any form (by photocopying, microfilm or any other method) or stored, processed, copied or distributed by electronic means without the written permission of Kollmorgen.

Table of Contents

| | | |
|----------|---|-----------|
| 1 | About this User Guide | 29 |
| 1.1 | About this User Guide | 29 |
| 1.2 | Abbreviations | 29 |
| 2 | AKD Models | 31 |
| 2.1 | CC Drive Models | 31 |
| 3 | Basic Drive Setup | 33 |
| 3.1 | Basic Drive Setup | 34 |
| 3.2 | Display Codes | 34 |
| 3.3 | AKD Setup Wizard | 35 |
| 4 | Connecting the Drive | 37 |
| 4.1 | Connected and Disconnected States | 38 |
| 4.2 | Disconnected | 38 |
| 4.3 | Device Not Shown | 38 |
| 4.4 | Find and Enter IP Address | 39 |
| 4.5 | Check Communications | 40 |
| 4.6 | Connect To Another Drive | 40 |
| 5 | Communicating with the Drive | 41 |
| 5.0.1 | Overview | 42 |
| 5.0.2 | Identifying the Drive IP Address | 42 |
| 5.0.2.1 | Automatic (Dynamic) IP Addressing | 42 |
| 5.0.2.2 | Static IP Addressing | 42 |
| 5.1 | Check Communications | 43 |
| 5.2 | Troubleshooting Connection and Communication Problems | 45 |
| 5.2.1 | No Drives are Shown | 46 |
| 5.2.2 | Device Not Shown | 46 |
| 5.2.3 | Find and Enter IP Address | 46 |
| 5.3 | TCP/IP Communication Protocols | 48 |
| 5.4 | IP Address | 48 |
| 5.4.1 | Port Number | 48 |
| 5.4.2 | MAC Address | 48 |
| 6 | Using WorkBench | 49 |
| 6.1 | Welcome Screen | 49 |
| 6.1.1 | Online | 49 |
| 6.1.2 | Offline | 50 |
| 6.2 | AKD Overview | 50 |
| 6.3 | Online and Offline | 51 |
| 6.3.1 | Online Drive | 51 |
| 6.3.2 | Offline Drive | 51 |
| 6.3.3 | Switching Between Online and Offline | 52 |
| 6.4 | Watch | 52 |
| 6.5 | Settings | 52 |
| 6.5.1 | Navigation Tree | 52 |

| | |
|--|-----------|
| 6.5.2 Settings View..... | 52 |
| 7 Configuring Drive Power..... | 55 |
| 7.1 Power..... | 56 |
| 7.1.1 Drive Setup for Power and Bus..... | 56 |
| 7.1.2 Direct DC Mains Operation..... | 56 |
| 7.2 Regeneration..... | 58 |
| 7.2.1 Overview..... | 58 |
| 7.2.2 Regen Resistor Options..... | 58 |
| 7.2.3 Calculating Motor Peak Energy and Regen Resistor Size..... | 58 |
| 7.2.4 Selecting a Compatible Regen Resistor..... | 61 |
| 7.2.5 Configuring Regen Parameter Values..... | 61 |
| 7.2.6 Related Parameters..... | 62 |
| 8 Configuring Motor Settings..... | 63 |
| 8.1 Motor..... | 64 |
| 8.1.1 Overview..... | 64 |
| 8.1.2 Motor Setup..... | 64 |
| 8.1.3 Using the Motor View..... | 64 |
| 8.1.4 Selecting a Motor..... | 65 |
| 8.1.4.1 Configuring Custom Motors..... | 66 |
| 8.1.4.2 Validating Motor Parameters..... | 66 |
| 8.2 Feedback 1..... | 66 |
| 8.2.1 Overview..... | 67 |
| 8.3 Using Feedback Options..... | 67 |
| 8.3.1 Auto..... | 67 |
| 8.3.2 Incremental Encoder..... | 67 |
| 8.3.3 Sine Encoder..... | 67 |
| 8.3.4 Endat 2.1, Endat 2.2..... | 67 |
| 8.3.5 BiSS..... | 67 |
| 8.3.6 Hiperface..... | 67 |
| 8.3.7 Resolver..... | 68 |
| 8.3.8 SFD..... | 68 |
| 8.3.9 Using Wake and Shake (WS)..... | 68 |
| 8.3.9.1 Overview..... | 68 |
| 8.3.9.2 Configuring WS..... | 68 |
| Wake and Shake, More View..... | 69 |
| Special Cases for WS..... | 70 |
| Operation with Motor Brake..... | 70 |
| End of Travel Limits..... | 70 |
| Large Load Inertia or High Friction System..... | 70 |
| 8.3.9.3 Using WS: Advanced..... | 70 |
| Maximizing WS Reliability..... | 71 |
| 8.3.9.4 Troubleshooting WS..... | 72 |

| | |
|--|----|
| 8.3.9.5 Related Parameters..... | 73 |
| 8.3.10 Commutation Alignment..... | 73 |
| 8.4 Feedback 2..... | 73 |
| 8.4.1 Encoder Emulation..... | 74 |
| 8.4.1.1 Overview..... | 74 |
| 8.4.1.2 Using Encoder Emulation..... | 74 |
| 8.4.1.3 Function Settings..... | 74 |
| Mode 0: Disable X9..... | 74 |
| Output Modes 1 and 2..... | 74 |
| Input Modes 3, 4, and 5..... | 75 |
| 8.4.1.4 Resolution..... | 76 |
| 8.4.1.5 Related Parameters and Commands..... | 76 |
| 8.5 Non-Plug and Play Feedback Devices..... | 76 |
| 8.6 Parameters..... | 76 |
| 8.7 Calculations..... | 76 |
| 8.7.1 Current Loop..... | 77 |
| 8.7.2 Velocity Loop..... | 77 |
| 8.7.3 Position Loop..... | 77 |
| 8.7.4 Slider Tuning..... | 77 |
| 8.7.4.1 Input - Motor Data..... | 77 |
| 8.7.4.2 Constants..... | 77 |
| 8.7.4.3 Output - Control Loop Gains..... | 77 |
| 8.8 Foldback..... | 77 |
| 8.8.1 Drive Foldback..... | 78 |
| 8.8.2 Setting up motor foldback..... | 78 |
| 8.8.3 Setting Fault and Warning Levels..... | 78 |
| 8.8.4 Motor Peak Current Time..... | 79 |
| 8.8.5 Motor Foldback Ramp..... | 79 |
| 8.8.6 Motor Recovery..... | 80 |
| 8.9 Overall Foldback..... | 80 |
| 8.10 Brake..... | 80 |
| 8.11 Using Position Capture..... | 81 |
| 8.11.1 Overview..... | 81 |
| 8.11.2 Configuring Position Capture..... | 81 |
| 8.11.2.1 Setting the Capture Source (CAP0.TRIGGER)..... | 81 |
| 8.11.2.2 Setting the Capture Mode (CAP0.MODE)..... | 81 |
| 8.11.2.3 Arming and Retrieving the Capture Value (CAP0.EN and CAP0.T)..... | 82 |
| 8.11.2.4 Setting the Capture Edge (CAP0.EDGE)..... | 82 |
| 8.11.2.5 Setting the Pre-Condition Event: (CAP0.EVENT)..... | 82 |
| 8.11.2.6 Setting up a Pre-Condition for complex capture..... | 82 |
| Setting the Precondition Edge: (Terminal Command: CAP0.PREEDGE)..... | 82 |
| Setting the Pre-Condition Select: (Terminal Command: CAP0.PRESELECT)..... | 82 |
| 8.11.3 Related Parameters..... | 82 |

| | | |
|-----------|---|-----------|
| 9 | Selecting Units for Your Application | 83 |
| 9.1 | Selecting and Saving Units | 84 |
| 9.2 | Units Example | 84 |
| 9.2.1 | Related Parameters | 86 |
| 10 | Configuring General Drive Settings | 87 |
| 10.1 | Digital Inputs and Outputs | 88 |
| 10.1.1 | Overview | 88 |
| 10.1.2 | Using Digital I/O | 88 |
| 10.1.3 | Digital Inputs | 88 |
| 10.1.4 | Digital Outputs | 94 |
| 10.2 | Command Buffer | 96 |
| 10.2.1 | Overview | 96 |
| 10.2.2 | Editing the Command Buffers | 97 |
| 10.2.3 | Behavior of the Command Buffer | 98 |
| 10.2.4 | Delays for the Buffer | 99 |
| 10.3 | Digital Inputs (X7/X8) | 99 |
| 10.3.1 | Digital Inputs 1 and 2 | 102 |
| 10.3.2 | Digital Inputs 3 to 7 | 102 |
| 10.3.3 | Digital Input 8 (ENABLE) | 102 |
| 10.4 | Analog Input | 103 |
| 10.4.1 | Analog Input | 103 |
| 10.4.2 | Related Parameters and Commands | 103 |
| 10.5 | Analog Output | 103 |
| 10.6 | I/O Connection | 104 |
| 10.6.1 | I/O Connectors (X7 and X8) | 105 |
| 10.7 | Electronic Gearing | 106 |
| 10.7.1 | Overview | 106 |
| 10.7.2 | Limits | 106 |
| 10.7.3 | Determining Maximum Cable Length | 107 |
| 10.7.4 | Related Parameters: | 108 |
| 10.7.5 | Limits | 109 |
| 10.8 | Programmable Limit Switch | 109 |
| 10.8.1 | Overview | 109 |
| 10.8.2 | Using Programmable Limit Switches | 109 |
| 10.8.3 | Single Shot Mode | 111 |
| 10.8.4 | Related Parameters and Commands | 111 |
| 10.9 | Enable/Disable | 111 |
| 10.9.1 | Enable Modes | 111 |
| 10.9.1.1 | Hardware Enable Mode | 111 |
| 10.9.1.2 | Software Enable Default | 112 |
| 10.9.2 | Disable Modes | 112 |
| 10.9.3 | Drive Status | 112 |
| 10.9.4 | Controlled Stop | 113 |
| 10.9.5 | More/Less Button | 113 |
| 10.10 | Controlled Stop | 115 |

| | | |
|-----------|--|------------|
| 10.10.0.1 | Related Parameters and Commands..... | 116 |
| 10.10.1 | Related topics:..... | 117 |
| 10.11 | Dynamic Braking..... | 117 |
| 10.11.1 | Drive Regeneration..... | 117 |
| 10.11.1.1 | AKD-x00306 to AKD-x00606..... | 117 |
| 10.11.1.2 | AKD-x01206 to AKD-x02406 and AKD-xzzz07..... | 117 |
| 10.12 | Emergency Stop..... | 118 |
| 10.13 | Safe Torque Off (STO)..... | 118 |
| 10.14 | Under Voltage Fault Behavior..... | 118 |
| 11 | Using Command Source and Operating Modes..... | 119 |
| 11.1 | Overview..... | 120 |
| 11.2 | Using Command Source and Operation Modes..... | 120 |
| 11.2.1 | Command Source..... | 120 |
| 11.2.1.1 | Service..... | 120 |
| 11.2.1.2 | Fieldbus..... | 120 |
| 11.2.1.3 | Electronic Gearing..... | 120 |
| 11.2.1.4 | Analog..... | 120 |
| 11.2.1.5 | Operation Mode..... | 120 |
| 11.2.2 | Related Parameters..... | 120 |
| 11.3 | Current Loop..... | 121 |
| 11.3.1 | Overview..... | 121 |
| 11.3.2 | Current Loop Gain..... | 121 |
| 11.3.3 | Current Loop Gain Scheduling..... | 122 |
| 11.3.3.1 | Using the Gain Scheduling View in WorkBench..... | 122 |
| 11.3.3.2 | Using the Terminal View for Gains Scheduling..... | 123 |
| 11.4 | Velocity Loop..... | 124 |
| 11.4.1 | Overview..... | 124 |
| 11.4.2 | Tabs in the Velocity Loop View..... | 124 |
| 11.4.3 | Velocity Loop Default Settings and Changes..... | 125 |
| 11.4.3.1 | Velocity Loop Changes Based on Slider Tuning..... | 125 |
| 11.4.3.2 | Velocity Loop Changes Based on PST..... | 126 |
| 11.5 | Position Loop..... | 126 |
| 11.5.1 | Overview..... | 126 |
| 11.5.2 | Tabs in the Position Loop View..... | 126 |
| 11.5.3 | Position Loop Default Behavior and Changes..... | 127 |
| 11.5.3.1 | Position Loop Changes Based on Slider Tuning..... | 127 |
| 11.5.3.2 | Position Loop Changes Based on PST..... | 127 |
| 11.5.4 | Modulo Position..... | 127 |
| 11.5.4.1 | Setting up the modulo axis in WorkBench..... | 128 |
| 11.5.4.2 | Setting up the modulo axis from the Terminal..... | 128 |
| 11.5.4.3 | Parameters affected by the modulo axis..... | 128 |
| 11.5.4.4 | Drive functions affected by modulo axis..... | 129 |
| | Software limit switch..... | 129 |
| | Programmable limit switch..... | 129 |
| | Digital output Modes 5 and 6..... | 129 |

| | |
|--|------------|
| Motion tasking to absolute target positions..... | 129 |
| 11.5.4.5 Using the modulo position feature with multiturn encoders..... | 129 |
| 12 Creating Motion..... | 133 |
| 12.1 Homing..... | 134 |
| 12.1.1 Overview..... | 134 |
| 12.1.2 Using Homing..... | 134 |
| 12.1.2.1 Home Default Window..... | 134 |
| Mode Selection:..... | 135 |
| Settings:..... | 135 |
| Controls:..... | 136 |
| 12.1.2.2 Selecting and Using Homing Modes..... | 136 |
| Homing Mode 0: Home Using Current Position..... | 136 |
| Homing Mode 1: Find Limit Input..... | 136 |
| Homing Mode 2: Find Input Limit then Find Zero Angle..... | 137 |
| Homing Mode 3: Find Input Limit then Find Index..... | 138 |
| Homing Mode 4: Find Home Input..... | 139 |
| Homing Mode 5: Find Home Input then Find Zero Angle..... | 139 |
| Homing Mode 6: Find Home Input then Find Index..... | 140 |
| Homing Mode 7: Find Zero Angle..... | 141 |
| Homing mode 8: Move Until Position Error Exceeded..... | 141 |
| Homing Mode 9: Move Until Position Error Exceeded then Find Zero Angle..... | 141 |
| 12.1.2.3 Homing Mode 10: Move Until Position Error Exceeded then Find Index..... | 142 |
| Homing Mode 11: Find Index Signal..... | 143 |
| 12.1.3 Using Homing: Advanced..... | 144 |
| 12.1.3.1 Related Parameters and Commands..... | 144 |
| 12.1.3.2 Related Topics..... | 144 |
| 12.2 Motion Tasks..... | 144 |
| 12.2.1 Overview..... | 144 |
| 12.2.2 Motion Task Input Table..... | 144 |
| 12.2.3 Using Motion Tasks..... | 145 |
| 12.2.3.1 Motion Profiles..... | 147 |
| 12.2.3.2 Motion Types..... | 147 |
| Absolute motion task..... | 148 |
| Motion task relative to command position (PL.CMD)..... | 148 |
| Motion task relative to previous target position..... | 148 |
| 12.2.4 Using Motion Tasks: Advanced..... | 148 |
| 12.2.4.1 Joining multiple tasks..... | 148 |
| 12.2.4.2 Start Conditions..... | 149 |

| | |
|---|------------|
| 12.2.4.3 Blending..... | 149 |
| 12.3 Registration Moves (Motion Task Blending)..... | 149 |
| 12.3.1 Configuring Registration Moves in WorkBench..... | 150 |
| 12.3.2 Configuring Registration Moves from the Terminal View..... | 150 |
| 12.4 Service Motion..... | 152 |
| 12.5 Jog Move..... | 153 |
| 12.6 Drive Motion Status..... | 153 |
| 13 Saving Your Drive Configuration..... | 155 |
| 13.1 Save Options..... | 156 |
| 13.2 Save On Exit..... | 156 |
| 13.3 Save On Disconnect..... | 157 |
| 13.4 Save On Firmware Download..... | 158 |
| 14 Tuning Your System..... | 159 |
| 14.1 Slider Tuning..... | 160 |
| 14.1.0.1 Gentle, Medium, and Stiff..... | 160 |
| 14.1.0.2 The Slider..... | 160 |
| 14.1.0.3 Inertia Ratio..... | 160 |
| 14.2 Load Inertia..... | 160 |
| 14.3 Using the Performance Servo Tuner..... | 160 |
| 14.3.1 Overview..... | 160 |
| 14.3.2 Using the PST..... | 161 |
| 14.3.2.1 Saving and Emailing Bode Plots..... | 162 |
| 14.3.2.2 Importing a Frequency Response..... | 163 |
| 14.3.3 Measurement Options..... | 163 |
| 14.3.3.1 Using Manual Excitation Levels..... | 163 |
| 14.3.3.2 Taking a Bode Measurement without the PST..... | 164 |
| 14.3.4 Using the Performance Servo Tuner: Advanced..... | 164 |
| 14.3.4.1 Typical Cases for Advanced PST Use..... | 165 |
| Tuning Systems with Low-Frequency Resonances..... | 165 |
| Tuning Systems with High-Frequency Resonances..... | 166 |
| Tuning systems with noisy frequency responses..... | 167 |
| 14.3.4.2 PST Options..... | 169 |
| Enable BiQuad 1 Tuner..... | 170 |
| Enable BiQuad 2 Tuner..... | 170 |
| Biquad Type..... | 170 |
| Tune Acceleration FF..... | 170 |
| Enable Stability Motion..... | 170 |
| Phase and Gain Margins..... | 170 |
| Tune Velocity Integral..... | 170 |
| Enable Low Pass Search..... | 170 |
| 14.3.4.3 Measurement Options..... | 170 |

| | |
|--|------------|
| Current Amplitude..... | 171 |
| Velocity Amplitude..... | 171 |
| Smooth %..... | 171 |
| Measurement..... | 174 |
| Injection Point..... | 174 |
| Excitation Type..... | 175 |
| FFT Points..... | 175 |
| Excite Gap..... | 175 |
| Number Points..... | 175 |
| Velocity Max..... | 176 |
| 14.3.4.4 Plot Options..... | 177 |
| Coherence..... | 177 |
| 14.3.4.5 Cursors..... | 177 |
| 14.3.4.6 Resizing Bode Plots..... | 179 |
| Reading and Understanding the Bode Plot..... | 180 |
| Using the Performance Servo Tuner to Manually Tune Systems..... | 183 |
| Using the Tuning Simulation..... | 184 |
| Using the Performance Servo Tuner Manual Tuning Interface..... | 184 |
| Simulating Modified Loop Gains with the Performance Servo Tuner..... | 185 |
| Simulating Filters with the Performance Servo Tuner..... | 188 |
| Using Filters to Reduce Noise..... | 189 |
| 14.4 Tuning Guide..... | 189 |
| 14.4.1 Overview..... | 189 |
| 14.4.2 Determining Tuning Criteria..... | 190 |
| 14.4.3 Before You Tune..... | 190 |
| 14.4.4 Closed Loop Tuning Methods..... | 190 |
| 14.4.4.1 Tuning the Velocity Loop..... | 191 |
| 14.4.4.2 Tuning the Position Loop..... | 193 |
| 14.4.5 Torque Feedforward Tuning Methods..... | 193 |
| 14.4.5.1 Shape Based Feedforward Tuning..... | 193 |
| 14.4.6 Using Anti-Resonance Filters..... | 194 |
| 14.4.6.1 Biquad Calculations..... | 202 |
| 14.4.6.2 Common Uses Of Anti Resonance Filters..... | 204 |
| 15 Scope..... | 205 |
| 15.1 Overview..... | 205 |
| 15.2 Using the Scope..... | 205 |
| 15.2.1 Scope Channels Tab..... | 205 |
| 15.2.1.1 Source Column..... | 205 |

| | | |
|-----------|--|------------|
| 15.2.1.2 | Color Column..... | 206 |
| 15.2.1.3 | Hide Column..... | 206 |
| 15.2.1.4 | Y-Axis Column..... | 206 |
| 15.2.1.5 | Filter and Filter Frequency Column..... | 206 |
| 15.2.2 | Scope Time-base and Trigger Tab..... | 206 |
| 15.2.2.1 | Scope Time-base and Trigger, More View..... | 207 |
| 15.2.2.2 | Trigger Type..... | 208 |
| 15.2.2.3 | Trigger Position..... | 208 |
| 15.2.2.4 | Trigger Value..... | 209 |
| 15.2.2.5 | Effects of Recorder Gap..... | 210 |
| 15.2.2.6 | Trigger Slope..... | 211 |
| 15.3 | Scope Settings..... | 211 |
| 15.3.1 | Load a setting (preset) to Scope screen..... | 212 |
| 15.3.2 | Create a new preset..... | 212 |
| 15.3.3 | Save or delete preset..... | 212 |
| 15.3.4 | Import preset..... | 212 |
| 15.3.5 | Export preset..... | 213 |
| 15.3.6 | Scope axis scaling and zooming..... | 214 |
| 15.3.7 | Manual range per axis..... | 215 |
| 15.3.8 | Unit display on Y axis..... | 215 |
| 16 | Using Parameters and the Terminal Screen..... | 217 |
| 16.1 | Terminal..... | 218 |
| 16.1.1 | Overview..... | 218 |
| 16.1.2 | Using the Terminal..... | 218 |
| 16.1.3 | Macros..... | 219 |
| 16.1.3.1 | Creating a Macro from Terminal commands..... | 219 |
| 16.1.3.2 | Macro Editor..... | 220 |
| 16.2 | Viewing Parameters..... | 221 |
| 16.3 | Parameter List..... | 221 |
| 16.4 | Parameter Load/Save..... | 221 |
| 16.5 | Parameter Comparer..... | 221 |
| 16.5.1 | Reference Parameter Selection..... | 222 |
| 16.5.2 | Target Parameter Selection..... | 222 |
| 16.5.3 | Display the comparison..... | 223 |
| 16.5.4 | Motion Task Comparison..... | 224 |
| | Summary of Parameters and Commands..... | 225 |
| 17 | Faults and Warnings..... | 239 |
| 17.1 | Fault and Warning Messages..... | 240 |
| 17.2 | Clearing Faults..... | 249 |
| 17.3 | Parameter and Command Error Messages..... | 250 |
| 17.4 | CANopen Emergency Messages and Error Codes..... | 253 |
| 17.5 | Unknown Fault..... | 256 |
| 17.5.0.1 | Remedies..... | 256 |
| 18 | Troubleshooting the AKD..... | 257 |
| 19 | Firmware and Firmware Updates..... | 258 |

| | | |
|---------------------------------|--|------------|
| 19.1 | Downloading Firmware..... | 259 |
| 19.2 | Firmware Compatibility..... | 259 |
| 19.3 | Invalid Firmware..... | 259 |
| 19.4 | Forcing the drive into firmware download mode..... | 260 |
| 19.5 | Forced download of AKD firmware..... | 260 |
| Connection Diagrams..... | | 263 |
| 19.6 | Connection Diagram, AKD-x00306 to x00606..... | 264 |
| 19.7 | Connection Diagram, AKD-x01206..... | 265 |
| 19.8 | Connection Diagram, AKD-x02406 and AKD-xzzz07..... | 266 |
| 19.9 | 24 V Auxiliary Supply (X1)..... | 267 |
| 19.10 | Motor Connection..... | 268 |
| 19.11 | External Regen Resistor (X3)..... | 269 |
| 19.12 | DC Bus Link (X3)..... | 270 |
| 19.13 | Mains Supply Connection (X3, X4)..... | 271 |
| 19.13.1 | Three Phase connection (all AKD types)..... | 271 |
| 19.13.2 | Single phase connection (AKD-xzzz06 only)..... | 272 |
| 19.14 | I/O Connection..... | 273 |
| 19.14.1 | I/O Connectors (X7 and X8)..... | 273 |
| 19.15 | Analog Output (X8)..... | 274 |
| 19.16 | Analog Input (X8)..... | 275 |
| 19.17 | Command encoder signal connection..... | 276 |
| 19.17.1 | Incremental encoder input 5 V (X9)..... | 276 |
| 19.17.2 | Incremental encoder input 24 V (X7)..... | 276 |
| 19.17.3 | Emulated Encoder Output (EEO) - A quad B (X9)..... | 277 |
| 19.18 | Pulse / Direction signal connection..... | 278 |
| 19.18.1 | Pulse / Direction input 5 V (X9)..... | 278 |
| 19.18.2 | Pulse / Direction input 24 V (X7)..... | 278 |
| 19.19 | Up / Down signal connection..... | 278 |
| 19.19.1 | Up / Down input 5 V (X9)..... | 278 |
| 19.19.2 | Up / Down input 24 V (X7)..... | 279 |
| 19.20 | Feedback Connector (X10)..... | 279 |
| Block Diagrams..... | | 281 |
| 19.21 | Block Diagram for Current Loop..... | 282 |
| 19.22 | Block Diagram for Position/Velocity Loop..... | 282 |
| Appendix A..... | | 1 |
| 20 | Modbus..... | 3 |
| 20.1 | Overview..... | 3 |
| 20.2 | Installation and Setup..... | 3 |
| 20.3 | Overview of Messaging..... | 3 |
| 20.4 | Supported Functions..... | 3 |
| 20.5 | Read Holding Registers (0x03)..... | 3 |
| 20.6 | Write Multiple Registers (0x10)..... | 4 |
| 20.7 | Exception Response Codes..... | 5 |
| 20.8 | Modbus..... | 6 |
| 20.8.1 | Overview..... | 6 |

| | | |
|-----------|--|-----------|
| 20.8.2 | Modbus Installation and Setup..... | 6 |
| 20.8.3 | Overview of Messaging..... | 6 |
| 20.8.4 | Supported Functions..... | 6 |
| 20.8.5 | Read Holding Registers (0x03)..... | 6 |
| 20.8.6 | Write Multiple Registers (0x10)..... | 8 |
| 20.8.7 | Exception Response Codes..... | 8 |
| 20.9 | Modbus Dynamic Mapping..... | 9 |
| 20.9.1 | Configuring Dynamic Mapping..... | 9 |
| 20.9.2 | Saving and Resetting Dynamic Mapping..... | 10 |
| 20.9.3 | Scaling parameters..... | 10 |
| 20.9.4 | Modbus specific registers (Parameters)..... | 11 |
| 20.9.5 | 32-bit versus 16-bit Values..... | 11 |
| 20.9.6 | Mapping of 64-bit Parameters to 32-bit Parameters..... | 11 |
| 20.9.7 | Fault Registers..... | 12 |
| 20.9.8 | Mapping Table..... | 12 |
| 20.10 | Modbus Parameter Table..... | 13 |
| 20.11 | Modbus 64-bit Parameters to 32-bit Mapping..... | 23 |
| 21 | Ethercat Manual..... | 27 |
| 21.1 | General..... | 58 |
| 21.1.1 | About this Manual..... | 59 |
| 21.1.2 | Target Group..... | 59 |
| 21.1.3 | Abbreviations Used..... | 60 |
| 21.2 | Safety..... | 61 |
| 21.2.1 | Safety Instructions..... | 62 |
| 21.2.2 | Use as directed..... | 62 |
| 21.2.3 | Prohibited use..... | 62 |
| 21.3 | Installation and Setup..... | 63 |
| 21.3.1 | Safety Instructions..... | 64 |
| 21.3.2 | EtherCAT Onboard..... | 65 |
| 21.3.2.1 | LED functions..... | 65 |
| 21.3.2.2 | Connection technology..... | 65 |
| 21.3.2.3 | Network Connection Example..... | 65 |
| 21.3.3 | Guide to Setup..... | 66 |
| 21.3.4 | Setup via TwinCAT NC/PTP System Manager..... | 66 |
| 21.3.4.1 | Scan devices..... | 67 |
| 21.3.4.2 | Select the device..... | 67 |
| 21.3.4.3 | Scan for boxes..... | 68 |
| 21.3.4.4 | Add Slaves to NC tasks..... | 68 |
| 21.3.4.5 | Enable the network configuration..... | 69 |
| 21.3.4.6 | Enable the axis and move the axis..... | 70 |
| 21.3.5 | Setup WorkBench over TwinCAT..... | 71 |
| 21.3.5.1 | TwinCAT and WorkBenchconfiguration..... | 72 |
| 21.3.5.2 | Connecting to a drive using WorkBench..... | 73 |
| 21.3.5.3 | Configuring and enabling a drive..... | 76 |
| 21.3.6 | Setup via KAS IDE..... | 77 |

| | |
|---|------------|
| 21.4 EtherCAT Profile..... | 78 |
| 21.4.1 Slave Register..... | 79 |
| 21.4.2 AL Event (Interrupt Event) and Interrupt Enable..... | 80 |
| 21.4.2.1 Interrupt Enable Register (Address 0x0204:0x0205)..... | 80 |
| 21.4.2.2 AL Event Request (Address 0x0220:0x0221)..... | 81 |
| 21.4.3 Phase Run-Up..... | 82 |
| 21.4.3.1 AL Control (Address 0x0120:0x0121)..... | 82 |
| 21.4.3.2 AL Status (Address 0x0130:0x0131)..... | 82 |
| 21.4.3.3 AL Status Code (Address 0x0134:0x0135)..... | 83 |
| 21.4.3.4 EtherCAT communication phases..... | 83 |
| 21.4.4 CANopen over EtherCAT (CoE) Status Machine..... | 84 |
| 21.4.4.1 Status Description..... | 84 |
| 21.4.4.2 Commands in the Control Word..... | 85 |
| 21.4.4.3 Status Machine Bits (status word)..... | 86 |
| 21.4.5 Fixed PDO Mappings..... | 87 |
| 21.4.6 Supported Cyclical Setpoint and Actual Values..... | 88 |
| 21.4.7 Supported Operation Modes..... | 88 |
| 21.4.8 Adjusting EtherCAT Cycle Time..... | 89 |
| 21.4.9 Maximum Cycle Times depending on operation mode..... | 89 |
| 21.4.10 Synchronization..... | 90 |
| 21.4.10.1 Synchronization behavior with distributed clocks (DC) enabled..... | 90 |
| 21.4.10.2 Synchronization behavior with distributed clocks (DC) disabled..... | 90 |
| 21.4.11 Latch Control Word and Latch Status Word..... | 91 |
| 21.4.12 Mailbox Handling..... | 92 |
| 21.4.12.1 Mailbox Output..... | 93 |
| 21.4.12.2 Mailbox Input..... | 94 |
| 21.4.12.3 Example: Mailbox Access..... | 95 |
| 21.4.13 Fieldbus Parameters..... | 96 |
| 21.5 Index..... | 98 |
| 22 CANopen Manual..... | 104 |
| 22.1 General..... | 135 |
| 22.1.1 About this Manual..... | 136 |
| 22.1.2 Target Group..... | 136 |
| 22.1.3 Abbreviations used..... | 137 |
| 22.2 Safety..... | 138 |
| 22.2.1 Safety Instructions..... | 139 |
| 22.2.2 Use As Directed..... | 139 |
| 22.2.3 Prohibited Use..... | 139 |
| 22.3 Installation and Setup..... | 140 |
| 22.3.1 Safety Instructions..... | 141 |
| 22.3.2 Guide to Setup..... | 142 |
| 22.4 CANopen Basics..... | 142 |
| 22.4.1 Basic Features implemented by CANopen..... | 143 |
| 22.4.1.1 Setup and general functions:..... | 143 |
| 22.4.1.2 Positioning functions:..... | 143 |

| | |
|---|-----|
| 22.4.1.3 Data transfer functions:..... | 143 |
| 22.4.2 Transmission Rate and Procedure..... | 143 |
| 22.4.3 Response to BUSOFF Communication Faults..... | 144 |
| 22.4.4 Important Configuration Parameters..... | 144 |
| 22.5 CANopen Communication Profile..... | 144 |
| 22.5.1 General Description of CAN..... | 145 |
| 22.5.2 Construction of the Communication Object Identifier..... | 146 |
| 22.5.3 Definition of the Used Data Types..... | 147 |
| 22.5.3.1 Basic data types..... | 147 |
| Unsigned Integer..... | 147 |
| Signed Integer..... | 148 |
| 22.5.3.2 Mixed data types..... | 148 |
| 22.5.3.3 Extended data types..... | 149 |
| Octet String..... | 149 |
| Visible String..... | 149 |
| 22.5.4 Communication Objects..... | 149 |
| 22.5.4.1 Network Management Objects (NMT)..... | 150 |
| 22.5.4.2 Synchronization Object (SYNC)..... | 150 |
| 22.5.4.3 Time-Stamp Object (TIME)..... | 150 |
| 22.5.4.4 Emergency Object (EMCY)..... | 151 |
| Application of the Emergency Object..... | 151 |
| Composition of the Emergency Object..... | 151 |
| 22.5.4.5 Service Data Objects (SDO)..... | 152 |
| Composition of the Service Data Object..... | 152 |
| Initiate SDO Download Protocol..... | 154 |
| Download SDO Segment Protocol..... | 154 |
| Initiate SDO Upload Protocol..... | 154 |
| Upload SDO Segment Protocol..... | 154 |
| Abort SDO Protocol..... | 154 |
| 22.5.4.6 Process Data Object (PDO)..... | 155 |
| Transmission modes..... | 156 |
| Trigger modes..... | 156 |
| 22.5.4.7 Nodeguard..... | 157 |
| 22.5.4.8 Heartbeat..... | 158 |
| 22.6 CANopen Drive Profile..... | 159 |
| 22.6.1 CANopen Emergency Messages and Error Codes..... | 160 |
| 22.6.2 General Definitions..... | 163 |
| 22.6.2.1 General objects..... | 163 |
| Object 1000h: Device Type (DS301)..... | 163 |

| | |
|---|-----|
| Object 1001h: Error register (DS301) | 164 |
| Object 1002h: Manufacturer Status Register (DS301) | 165 |
| Object 1003h: Predefined Error Field (DS301) | 166 |
| Object 1005h: COB-ID of the SYNC Message (DS301) | 167 |
| Object 1006h: Communication Cycle Period (DS301) | 167 |
| Object 1008h: Manufacturer Device Name (DS301) | 168 |
| Object 1009h: Manufacturer Hardware Version | 168 |
| Object 100Ah: Manufacturer Software Version (DS301) | 168 |
| Object 100Ch: Guard Time (DS301)Response monitoring | 169 |
| Object 100Dh: Lifetime Factor (DS301) | 169 |
| Object 1010h: Store Parameters (DS301) | 170 |
| Object 1014h: COB-ID for Emergency Message (DS301) | 171 |
| Object 1016h: Consumer Heartbeat Time | 171 |
| Object 1017h: Producer Heartbeat Time | 172 |
| Object 1018h: Identity Object (DS301) | 172 |
| Object 1026h: OS Prompt | 174 |
| 22.6.2.2 Manufacturer specific objects | 175 |
| Object 2014-2017h: 1st-4th Mask 1 to 4 for Transmit-PDO | 175 |
| Object 2018h: Firmware Version | 176 |
| Object 2026h: ASCII Channel | 177 |
| Object 20A0h: Latch position 1, positive edge | 178 |
| Object 20A1h: Latch position 1, negative edge | 178 |
| Object 20A2h: Latch position 2, positive edge | 178 |
| Object 20A3h: Latch position 2, negative edge | 179 |
| Object 20A4h: Latch Control Register | 179 |
| Object 20A5h: Latch Status Register | 180 |
| Object 20A6h: Latch position 1, positive or negative edge | 180 |
| Object 20B8h: Reset of changed input information | 181 |
| 22.6.2.3 Profile specific objects | 182 |
| Object 60FDh: Digital inputs (DS402) | 182 |
| Object 60FEh: Digital outputs (DS402) | 182 |
| Object 6502h: Supported drive modes (DS402) | 183 |
| 22.6.3 PDO Configuration | 184 |
| 22.6.3.1 Receive PDOs (RXPDO) | 185 |

| | |
|--|-----|
| Objects 1400-1403h: 1st - 4th RXPDO communication parameter (DS301)..... | 185 |
| Objects 1600-1603h: 1st - 4th RXPDO mapping parameter (DS301)..... | 186 |
| Default RXPDO definition..... | 187 |
| 22.6.3.2 Transmit PDOs (TXPDO)..... | 188 |
| Objects 1800-1803h: 1st - 4th TXPDO communication parameter (DS301)..... | 188 |
| Objects 1A00-1A03h: 1st - 4th TXPDO mapping parameter (DS301)..... | 190 |
| Default TXPDO definition..... | 191 |
| 22.6.4 Device Control (dc)..... | 192 |
| 22.6.4.1 Status Machine (DS402)..... | 192 |
| States of the Status Machine..... | 193 |
| Transitions of the status machine..... | 194 |
| 22.6.4.2 Object Description..... | 195 |
| Object 6040h: Control word (DS402)..... | 195 |
| Object 6041h: Status word (DS402)..... | 196 |
| Object 6060h: Modes of Operation (DS402)..... | 198 |
| Object 6061h: Modes of Operation Display (DS402)..... | 199 |
| 22.6.5 Factor Groups (fg) (DS402)..... | 199 |
| 22.6.5.1 General Information..... | 199 |
| Factors..... | 199 |
| Relationship between Physical and Internal Units..... | 199 |
| 22.6.5.2 Objects for velocity scaling..... | 200 |
| Object 204Ch: pv scaling factor..... | 200 |
| 22.6.5.3 Objects for position calculation..... | 201 |
| Object 608Fh: Position encoder resolution (DS402)..... | 201 |
| Object 6092h: Feed constant (DS402)..... | 202 |
| 22.6.6 Profile Velocity Mode (pv) (DS402)..... | 203 |
| 22.6.6.1 General Information..... | 203 |
| Objects that are defined in this section..... | 203 |
| Objects that are defined in other sections..... | 203 |
| 22.6.6.2 Object description..... | 203 |
| Object 606Ch: velocity actual value (DS402)..... | 203 |
| Object 60FFh: target velocity (DS402)..... | 204 |
| 22.6.7 Profile Torque Mode (tq) (DS402)..... | 205 |
| 22.6.7.1 General Information..... | 205 |
| Objects that are defined in this section..... | 205 |

| | |
|---|-----|
| Objects that are defined in other sections..... | 205 |
| 22.6.7.2 Object description..... | 205 |
| Object 6071h: Target torque (DS402)..... | 205 |
| Object 6073h: Max current (DS402)..... | 205 |
| Object 6077h: Torque actual value (DS402)..... | 206 |
| 22.6.8 Position Control Function (pc) (DS402)..... | 207 |
| 22.6.8.1 General Information..... | 207 |
| Objects that are defined in this section..... | 207 |
| Objects that are defined in other sections..... | 207 |
| 22.6.8.2 Object Description..... | 207 |
| Object 6063h: position actual value* (DS402)..... | 207 |
| Object 6064h: position actual value (DS402)..... | 208 |
| Object 6065h: Following error window..... | 208 |
| Object 60F4h: Following error actual value (DS402)..... | 208 |
| 22.6.9 Interpolated Position Mode (ip) (DS402)..... | 209 |
| 22.6.9.1 General information..... | 209 |
| Objects defined in this section..... | 209 |
| Objects defined in other sections..... | 209 |
| 22.6.9.2 Object description..... | 209 |
| Object 60C0h: Interpolation sub mode select..... | 209 |
| Object 60C1h: Interpolation data record..... | 210 |
| Object 60C2h: Interpolation time period..... | 211 |
| Object 60C4h: Interpolation data configuration..... | 212 |
| 22.6.10 Homing Mode (hm) (DS402)..... | 214 |
| 22.6.10.1 General information..... | 214 |
| Objects that are defined in this section..... | 214 |
| Objects that are defined in other sections..... | 214 |
| 22.6.10.2 Object Description..... | 214 |
| Object 607Ch: homing offset (DS402)..... | 214 |
| Object 6098h: homing method (DS402)..... | 214 |
| Object 6099h: homing speeds (DS402)..... | 216 |
| Object 609Ah: homing acceleration (DS402)..... | 216 |
| Homing Mode Sequence..... | 217 |
| 22.6.11 Profile Position Mode (pp)..... | 218 |
| 22.6.11.1 General Information..... | 218 |

| | |
|---|------------|
| Objects that are defined in this section..... | 218 |
| Objects that are defined in other sections..... | 218 |
| 22.6.11.2 Object Description..... | 219 |
| Object 607Ah: target position (DS402)..... | 219 |
| Object 607Dh: Software position limit (DS402)..... | 219 |
| Object 6081h: profile velocity (DS402)..... | 220 |
| Object 6083h: profile acceleration (DS402)..... | 220 |
| Object 6084h: profile deceleration (DS402)..... | 220 |
| Functional Description..... | 221 |
| 22.7 Appendix..... | 223 |
| 22.7.1 Object Dictionary..... | 223 |
| 22.7.1.1 Float Scaling..... | 223 |
| 22.7.1.2 Communication SDOs..... | 223 |
| 22.7.1.3 Manufacturer specific SDOs..... | 226 |
| 22.7.1.4 Profile specific SDOs..... | 234 |
| 22.7.2 Examples..... | 236 |
| 22.7.2.1 Examples, setup..... | 236 |
| Basic testing of the connection to the AKD controls..... | 236 |
| Example: Operating the Status Machine..... | 237 |
| Example: Jog Mode via SDO..... | 238 |
| Example: Torque Mode via SDO..... | 238 |
| Example: Jog Mode via PDO..... | 239 |
| Example: Torque Mode via PDO..... | 241 |
| Example: Homing via SDO..... | 242 |
| Example: Using the Profile Position Mode..... | 244 |
| Example: ASCII Communication..... | 247 |
| Test for SYNC telegrams..... | 248 |
| 22.7.2.2 Examples, special applications..... | 249 |
| Example: External Trajectory with Interpolated Position Mode..... | 249 |
| 22.8 Index..... | 255 |
| Appendix B..... | 261 |
| | 262 |
| About the Parameter and Command Reference Guide..... | 263 |
| Parameter and Command Naming Conventions..... | 265 |
| Summary of Parameters and Commands..... | 265 |
| AIN Parameters..... | 279 |
| AIN.CUTOFF..... | 280 |
| AIN.DEADBAND..... | 281 |

| | |
|-------------------------------------|------------|
| AIN.DEADBANDMODE..... | 283 |
| AIN.ISCALE..... | 285 |
| AIN.OFFSET..... | 286 |
| AIN.PSCALE..... | 287 |
| AIN.VALUE..... | 288 |
| AIN.VSCALE..... | 289 |
| AIN.ZERO..... | 290 |
| AIO Parameters..... | 291 |
| AIO.ISCALE..... | 291 |
| AIO.PSCALE..... | 292 |
| AIO.VSCALE..... | 293 |
| AOUT Parameters..... | 295 |
| AOUT.DEBUGADDR..... | 295 |
| AOUT.DEBUGSCALE..... | 296 |
| AOUT.ISCALE..... | 297 |
| AOUT.MODE..... | 298 |
| AOUT.OFFSET..... | 299 |
| AOUT.PSCALE..... | 300 |
| AOUT.VALUE..... | 302 |
| AOUT.VALUEU..... | 303 |
| AOUT.VSCALE..... | 304 |
| BODE Parameters..... | 305 |
| BODE.EXCITEGAP..... | 305 |
| BODE.FREQ..... | 306 |
| BODE.IAMP..... | 307 |
| BODE.IFLIMIT..... | 308 |
| BODE.IFTHRESH..... | 309 |
| BODE.INJECTPOINT..... | 310 |
| BODE.MODE..... | 311 |
| BODE.MODETIMER..... | 314 |
| BODE.PRDEPTH..... | 316 |
| BODE.VAMP..... | 317 |
| BODE.VFLIMIT..... | 318 |
| BODE.VFTHRESH..... | 319 |
| CAP Parameters..... | 320 |
| CAP0.EDGE, CAP1.EDGE..... | 320 |
| CAP0.EN, CAP1.EN..... | 321 |
| CAP0.EVENT, CAP1.EVENT..... | 322 |
| CAP0.FILTER, CAP1.FILTER..... | 325 |
| CAP0.MODE, CAP1.MODE..... | 326 |
| CAP0.PLFB, CAP1.PLFB..... | 327 |
| CAP0.PREEDGE, CAP1.PREEDGE..... | 328 |
| CAP0.PREFILTER, CAP1.PREFILTER..... | 329 |
| CAP0.PRESELECT, CAP1.PRESELECT..... | 330 |
| CAP0.STATE, CAP1.STATE..... | 331 |

| | |
|---|------------|
| CAP0.T, CAP1.T..... | 332 |
| CAP0.TRIGGER, CAP1.TRIGGER..... | 333 |
| CS Parameters..... | 334 |
| CS.DEC..... | 334 |
| CS.STATE..... | 336 |
| CS.TO..... | 337 |
| CS.VTHRESH..... | 337 |
| DIN Parameters..... | 339 |
| DIN.HCMD1 TO DIN.HCMD4..... | 340 |
| DIN.LCMD1 to DIN.LCMD4..... | 341 |
| DIN.ROTARY..... | 341 |
| DIN.STATES..... | 343 |
| DIN1.FILTER TO DIN7.FILTER..... | 344 |
| DIN1.INV TO DIN7.INV..... | 345 |
| DIN1.MODE TO DIN7.MODE..... | 346 |
| DIN1.PARAM TO DIN7.PARAM..... | 348 |
| DIN1.STATE TO DIN7.STATE..... | 349 |
| DOUT Parameters..... | 351 |
| DOUT.CTRL..... | 351 |
| DOUT.RELAYMODE..... | 352 |
| DOUT.STATES..... | 353 |
| DOUT1.MODE AND DOUT2.MODE..... | 354 |
| DOUT1.PARAM AND DOUT2.PARAM..... | 355 |
| DOUT1.STATE AND DOUT2.STATE..... | 356 |
| DOUT1.STATEU AND DOUT2.STATEU..... | 357 |
| DRV Parameters and Commands..... | 359 |
| DRV.ACC..... | 359 |
| DRV.ACTIVE..... | 361 |
| DRV.BLINKDISPLAY..... | 362 |
| DRV.CLRFAULTHIST..... | 363 |
| DRV.CLRFAULTS..... | 364 |
| DRV.CMDDELAY..... | 365 |
| DRV.CMDSOURCE..... | 366 |
| DRV.CRASHDUMP..... | 367 |
| DRV.DBILIMIT..... | 368 |
| DRV.DEC..... | 369 |
| DRV.DIR..... | 370 |
| DRV.DIS..... | 372 |
| DRV.DISMODE..... | 373 |
| DRV.DISSOURCES..... | 374 |
| DRV.DISTO..... | 375 |
| DRV.EMUEDIR..... | 376 |
| DRV.EMUEMODE..... | 377 |
| DRV.EMUEMTURN..... | 378 |
| DRV.EMUERES..... | 379 |

| | |
|----------------------------|------------|
| DRV.EMUEZOFFSET..... | 380 |
| DRV.EN..... | 381 |
| DRV.ENDEFAULT..... | 382 |
| DRV.FAULTHIST..... | 383 |
| DRV.FAULTS..... | 384 |
| DRV.HANDWHEEL..... | 385 |
| DRV.HELP..... | 386 |
| DRV.HELPALL..... | 387 |
| DRV.HWENMODE..... | 388 |
| DRV.ICONT..... | 389 |
| DRV.INFO..... | 390 |
| DRV.IPEAK..... | 391 |
| DRV.IZERO..... | 392 |
| DRV.LIST..... | 393 |
| DRV.LOGICVOLTS..... | 394 |
| DRV.MEMADDR..... | 395 |
| DRV.MEMDATA..... | 396 |
| DRV.MOTIONSTAT..... | 397 |
| DRV.NAME..... | 398 |
| DRV.NVLIST..... | 399 |
| DRV.NVLOAD..... | 400 |
| DRV.NVSAVE..... | 401 |
| DRV.ONTIME..... | 402 |
| DRV.OPMODE..... | 403 |
| DRV.READFORMAT..... | 404 |
| DRV.RSTVAR..... | 405 |
| DRV.RUNTIME..... | 406 |
| DRV.SETUPREQBITS..... | 407 |
| DRV.SETUPPREQLIST..... | 408 |
| DRV.STOP..... | 409 |
| DRV.TEMPERATURES..... | 410 |
| DRV.TYPE..... | 411 |
| DRV.VER..... | 412 |
| DRV.VERIMAGE..... | 413 |
| DRV.WARNINGS..... | 414 |
| DRV.ZERO..... | 415 |
| FB1 Parameters..... | 416 |
| FB1.BISSBITS..... | 416 |
| FB1.ENCRES..... | 417 |
| FB1.HALLSTATE..... | 418 |
| FB1.HALLSTATEU..... | 419 |
| FB1.HALLSTATEV..... | 420 |
| FB1.HALLSTATEW..... | 421 |
| FB1.IDENTIFIED..... | 422 |
| FB1.INITSIGNED..... | 423 |

| | |
|----------------------------------|------------|
| FB1.MECHPOS..... | 424 |
| FB1.MEMVER..... | 425 |
| FB1.OFFSET..... | 426 |
| FB1.ORIGIN..... | 427 |
| FB1.PFIND..... | 429 |
| FB1.PFINDCMDU..... | 430 |
| FB1.POLES..... | 431 |
| FB1.PSCALE..... | 432 |
| FB1.RESKTR..... | 433 |
| FB1.RESREFPHASE..... | 434 |
| FB1.SELECT..... | 435 |
| FB1.TRACKINGCAL..... | 437 |
| FB2 Parameters..... | 438 |
| FB2.ENCREC..... | 438 |
| FB2.MODE..... | 439 |
| FB2.SOURCE..... | 440 |
| FBUS Parameters..... | 441 |
| FBUS.PARAM1 TO FBUS.PARAM20..... | 441 |
| FBUS.PLLSTATE..... | 443 |
| FBUS.PLLTHRESH..... | 444 |
| FBUS.SAMPLEPERIOD..... | 445 |
| FBUS.SYNCACT..... | 446 |
| FBUS.SYNCDIST..... | 447 |
| FBUS.SYNCWND..... | 448 |
| FBUS.TYPE..... | 449 |
| GEAR Parameters..... | 450 |
| GEAR.ACCMAX..... | 450 |
| GEAR.DECMAX..... | 451 |
| GEAR.IN..... | 452 |
| GEAR.MODE..... | 453 |
| GEAR.MOVE..... | 455 |
| GEAR.OUT..... | 456 |
| GEAR.VMAX..... | 457 |
| GUI Parameters..... | 458 |
| GUI.DISPLAY..... | 458 |
| GUI.PARAM01..... | 459 |
| GUI.PARAM02..... | 459 |
| GUI.PARAM03..... | 459 |
| GUI.PARAM04..... | 460 |
| GUI.PARAM05..... | 460 |
| GUI.PARAM06..... | 460 |
| GUI.PARAM07..... | 461 |
| GUI.PARAM08..... | 461 |
| GUI.PARAM09..... | 461 |
| GUI.PARAM10..... | 462 |

| | |
|------------------------------|------------|
| HOME Parameters | 463 |
| HOME.ACC..... | 463 |
| HOME.AUTOMOVE..... | 465 |
| HOME.DEC..... | 466 |
| HOME.DIR..... | 467 |
| HOME.DIST..... | 468 |
| HOME.FEEDRATE..... | 469 |
| HOME.IPEAK..... | 470 |
| HOME.MODE..... | 471 |
| HOME.MOVE..... | 472 |
| HOME.P..... | 473 |
| HOME.PERRTHRESH..... | 474 |
| HOME.REQUIRE..... | 475 |
| HOME.SET..... | 476 |
| HOME.V..... | 477 |
| HWLS Parameters | 478 |
| HWLS.NEGSTATE..... | 478 |
| HWLS.POSSTATE..... | 479 |
| IL Parameters | 481 |
| IL.BUSFF..... | 481 |
| IL.CMD..... | 482 |
| IL.CMDU..... | 483 |
| IL.DIFOLD..... | 484 |
| IL.FB..... | 485 |
| IL.FF..... | 486 |
| IL.FOLDFTHRESH..... | 487 |
| IL.FOLDFTHRESHU..... | 488 |
| IL.FOLDWTHRESH..... | 489 |
| IL.FRCTION..... | 490 |
| IL.IFOLD..... | 491 |
| IL.IUFB..... | 492 |
| IL.IVFB..... | 493 |
| IL.KACFF..... | 494 |
| IL.KBUSFF..... | 495 |
| IL.KP..... | 496 |
| IL.KPDRATIO..... | 497 |
| IL.KPLOOKUPINDEX..... | 498 |
| IL.KPLOOKUPVALUE..... | 499 |
| IL.KPLOOKUPVALUES..... | 500 |
| IL.KVFF..... | 501 |
| IL.LIMITN..... | 502 |
| IL.LIMITP..... | 503 |
| IL.MFOLDD..... | 504 |
| IL.MFOLDR..... | 505 |
| IL.MFOLDT..... | 506 |

| | |
|--|------------|
| IL.MIFOLD..... | 507 |
| IL.OFFSET..... | 508 |
| IL.VCMD..... | 509 |
| IL.VUFB..... | 510 |
| IL.VVFB..... | 511 |
| 23 LOAD.INERTIA..... | 512 |
| 24 MODBUS Parameters..... | 513 |
| MODBUS.PIN AND MODBUS.POUT..... | 514 |
| MODBUS.PSCALE..... | 515 |
| MODBUS.UNITLABEL..... | 516 |
| MOTOR Parameters..... | 517 |
| MOTOR.AUTOSET..... | 517 |
| MOTOR.BRAKE..... | 518 |
| MOTOR.BRAKERLS..... | 519 |
| MOTOR.BRAKESTATE..... | 520 |
| MOTOR.CTF0..... | 521 |
| MOTOR.ICONT..... | 522 |
| MOTOR.IDDATAVALID..... | 523 |
| MOTOR.INERTIA..... | 524 |
| MOTOR.IPEAK..... | 525 |
| MOTOR.KE..... | 526 |
| MOTOR.KT..... | 527 |
| MOTOR.LQLL..... | 528 |
| MOTOR.NAME..... | 529 |
| MOTOR.PHASE..... | 530 |
| MOTOR.PITCH..... | 531 |
| MOTOR.POLES..... | 532 |
| MOTOR.R..... | 533 |
| MOTOR.RTYPE..... | 534 |
| MOTOR.TBRAKEAPP..... | 535 |
| MOTOR.TBRAKERLS..... | 536 |
| MOTOR.TEMP..... | 537 |
| MOTOR.TEMPFAULT..... | 538 |
| MOTOR.TEMPWARN..... | 539 |
| MOTOR.TYPE..... | 540 |
| MOTOR.VMAX..... | 541 |
| MOTOR.VOLTMAX..... | 542 |
| MOTOR.VOLTMIN..... | 543 |
| MOTOR.VOLTRATED..... | 544 |
| MOTOR.VRATED..... | 545 |
| MT Parameters and Commands..... | 547 |
| MT.ACC..... | 547 |
| MT.CLEAR..... | 549 |
| MT.CNTL..... | 550 |
| MT.CONTINUE..... | 553 |

| | |
|---|------------|
| MT.DEC..... | 554 |
| MT.EMERGMT..... | 556 |
| MT.HOMEREQUIRE..... | 557 |
| MT.LIST..... | 558 |
| MT.LOAD..... | 559 |
| MT.MOVE..... | 560 |
| MT.MTNEXT..... | 561 |
| MT.NUM..... | 562 |
| MT.P..... | 563 |
| MT.PARAMS..... | 564 |
| MT.SET..... | 565 |
| MT.TNEXT..... | 566 |
| MT.TNUM..... | 567 |
| MT.TNVSAVE..... | 568 |
| MT.TPOSWND..... | 569 |
| MT.TVELWND..... | 570 |
| MT.V..... | 571 |
| MT.VCMD..... | 572 |
| PL Parameters..... | 573 |
| PL.CMD..... | 573 |
| PL.ERR..... | 574 |
| PL.ERRFTHRESH..... | 575 |
| PL.ERRMODE..... | 577 |
| PL.ERRWTHRESH..... | 578 |
| PL.FB..... | 579 |
| PL.FBSOURCE..... | 580 |
| PL.INTINMAX..... | 581 |
| PL.INTOUTMAX..... | 582 |
| PL.KI..... | 583 |
| PL.KP..... | 584 |
| PL.MODP1..... | 585 |
| PL.MODP2..... | 586 |
| PL.MODPDIR..... | 587 |
| PL.MODPEN..... | 588 |
| PLS Parameters and Commands..... | 589 |
| PLS.EN..... | 589 |
| PLS.MODE..... | 590 |
| PLS.P1 TO PLS.P8..... | 591 |
| PLS.RESET..... | 592 |
| PLS.STATE..... | 593 |
| PLS.T1 TO PLS.T8..... | 594 |
| PLS.UNITS..... | 595 |
| PLS.WIDTH1 TO PLS.WIDTH8..... | 597 |
| REC Parameters and Commands..... | 599 |
| REC.ACTIVE..... | 599 |

| | |
|------------------------------|------------|
| REC.CH1 to REC.CH6..... | 600 |
| REC.DONE..... | 601 |
| REC.GAP..... | 602 |
| REC.NUMPOINTS..... | 603 |
| REC.OFF..... | 604 |
| REC.RECPRMLIST..... | 605 |
| REC.RETRIEVE..... | 606 |
| REC.RETRIEVEDATA..... | 607 |
| REC.RETRIEVEFRMT..... | 608 |
| REC.RETRIEVEHDR..... | 609 |
| REC.RETRIEVESIZE..... | 610 |
| REC.STOPTYPE..... | 611 |
| REC.TRIG..... | 612 |
| REC.TRIGPARAM..... | 613 |
| REC.TRIGPOS..... | 614 |
| REC.TRIGPRMLIST..... | 615 |
| REC.TRIGSLOPE..... | 616 |
| REC.TRIGTYPE..... | 617 |
| REC.TRIGVAL..... | 618 |
| REGEN Parameters..... | 619 |
| REGEN.POWER..... | 619 |
| REGEN.REXT..... | 620 |
| REGEN.TEXT..... | 621 |
| REGEN.TYPE..... | 622 |
| REGEN.WATTEXT..... | 623 |
| SM Parameters..... | 625 |
| SM.I1..... | 625 |
| SM.I2..... | 626 |
| SM.MODE..... | 627 |
| SM.MOVE..... | 630 |
| SM.T1..... | 631 |
| SM.T2..... | 632 |
| SM.V1..... | 633 |
| SM.V2..... | 634 |
| STO Parameters..... | 635 |
| STO.STATE..... | 635 |
| SWLS Parameters..... | 636 |
| SWLS.EN..... | 636 |
| SWLS.LIMIT0..... | 637 |
| SWLS.LIMIT1..... | 638 |
| SWLS.STATE..... | 639 |
| UNIT Parameters..... | 641 |
| UNIT.ACCLINEAR..... | 641 |
| UNIT.ACCROTARY..... | 642 |
| UNIT.LABEL..... | 643 |

| | |
|-------------------------------|------------|
| UNIT.PIN..... | 644 |
| UNIT.PLINEAR..... | 645 |
| UNIT.POUT..... | 646 |
| UNIT.PROTARY..... | 647 |
| UNIT.VLINEAR..... | 648 |
| UNIT.VROTARY..... | 649 |
| VBUS Parameters..... | 650 |
| VBUS.OVFTHRESH..... | 650 |
| VBUS.OVWTHRESH..... | 651 |
| VBUS.RMSLIMIT..... | 652 |
| VBUS.UVFTHRESH..... | 653 |
| VBUS.UVMODE..... | 654 |
| VBUS.UVWTHRESH..... | 655 |
| VBUS.VALUE..... | 656 |
| VL Parameters..... | 657 |
| VL.ARPF1 TO VL.ARPF4..... | 657 |
| VL.ARPQ1 TO VL.ARPQ4..... | 659 |
| VL.ARTYPE1 TO VL.ARTYPE4..... | 660 |
| VL.ARZF1 TO VL.ARZF4..... | 661 |
| VL.ARZQ1 TO VL.ARZQ4..... | 662 |
| VL.BUSFF..... | 663 |
| VL.CMD..... | 664 |
| VL.CMDU..... | 665 |
| VL.ERR..... | 666 |
| VL.FB..... | 667 |
| VL.FBFILTER..... | 668 |
| VL.FBSOURCE..... | 669 |
| VL.FBUNFILTERED..... | 670 |
| VL.FF..... | 671 |
| VL.GENMODE..... | 672 |
| VL.KBUSFF..... | 673 |
| VL.KI..... | 674 |
| VL.KO..... | 675 |
| VL.KP..... | 676 |
| VL.KVFF..... | 677 |
| VL.LIMITN..... | 678 |
| VL.LIMITP..... | 679 |
| VL.LMJR..... | 679 |
| VL.MODEL..... | 681 |
| VL.OBSBW..... | 682 |
| VL.OBSMODE..... | 683 |
| VL.THRESH..... | 684 |
| WS Parameters..... | 685 |
| WS.ARM..... | 685 |
| WS.DISARM..... | 686 |

| | |
|----------------------|------------|
| WS.DISTMAX..... | 687 |
| WS.DISTMIN..... | 688 |
| WS.IMAX..... | 689 |
| WS.MODE..... | 690 |
| WS.NUMLOOPS..... | 691 |
| WS.STATE..... | 692 |
| WS.T..... | 693 |
| WS.TDELAY1..... | 694 |
| WS.TDELAY2..... | 695 |
| WS.TDELAY3..... | 696 |
| WS.VTHRESH..... | 697 |
| 25 Index..... | 700 |

This page intentionally left blank.

21.1 General

21.1.1 About this Manual

This manual, *AKD EtherCAT Communication*, describes the installation, setup, range of functions, and software protocol for the EtherCAT AKD product series. All AKD EtherCAT drives have built-in EtherCAT functionality; therefore an additional option card is not required.

A digital version of this manual (pdf format) is available on the CD-ROM included with your drive. Manual updates can be downloaded from the Kollmorgen website.

Related documents for the AKD series include:

- *AKD Quick Start* (also provided in hard copy). This guide provides instructions for basic drive setup and connection to a network.
- *AKD Installation Manual* (also provided in hard copy for EU customers). This manual provides instructions for installation and drive setup.
- *AKD Parameter and Command Reference Guide*. This guide provides documentation for the parameters and commands used to program the AKD.
- *AKD CANopen Communication*. This manual includes setup information for the CAN interface and describes the CANopen profile.

Additionally, an EtherCAT XML file, entitled *AKD EtherCAT Device Description*, describes the drive SDO and PDO. This file is available on the Kollmorgen website:

http://www.kollmorgen.com/website/com/eng/products/drives/ac_servo_drives/akd_drives_akd_software_tab.php

21.1.2 Target Group

This manual addresses personnel with the following qualifications:

- Installation: only by electrically qualified personnel.
- Setup: only by qualified personnel with extensive knowledge of electrical engineering and drive technology
- Programming: Software developers, project-planners

The qualified personnel must know and observe the following standards:

- ISO 12100, IEC 60364 and IEC 60664
- National accident prevention regulations

| |
|---|
| ⚠ WARNING During operation there are deadly hazards, with the possibility of death, severe injury or material damage. The operator must ensure that the safety instructions in this manual are followed. The operator must ensure that all personnel responsible for working with the servo drive have read and understand the manual. |
|---|

21.1.3 Abbreviations Used

| Abbreviation | Meaning |
|--------------|--|
| AL | Application Layer: the protocol that directly used by the process entities. |
| Cat | Category – classification for cables that is also used in Ethernet. |
| DC | Distributed Clocks Mechanism to synchronize EtherCAT slaves and master |
| DL | Data Link(=Layer 2). EtherCAT uses Ethernet, which is standardized as IEEE 802.3. |
| FPGA | Field Programmable Gate Array |
| FTP | File Transfer Protocol |
| HW | Hardware |
| ICMP | Internet Control Message Protocol: Mechanisms for signaling IP errors. |
| IEC | International Electrotechnical Commission: The international standards |
| IEEE | Institute of Electrical and Electronics Engineers, Inc. |
| LLDP | Link Layer Discovery Protocol |
| MAC | Media Access Control |
| MII | Media Independent Interface: Standardized interface Ethernet controller <-> routing equipment. |
| MDI | Media Dependant Interface: Use of connector Pins and Signaling. |
| MDI-X | Media Dependant Interface (crossed): Use of connector Pins and Signaling with crossed lines. |
| OSI | Open System Interconnect |
| OUI | Organizationally Unique Identifier – the first 3 Bytes of an Ethernet-Address, that will be assign to companies or organizations and can be used for protocoll identifiers as well (e.g. LLDP) |
| PDI | Physical Device Interface: set of elements that allows access to ESC from the process side. |
| PDO | Process Data Object |
| PDU | Protocol Data Unit: Contains protocol information transferred from a protocol instance of transparent data to a subordinate level |
| PHY | Physical interface that converts data from the Ethernet controller to electric or optical signals. |
| PLL | Phase Locked Loop |
| PTP | Precision Time Protocol in accordance with IEEE 1588 |
| RSTP | Rapid Spanning Tree Protocol |
| RT | Real-time, can be run in Ethernet controllers without special support. |
| RX | Receive |
| RXPDO | Receive PDO |
| SNMP | Simple Network Management Protocol |
| SPI | Serial Peripheral Interface |
| Src Addr | Source Address: Source address of a message. |
| STP | Shielded Twisted Pair |
| TCP | Transmission Control Protocol |
| TX | Transmit |
| TXPDO | Transmit PDO |
| UDP | User Datagram Protocol: Non-secure multicast/broadcast frame. |
| UTP | Unshielded Twisted Pair |
| ZA ECAT | Access mode EtherCAT |
| ZA Drive | Acces mode drive |

21.2 Safety

21.2.1 Safety Instructions

| | |
|------------------|---|
| ⚠ DANGER | <p>During operation there are deadly hazards, with the possibility of death, severe injury or material damage. Do not open or touch the equipment during operation. Keep all covers and cabinet doors closed during operation. Touching the equipment is allowed during installation and commissioning for properly qualified persons only.</p> <ul style="list-style-type: none"> • During operation, drives may have uncovered live components, depending on their level of enclosure protection. • Control and power connections may be live, even though the motor is not rotating. • Drives may have hot surfaces during operation. Heat sink can reach temperatures above 80°C. |
| ⚠ WARNING | <p>Electronic equipment can fail. The user is responsible for ensuring that, in the event of a failure of the servo amplifier, the drive is set to a state that is safe for both machinery and personnel, for instance with the aid of a mechanical brake.</p> <p>Drives with servo amplifiers and EtherCAT are remote-controlled machines. They can start to move at any time without previous warning. Take appropriate measures to ensure that the operating and service personnel is aware of this danger.</p> <p>Implement appropriate protective measures to ensure that any unintended start-up of the machines cannot result in dangerous situations for personnel or machinery. Software limit-switches are not a substitute for the hardware limit-switches in the machine.</p> |
| NOTICE | <p>Install the drive as described in the <i>Installation Manual</i>. Never break any of the electrical connections to the drive while it is live. This can result in destruction of the electronics</p> |
| NOTICE | <p>Do not connect the Ethernet line for the PC with the set up software to the EtherCAT interface X5/X6. The set up Ethernet cable must be connected to the service interface on X11</p> |

21.2.2 Use as directed

Drives are components that are built into electrical plants or machines and can only be operated as integral components of these plants or machines. The manufacturer of the machine used with a drive must generate a hazard analysis for the machine and take appropriate measures to ensure that unforeseen movements cannot cause personnel injury or property damage.

- Observe the chapters "Use as directed" and "Prohibited use" in the *AKD Installation Manual*.
- The EtherCAT interface serves only for the connection of the *AKD* to a master with EtherCAT connectivity.

21.2.3 Prohibited use

Other use than that described in chapter "Use as directed" is not intended and can lead to personnel injuries and equipment damage. The drive may not be used with a machine that does not comply with appropriate national directives or standards. The use of the drive in the following environments is also prohibited:

- potentially explosive areas
- environments with corrosive and/or electrically conductive acids, alkaline solutions, oils, vapors, dusts
- ships or offshore applications

The connectors X5 and X6 of the AKD EtherCAT drive may not be used for any ethernet protocol except EtherCAT (CoE, Can over EtherCAT).

21.3 Installation and Setup

21.3.1 Safety Instructions

⚠ DANGER Never disconnect any electrical connections to the drive while the drive is live. There is a danger of electrical arcing with damage to contacts and serious personal injury. Wait at least seven minutes after disconnecting the drive from the main supply power before touching potentially live sections of the equipment (e.g. contacts) or undoing any connections. Capacitors can still have dangerous voltages present up to 7 minutes after switching off the supply power. To be sure, measure the voltage in the DC Bus link and wait until it has fallen below 40 V.
Control and power connections can still be live, even if the motor is not rotating.

⚠ WARNING Electronic equipment can fail. The user is responsible for ensuring that, in the event of a failure of the servo amplifier, the drive is set to a state that is safe for both machinery and personnel, for instance with the aid of a mechanical brake.
Drives with servo amplifiers and EtherCAT are remote-controlled machines. They can start to move at any time without previous warning. Take appropriate measures to ensure that the operating and service personnel is aware of this danger.
Implement appropriate protective measures to ensure that any unintended start-up of the machines cannot result in dangerous situations for personnel or machinery. Software limit-switches are not a substitute for the hardware limit-switches in the machine.

NOTICE Install the drive as described in the *Installation Manual*. The wiring for the analog setpoint input and the positioning interface, as shown in the wiring diagram in the *Installation Manual*, is not required. Never break any of the electrical connections to the drive while it is live. This action can result in destruction of the electronics.

NOTICE The drive's status must be monitored by the PLC to acknowledge critical situations. Wire the FAULT contact in series into the emergency stop circuit of the installation. The emergency stop circuit must operate the supply contactor.

NOTE Use WorkBench to alter drive settings. Any other alterations will invalidate the warranty.

NOTE Because of the internal representation of the position-control parameters, the position controller can only be operated if the final limit speed of the drive does not exceed:

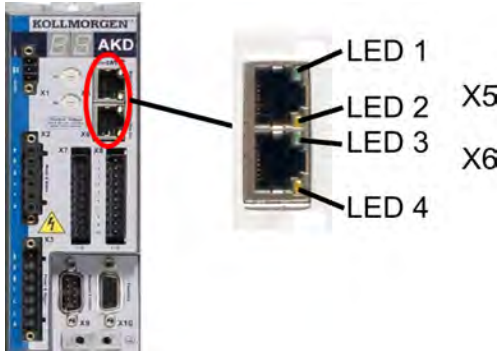
rotary
at sinusoidal² commutation: 7500 rpm
at trapezoidal commutation: 12000 rpm.

linear
at sinusoidal² commutation: 4 m/s
at trapezoidal commutation: 6.25 m/s

NOTE All the data on resolution, step size, positioning accuracy etc. refer to calculatory values. Non-linearities in the mechanism (backlash, flexing, etc.) are not taken into account. If the final limit speed of the motor must be altered, then all the parameters that were previously entered for position control and motion blocks must be adapted.

21.3.2 EtherCAT Onboard

Connection to the EtherCAT Network via X5 (in port) and X6 (out port).



21.3.2.1 LED functions

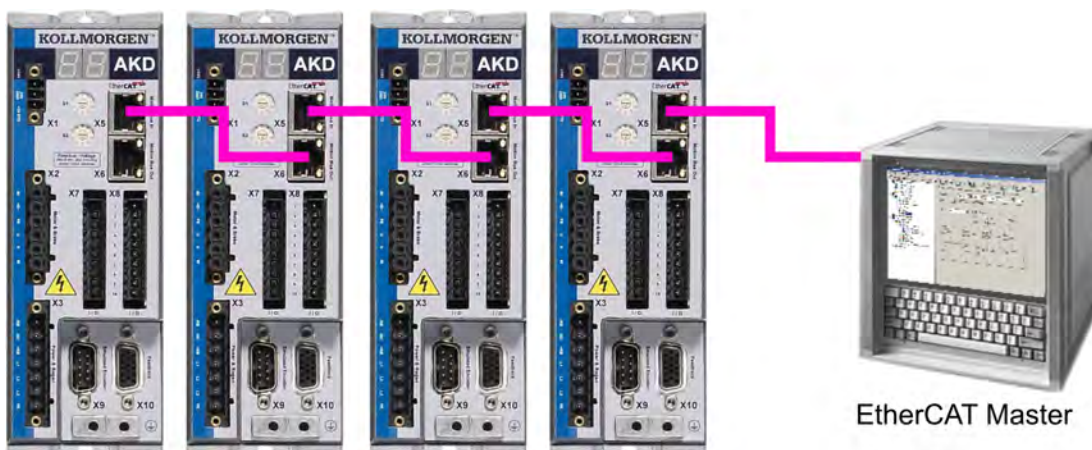
The communication status is indicated by the built-in LEDs.

| Connector | LED# | Name | Function |
|-----------|------|---------------|-----------------------------------|
| X5 | LED1 | IN port Link | ON = active OFF = not active |
| | LED2 | RUN | ON = running OFF = not running |
| X6 | LED3 | OUT port Link | ON = active OFF = not active |
| | LED4 | - | - |

21.3.2.2 Connection technology

You can connect to the EtherCAT network using RJ-45 connectors.

21.3.2.3 Network Connection Example



EtherCAT Master

21.3.3 Guide to Setup

⚠ WARNING

Only professional personnel with extensive knowledge of control and drive technology are allowed to setup the drive.

⚠ CAUTION

Make sure that any unintended movement of the drive cannot endanger machinery or personnel.

1. Check assembly/installation. Check that all the safety instructions in the product manual for the drive and this manual have been observed and implemented. Check the setting for the station address and baud rate.
2. Connect PC, start WorkBench. Use the setup software WorkBench to set the parameters for the drive.
3. Setup basic functions. Start up the basic functions of the drive and optimize the current, speed and position controllers. This section of the setup is described in the in the online help of the setup software.
4. Save parameters. When the parameters have been optimized, save them in the drive.

21.3.4 Setup via TwinCAT NC/PTP System Manager

Before you set up the drive, make sure the following have been completed:

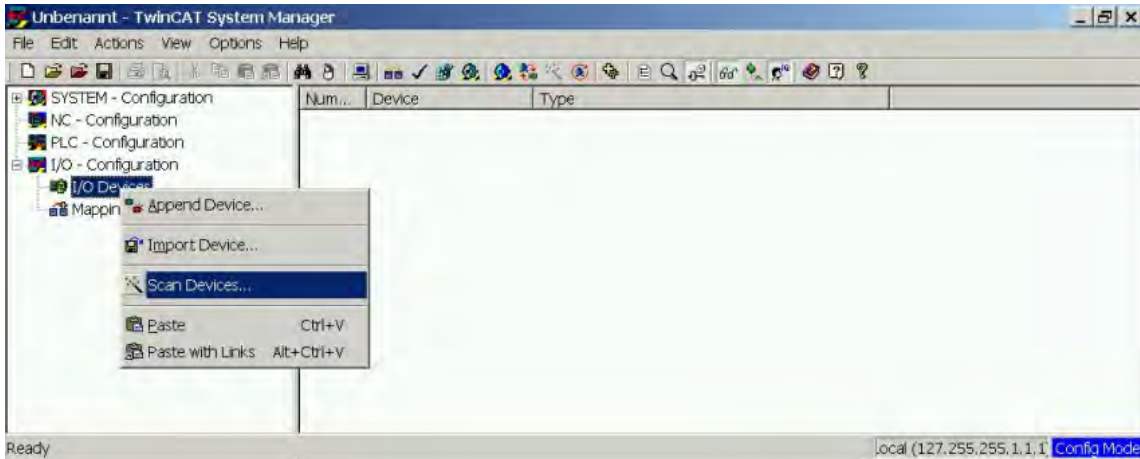
- The AKD is configured with WorkBench and the servomotor is able to move
- A correctly configured EtherCAT card is present in the master.
- TwinCAT software from Beckhoff (NC/PTP-Mode setup) is installed. Install first the TwinCAT System Manager, restart your PC, then install the option package NC/PTP-Mode.
- The XML description of the drive is available (the XML file on the CD-ROM or on the Kollmorgen website).
- An AKD EtherCAT slave is connected to the EtherCAT master PC.
- The TwinCAT system manager resides in Config-Mode. The current mode of the system manager is displayed of the bottom right side of the TwinCAT main-screen window.

Copy the XML description of the drive to the TwinCAT system (usually to the folder c:\TwinCAT\IO\EtherCAT) and restart the TwinCAT system since TwinCAT analyzes all device description files during start-up.

The following example explains the automatic EtherCAT network setup. The network setup can also be done manually; please refer to the TwinCAT manual for more details.

21.3.4.1 Scan devices

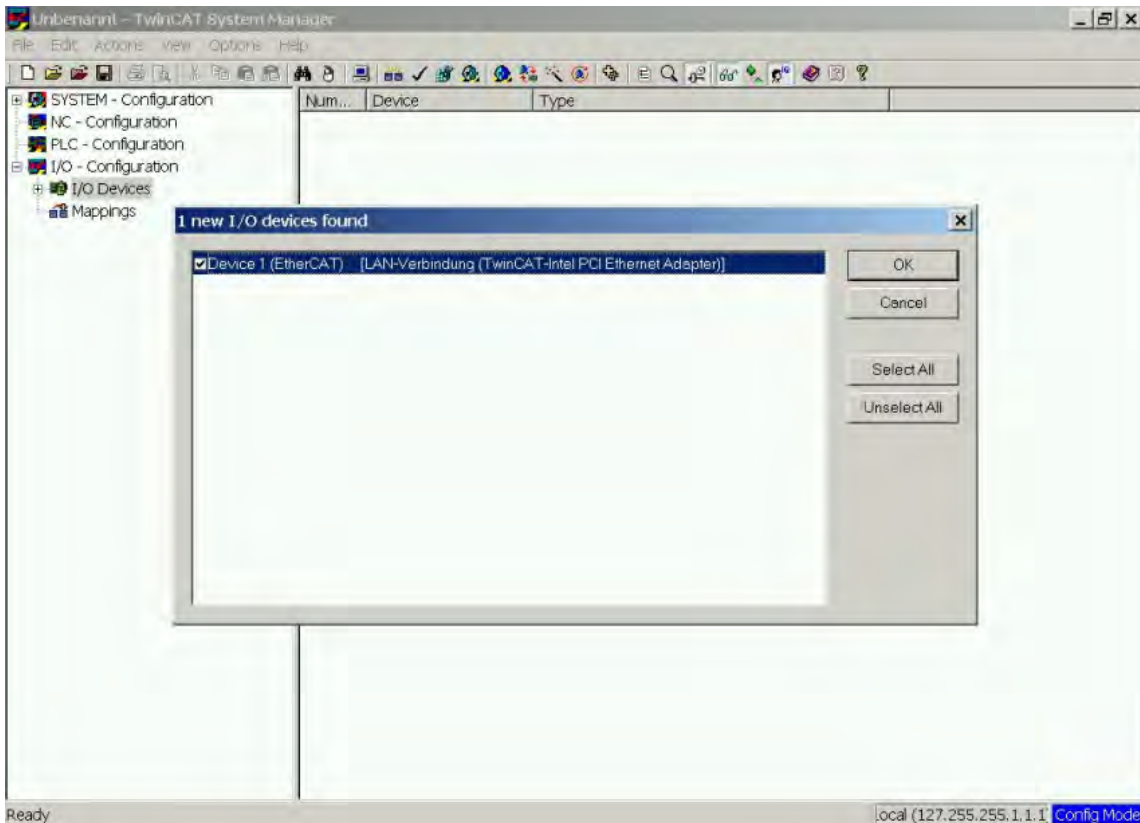
First ensure that the EtherCAT master is physically connected to the EtherCAT AKD. Create a new (empty) project. Right click I/O-Devices and scan for the devices. An example is included in the EtherCAT network card, which is plugged into the PC.



A pop-up window informs you that not all devices can be detected by the TwinCAT software. Click **OK** to continue.

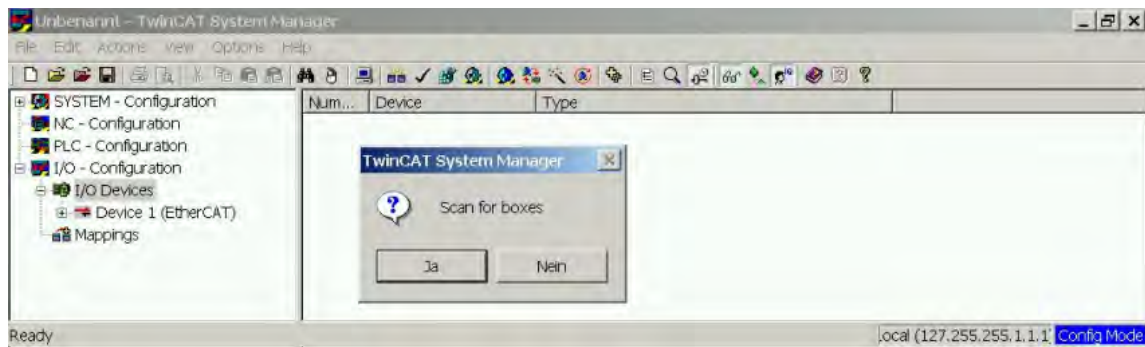
21.3.4.2 Select the device

TwinCAT must be able to find the EtherCAT network card. An EtherCAT slave must be connected to the network card; otherwise TwinCAT will find a real-time EtherNET card instead of the EtherCAT card. Press the **OK** button.



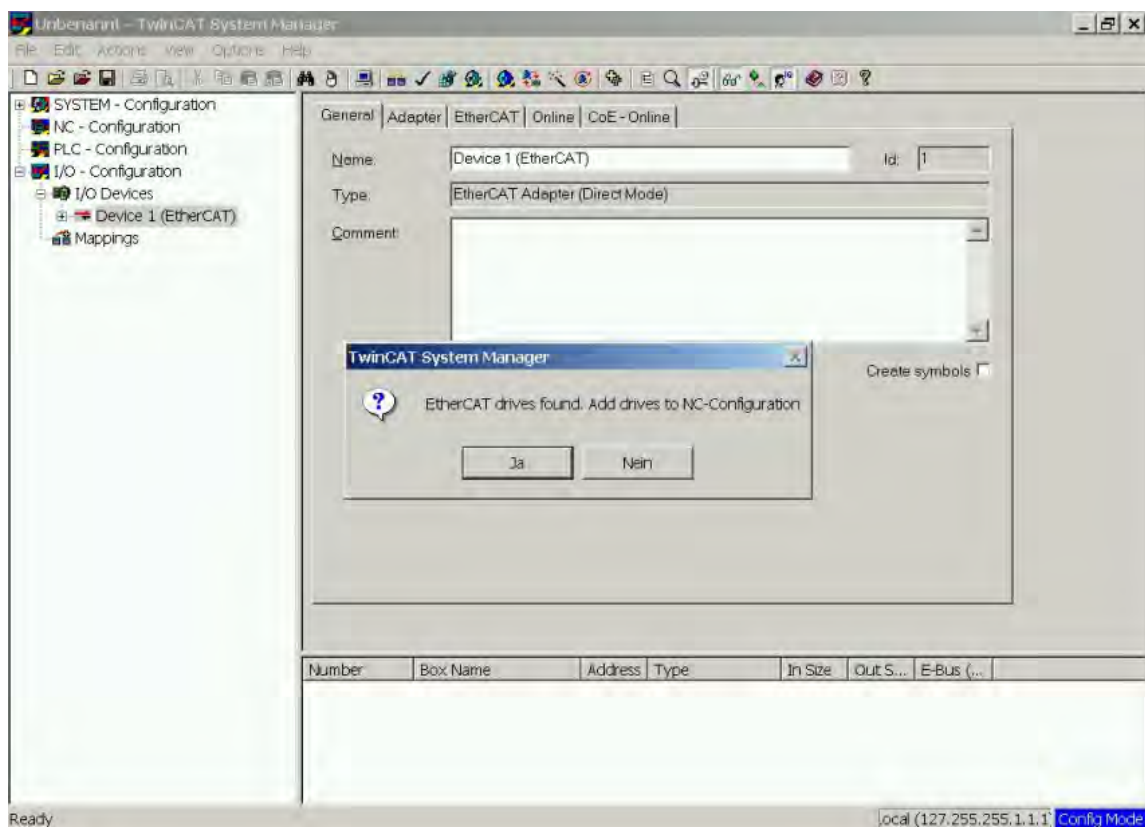
21.3.4.3 Scan for boxes

Click **Yes** to allow TwinCAT to scan for boxes. A box is an alias for a slave device and is always used in Beckhoff software products.






21.3.4.4 Add Slaves to NC tasks

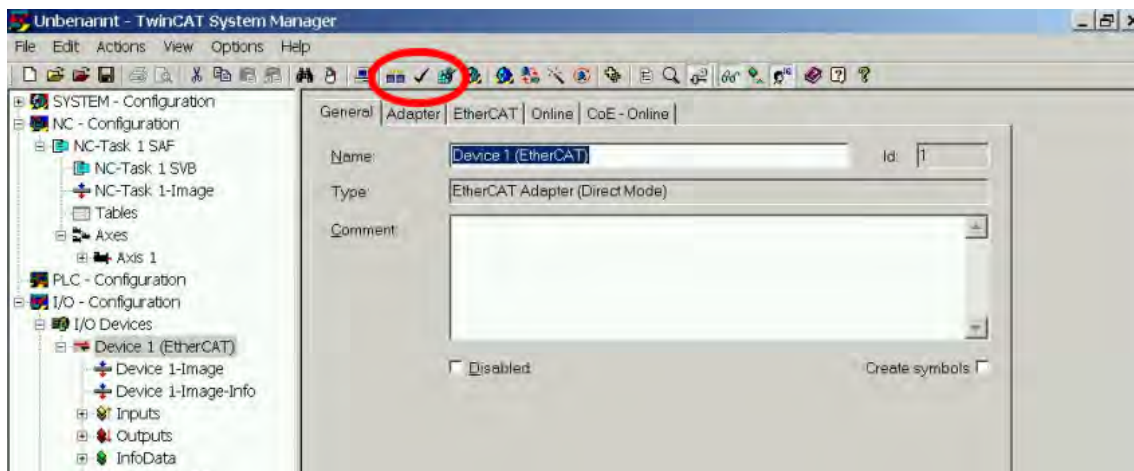
TwinCAT should now have identified the AKD according to the Device Description file. TwinCAT next asks if the slaves should be connected to NC tasks. Click **Yes** to continue. An NC task can, for example, contain a PLC program, which can be programmed by the user.



21.3.4.5 Enable the network configuration

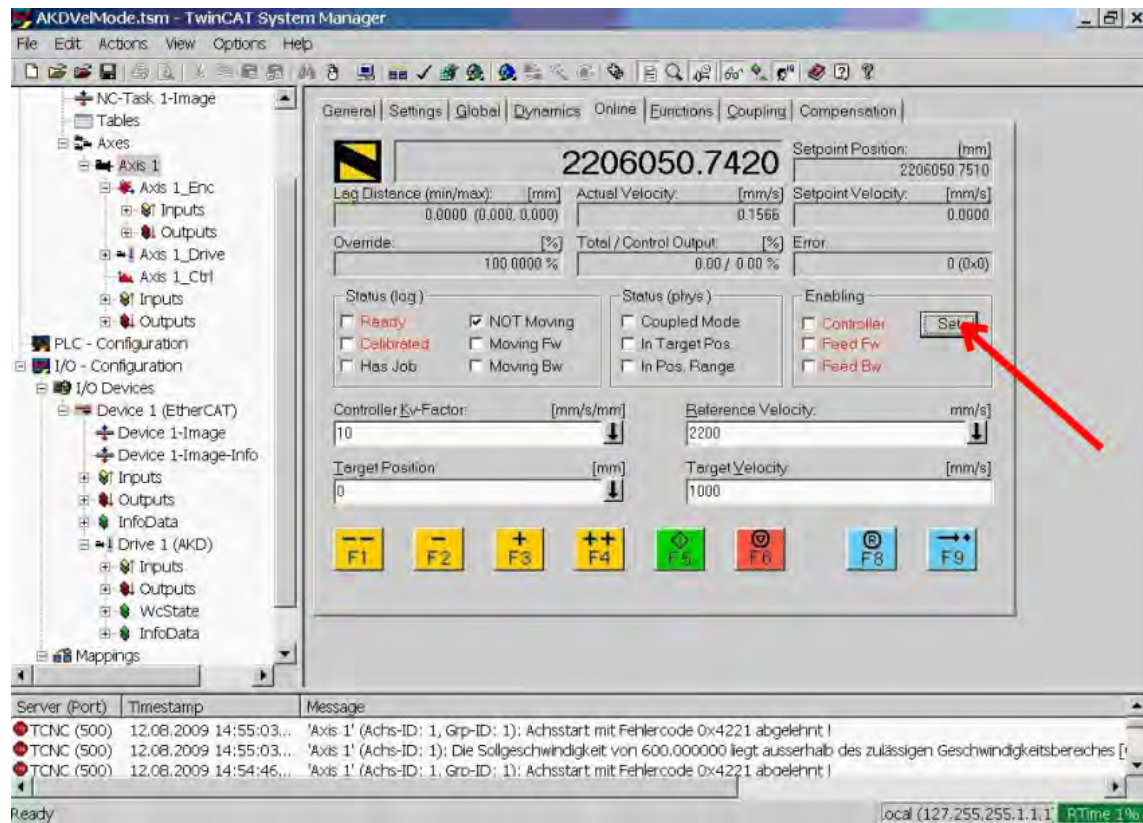
Confirm that the AKD appears in the device tree. Next, enable the network configuration. Press first the  button in order to generate the mappings, afterwards press the  button in order to let TwinCAT check the configuration and use finally the  button in order to step into run-mode.

Confirm afterwards that TwinCAT is allowed to jump into run-mode.



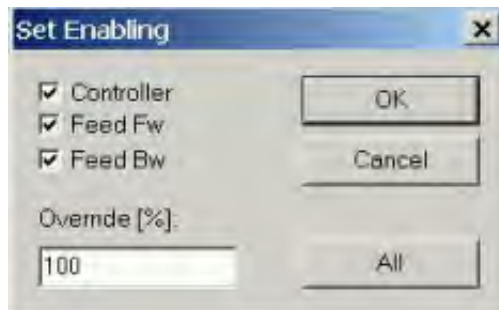
21.3.4.6 Enable the axis and move the axis

The Axis can be enabled by a mouse-click on the Set button within the Online window inside of each Axis, see also the next picture.



Afterwards a pop-up window appears.

The following setting enables the drive and allows command values in both directions.



Afterwards the motor should move in positive or negative direction as soon as the clicks on the following yellow buttons within the Online window:



21.3.5 Setup WorkBench over TwinCAT

This chapter describes a quick start guide for a user to be able to setup a WorkBench over TwinCAT system and be able to make a motor spin under that system.

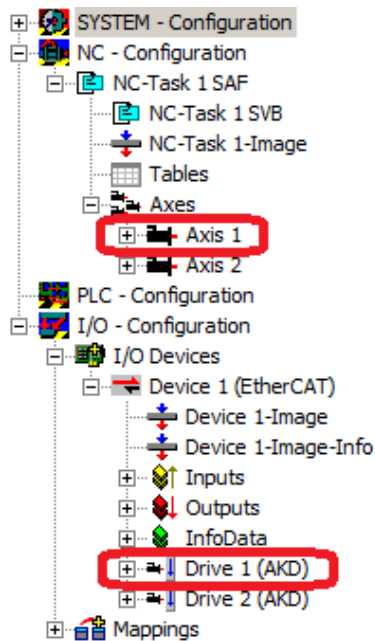
This chapter does not give any specific details on TwinCAT system or WorkBench alone but is giving guidelines and information on how TwinCAT master and WorkBench works together.

Main steps in configuring a WorkBench over TwinCAT system are:

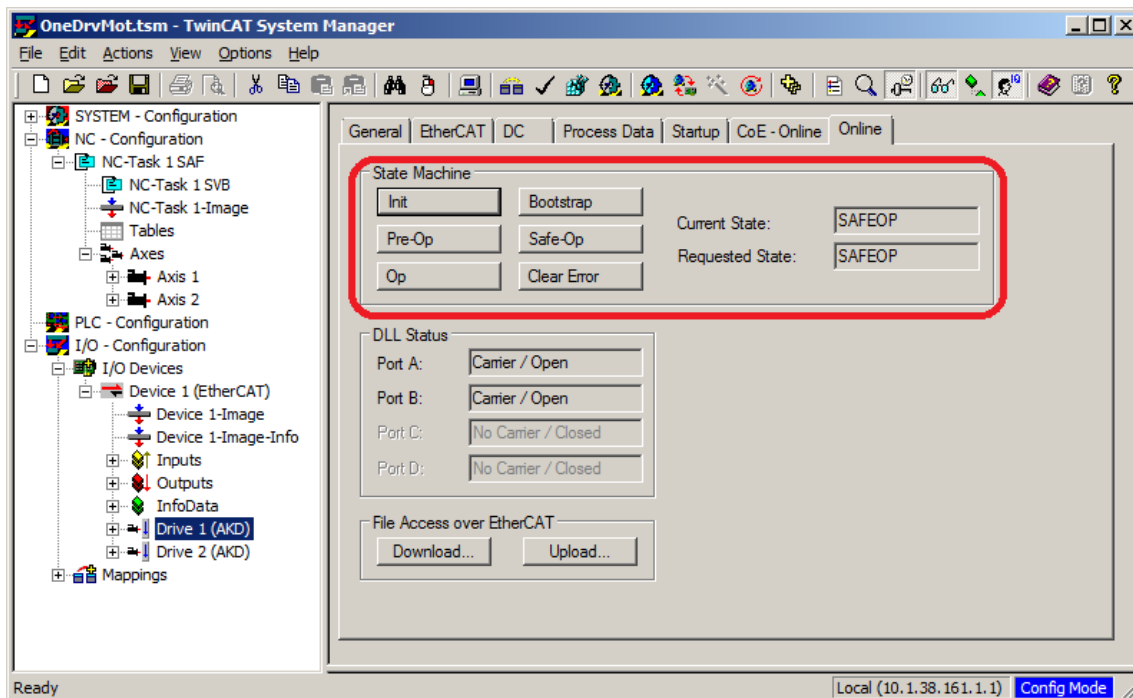
1. TwinCAT and WorkBench configuration
2. Connecting to a drive using WorkBench
3. Configuring and enabling a drive

21.3.5.1 TwinCAT and WorkBenchconfiguration

The EtherCAT network must be setup and managed using TwinCAT System Manager. To be able to connect to a drive and enable it, the drive must be loaded under the I/O Devices node in TwinCAT System Manager and axis must be added to NC - Configuration as shown in "Setup via TwinCAT NC/PTP System Manager" (=> p. 66) .



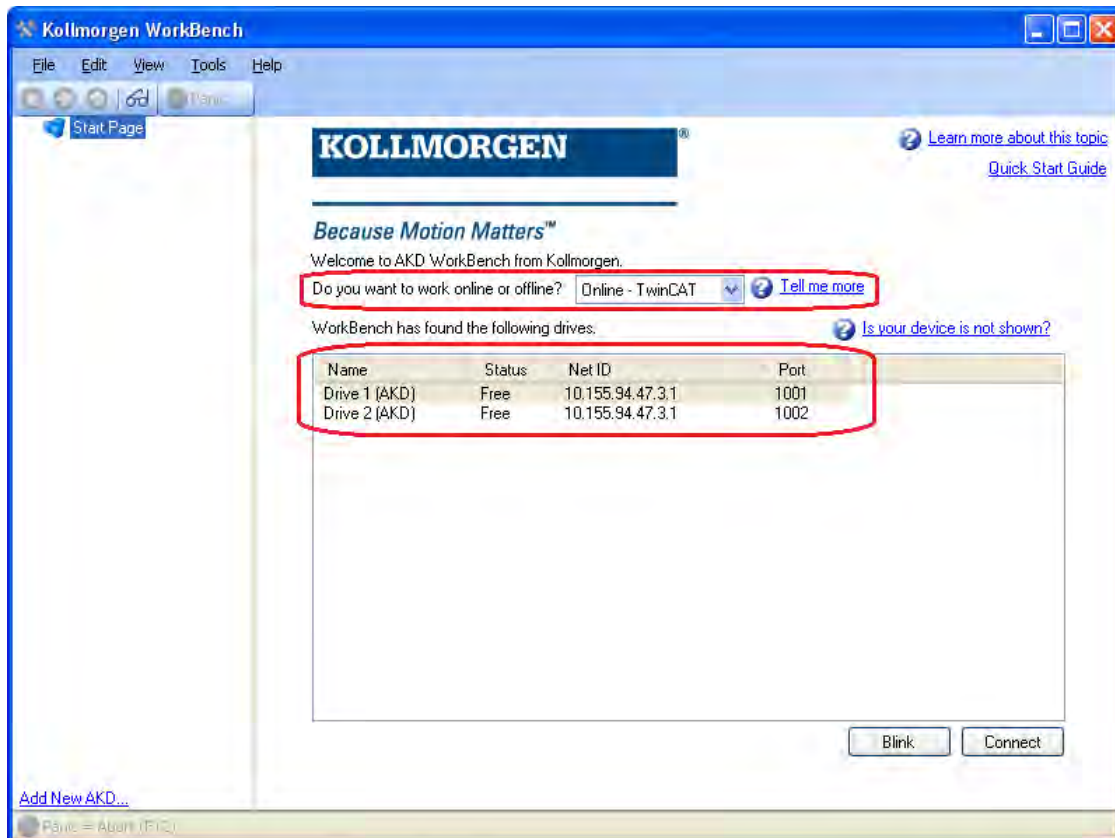
In order to connect to the drives using WorkBench, the drives must be either in Pre-Op, Safe-Op or Op state. State machine for a drive can be accessed from the Online tab for the corresponding drive under the I/O Configuration → I/O Devices → Device [x] → Drive [x] node (see screenshot below).



Installation process for WorkBench is the same process as normal, except that it must be installed on the same machine as TwinCAT. Communication to the drive is done thru TwinCAT master and it's not possible to connect WorkBench to the master remotely.

21.3.5.2 Connecting to a drive using WorkBench

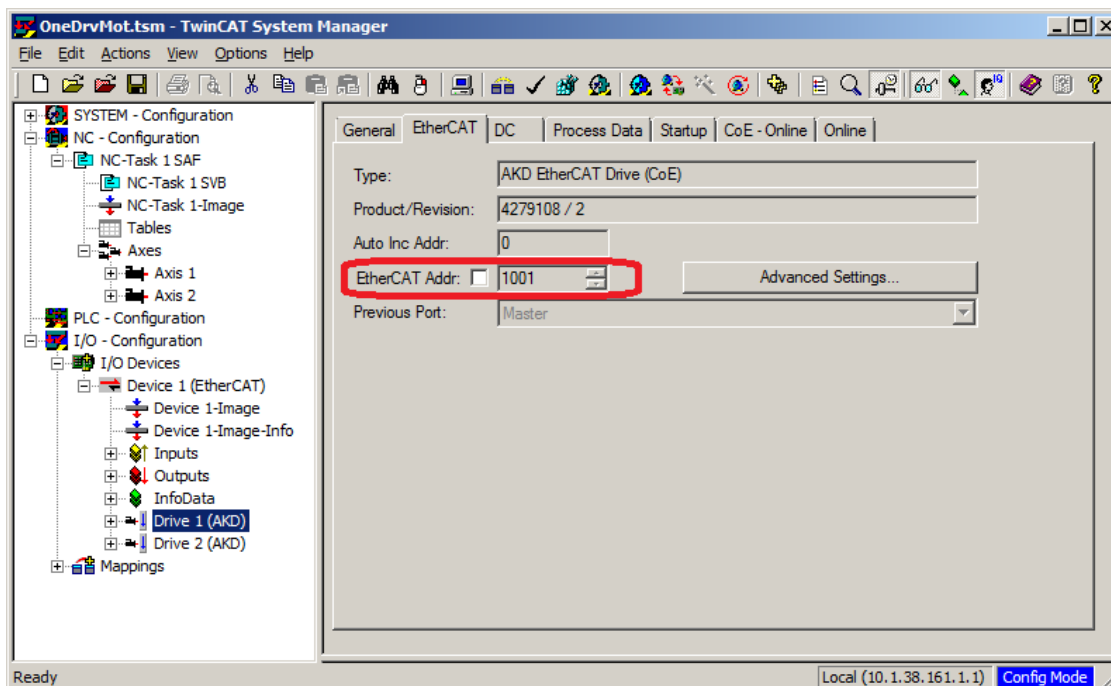
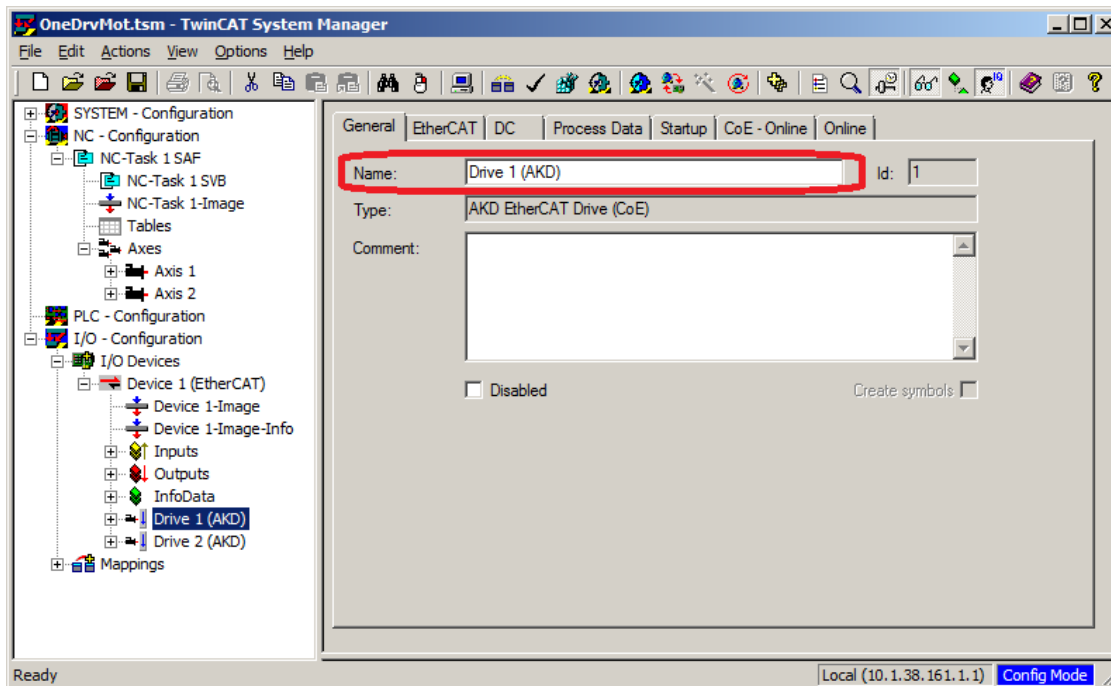
In order to connect to a drive, a TwinCAT device must be added in WorkBench. The start page of WorkBench can be used to do this. First, the type of drive (Online - TwinCAT) must be specified. Then, a list of available drives will be provided.



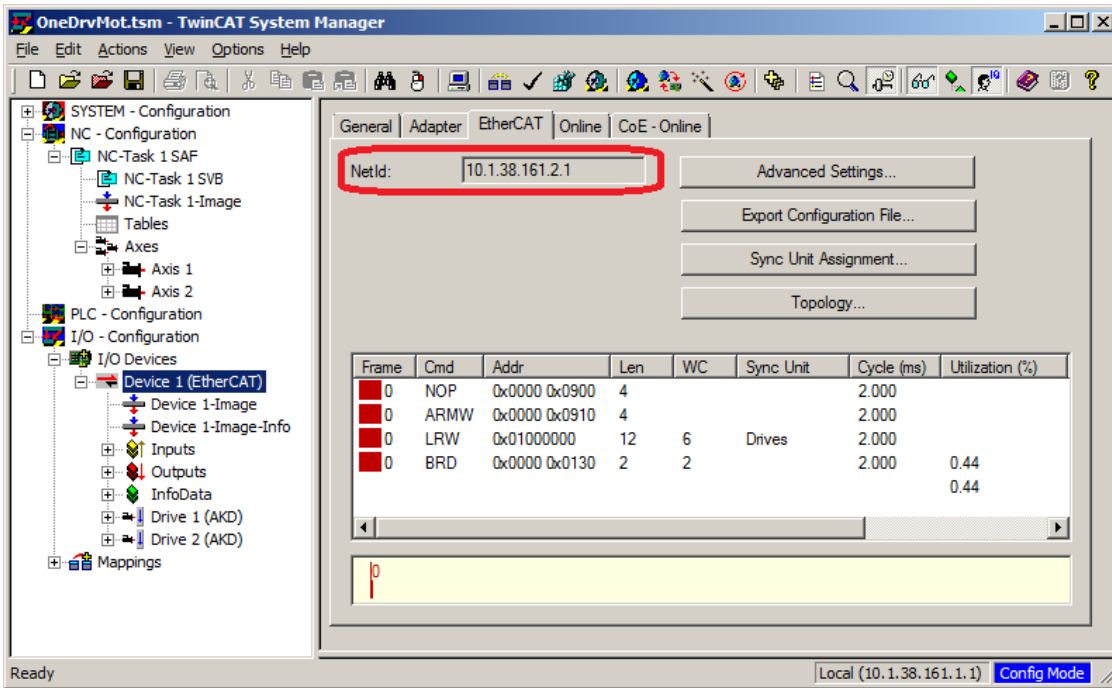
The information provided for a drive are its name, status, Net ID and Port number. After selecting a drive from the list, clicking on the "Connect" button will create a device in the left frame of WorkBench and connect the device.

The name, Net ID and port number are information coming from the TwinCAT master configuration file (the name may be different than the drive name returned by the *DRV.NAME* command). While the status is an indicator that tells if there is already a device created within WorkBench which is already connected to that particular drive.

Using TwinCAT System Manager, the drive name and port number can be found in the General and EtherCAT tab respectively for the corresponding drive under the I/O Configuration → I/O Devices → Device [x] → Drive [x] node.



The Net ID can be found in the EtherCAT tab in the I/O Configuration → I/O Devices → Device [x] node.



It is important to understand that these information are coming from the TwinCAT master and its configuration file but not from the drive itself. Thus, if the TwinCAT configuration is not reflecting the actual network configuration, you may have a drive listed in WorkBench which is not be powered up or even connected in the EtherCAT network, or you have a drive powered up and connected to the TwinCAT network but not shown in the WorkBench list.

21.3.5.3 Configuring and enabling a drive

Once connected with WorkBench, a drive can be configured using all normal functionalities of WorkBench.

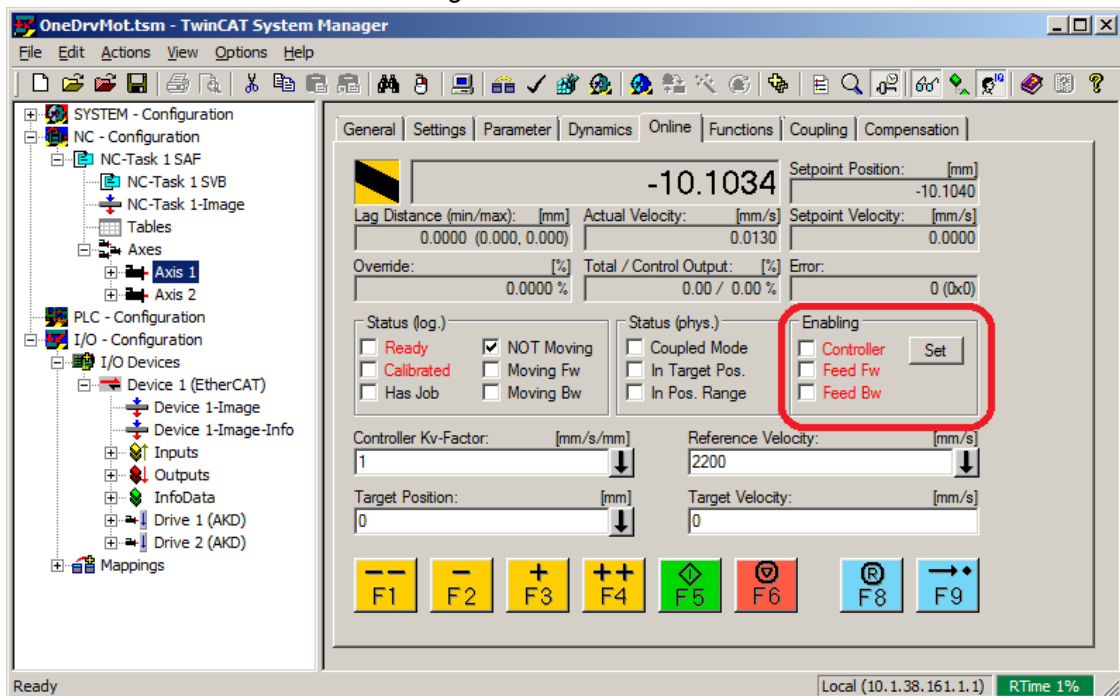
The only operation that is not possible to do using WorkBench over TwinCAT is the download of a new firmware in the drive. Downloading a new firmware in the drive must be performed using File over EtherCAT (FoE) feature of TwinCAT server.

NOTICE

If the cyclic communication of the TwinCAT master is enabled, it is possible that some commands sent by WorkBench using the ASCII channel are overwritten by the TwinCAT master. Typically, the drive enable command will have no effect if sent from WorkBench because the control word is usually mapped.

Using TwinCAT, enabling the drive can be done with the following procedure:

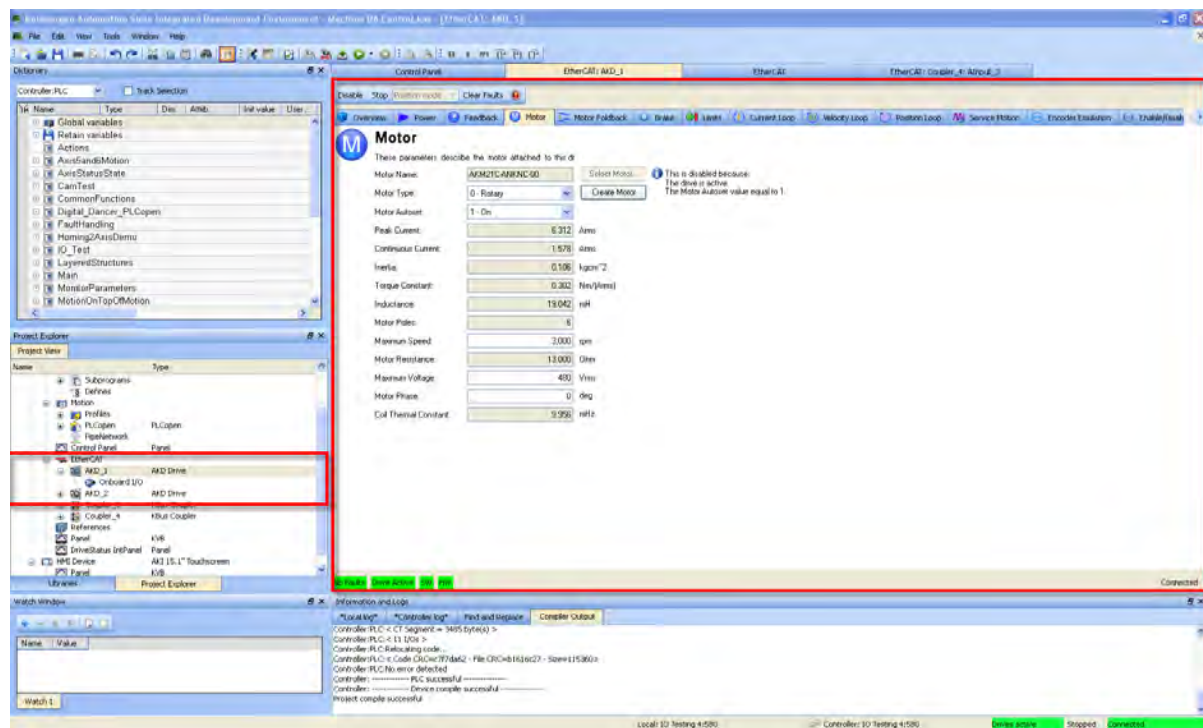
1. Under NC Configuration → Axes → Axis [x] node, choose the Online tab.
2. Press the Set button within the Enabling section.



3. In the pop-up dialog box, check the Controller checkbox to enable the drive (or un-check to disable the drive) and press on the OK button.

21.3.6 Setup via KAS IDE

If you are using a Kollmorgen Automation Suite (KAS) system, the AKD setup is completely integrated into the KAS Integrated Development Environment (IDE), as shown below:



For further information on the setup for a KAS system, see the following sections in the KAS documentation:

- *KAS IDE User Manual*: See section 4.2.3 Add and Configure Drive.
- *KAS Online Help*: See **Using the KAS IDE> Creating a Project> Step 3 - Add and Configure Drive**.

21.4 EtherCAT Profile

21.4.1 Slave Register

The table below gives the addresses of individual registers in the FPGA memory. The data is provided in little-endian format, with the 'least significant byte' occupying the lowest address. A detailed description of all registers and FPGA memory locations is available in the "EtherCAT Slave Controller" description of the EtherCAT user organization (www.EtherCAT.org).

| Address | Length (Byte) | Description | ZA ECAT* | ZA Drive* |
|---------|---------------|---|----------|-----------|
| 0x0120 | 2 | AL Control | R/W | R/O |
| 0x0130 | 2 | AL Status | R/O | R/W |
| 0x0134 | 2 | AL Status Code | R/O | R/W |
| 0x0204 | 2 | Interrupt Enable Register | R/O | R/W |
| 0x0220 | 2 | AL Event (IRQ Event) | R/W | R/O |
| 0x0800 | 8 | Sync Manager 0 (Mail Out Control Register) | R/W | R/O |
| 0x0808 | 8 | Sync Manager 1 (Mail In Control Register) | R/W | R/O |
| 0x0810 | 8 | Sync Manager 2 (Process data Output Control Register) | R/W | R/O |
| 0x0818 | 8 | Sync Manager 3 (Process data Input Control Register) | R/W | R/O |
| 0x0820 | 8 | Sync Manager 4 | R/W | R/O |
| 0x0828 | 8 | Sync Manager 5 | R/W | R/O |
| 0x0830 | 8 | Sync Manager 6 | R/W | R/O |
| 0x0838 | 8 | Sync Manager 7 | R/W | R/O |
| 0x0840 | 8 | Sync Manager 8 | R/W | R/O |
| 0x1100 | Max. 64 | ProOut Buffer (Process data Output, setpoints ECAT) | R/W | R/O |
| 0x1140 | Max. 64 | ProIn (Process data Input, act. values ECAT) | R/O | R/W |
| 0x1800 | 512 | Mail Out Buffer (Object Channel Buffer ECAT, byte-length is specified in the device description file) | R/W | R/O |
| 0x1C00 | 512 | Mail In Buffer (Object Channel Buffer Drive, byte-length is specified in the device description file) | R/O | R/W |

* ZA ECAT = Access mode EtherCAT

* ZA Drive = Access mode drive

21.4.2 AL Event (Interrupt Event) and Interrupt Enable

Communication between the drive and the EtherCAT FPGA can be interrupt-driven. The interrupt enable register and the AL event register are responsible for the EtherCAT interface interrupt functionality.

There are two events which lead also to a HW interrupt within the drive, the EEPROM emulation event and the SyncManager 2 event. The actual values of the drive (SyncManager 3 data) are written without any AL event request during each HW IRQ, e.g. triggered by a SyncManager 2 event. The Mailbox exchange between the master and the AKD is completely handled by polling the AL event register within the background task of the drive.

The drive activates individual EtherCAT interface events when the corresponding bit of the interrupt enable register is set to 1. When it is set to 0, the hardware interrupts for the specific events are deactivated.

21.4.2.1 Interrupt Enable Register (Address 0x0204:0x0205)

| Parameter | Address | Bit | ZA Drive | ZA ECAT | Description |
|--|---------|--------|----------|---------|---|
| AL Control Event | 0x204 | 0 | R/W | R/O | Activation of AL control event for phase run-up |
| - | 0x204 | 1 | R/W | R/O | Reserved |
| Sync0 DC Distributed Clock | 0x204 | 2 | R/W | R/O | Activation of distributed clock (DC) sync 0 interrupts for entire communication |
| Sync1 DC Distributed Clock | 0x204 | 3 | R/W | R/O | Activation of distributed clock (DC) sync 1 interrupts for entire communication |
| SyncManager activation register change | 0x204 | 4 | R/W | R/O | Activation of 'SyncManager activation register change' IRQ. |
| EEPROM emulation event | 0x204 | 5 | R/W | R/O | Activation of the EEPROM emulation interrupts. |
| - | 0x204 | 3 to 7 | R/W | R/O | Reserved |
| Sync Manager 0 Event (Mail Out Event) | 0x205 | 0 | R/W | R/O | Activation of output event mailbox (SDO, Sync Manager 0) for object channel. |
| Sync Manager 1 Event (Mail In Event) | 0x205 | 1 | R/W | R/O | Activation of input event mailbox (SDO, Sync Manager 1) for object channel. |
| Sync Manager 2 Event (Pro Out Event) | 0x205 | 2 | R/W | R/O | Activation of output event process data (PDO, card's cyclical setpoints) |
| Sync Manager 3 Event (Pro In Event) | 0x205 | 3 | R/W | R/O | Activation of input event process data (PDO, drive's cyclical actual values) |
| - | 0x205 | 4 to 7 | R/W | R/O | Reserved |

21.4.2.2 AL Event Request (Address 0x0220:0x0221)

When the relevant bit of the AL event request register is set to 1, the EtherCAT interface tells the drive which event it should process by the AKD.

| Parameter | Address | Bit | ZA Drive | ZA ECAT | Description |
|--|---------|--------|----------|---------|--|
| AL Control Event | 0x220 | 0 | R/O | R/W | Processing of AL control event for phase run-up |
| Sync0 Distributed Clock (DC) Event | 0x220 | 2 | R/O | R/W | Processing of a distributed clock (DC) event |
| Sync1 Distributed Clock (DC) Event | 0x220 | 3 | R/O | R/W | Processing of a distributed clock (DC) event |
| SyncManager activation register change | 0x220 | 4 | R/O | R/W | The content of the SyncManager activation register has been changed. |
| EEPROM emulation event | 0x220 | 5 | R/O | R/W | Processing of an EEPROM emulation event in order to identify the AKD within the network. |
| - | 0x220 | 6 to 7 | R/O | R/W | Reserved |
| Sync Manager 0 Event | 0x221 | 0 | R/O | R/W | Mailbox request (SDO, Sync Manager 0) for object channel. |
| Sync Manager 1 Event | 0x221 | 1 | R/O | R/W | Mailbox response (SDO, Sync Manager 1) for object channel. |
| Sync Manager 2 Event | 0x201 | 2 | R/O | R/W | Process data output (PDO, card's cyclical set-points) |
| Sync Manager 3 Event | 0x201 | 3 | R/O | R/W | Process data input (PDO, drive's cyclical actual values) |
| Sync Manager 4 – | | | | | |
| Sync Manager 7 Event | 0x221 | 4 to 7 | R/O | R/W | Reserved |
| Sync Manager 8 – | | | | | |
| Sync Manager 15 Event | 0x222 | 0...7 | R/O | R/W | Reserved |

21.4.3 Phase Run-Up

The AL control, AL status and AL status code registers are responsible for communication phase run-up (also referred to as EtherCAT status change), for current status display and for any fault messages. The drive responds to every EtherCAT interface transition request made by the AL control register via the AL Status and AL Status Code registers. Any fault messages are displayed in the AL status code register.

A status change within the AL control register is polled within the AKD, which means that an AL control event does not lead to a HW interrupt within the drive.

21.4.3.1 AL Control (Address 0x0120:0x0121)

| Parameter | Address | Bit | ZA Drive | ZA ECAT | Description |
|--------------------------------|---------|---------|----------|---------|---|
| Status | 0x120 | 3 to 0 | R/O | W/O | 0x01: Init Request |
| 0x02: PreOperational Request | | | | | |
| 0x03: Bootstrap Mode Request | | | | | |
| 0x04: Safe Operational Request | | | | | |
| 0x08: Operational Request | | | | | |
| Acknowledgement | 0x120 | 4 | R/O | W/O | 0x00: No fault acknowledgement 0x01: Fault acknowledgement (positive edge) |
| Reserved | 0x120 | 7 to 5 | R/O | W/O | - |
| Applic. specific | 0x120 | 15 to 8 | R/O | W/O | - |

21.4.3.2 AL Status (Address 0x0130:0x0131)

| Parameter | Address | Bit | ZA Drive | ZA ECAT | Description |
|------------------------|---------|---------|----------|---------|---|
| Status | 0x130 | 3 to 0 | W/O | R/O | 0x01: Init |
| 0x02: PreOperational | | | | | |
| 0x03: Bootstrap Mode | | | | | |
| 0x04: Safe Operational | | | | | |
| 0x08: Operational | | | | | |
| Status change | 0x130 | 4 | W/O | R/O | 0x00: Acknowledgement 0x01: Error, e.g. forbidden transition |
| Reserved | 0x130 | 7 to 5 | W/O | R/O | - |
| Applic. specific | 0x130 | 15 to 8 | W/O | R/O | - |

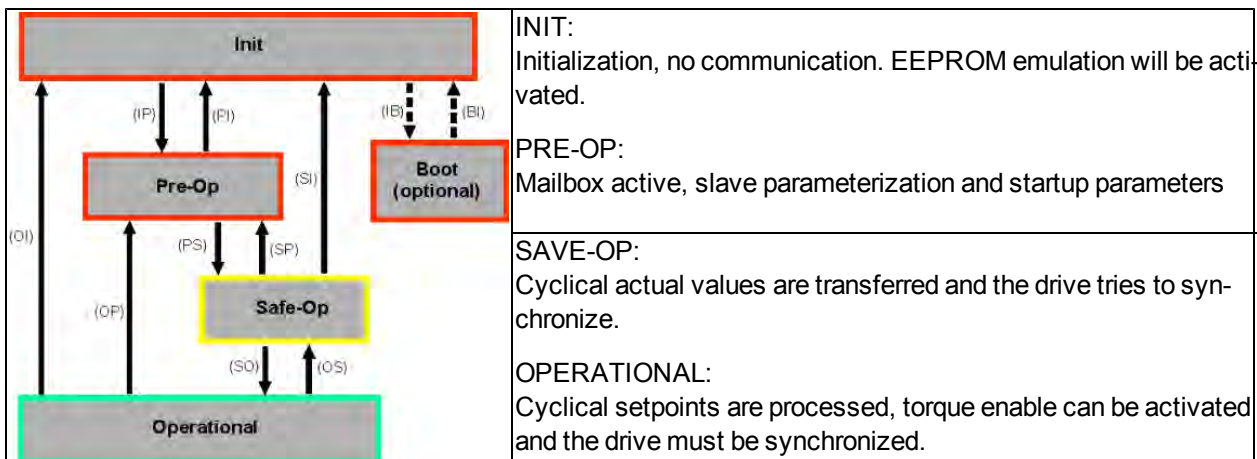
21.4.3.3 AL Status Code (Address 0x0134:0x0135)

| Parameter | Address | Bit | ZA Drive | ZA ECAT | Description |
|-----------|---------|--------|----------|---------|-----------------|
| Status | 0x134 | 7 to 0 | W/O | R/O | See table below |
| Status | 0x135 | 7 to 0 | W/O | R/O | See table below |

| Code | Description | Current Status (Status change) | Resulting Status |
|--------|------------------------------------|---|--------------------|
| 0x0000 | No error | All | Current Status |
| 0x0011 | Invalid requested state change | I -> S, I -> O, P -> O, O -> B, S -> B, P -> B | Current Status + E |
| 0x0017 | Invalid sync manager configuration | I -> P, P -> S | Current Status + E |

No other codes are supported.

21.4.3.4 EtherCAT communication phases

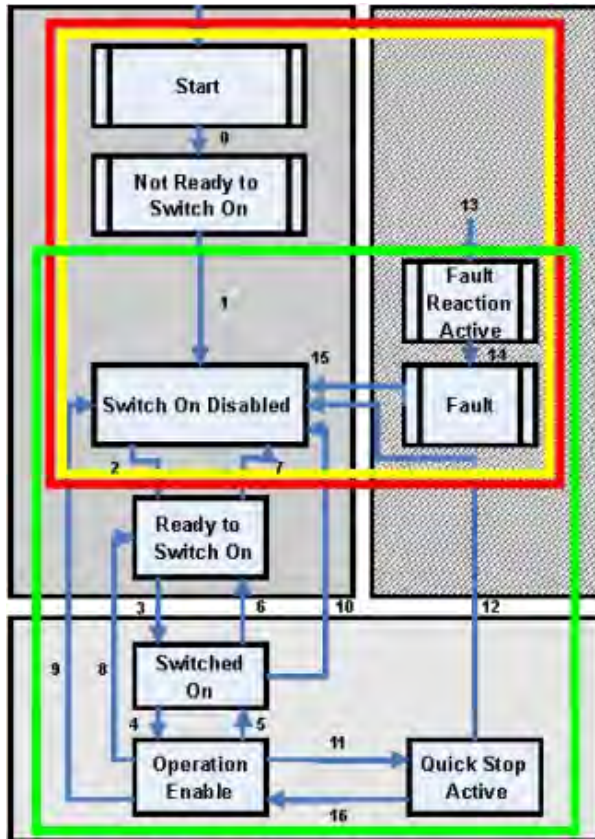


Individual communication transitions

| Transition | AL Control (Bit 3 to 0) | Description |
|------------|-------------------------|--|
| (IB) | 0x03 | - |
| (BI) | - | - |
| (IP) | 0x02 | AKD reads the SyncManager 0 & 1 configuration and verifies the value of the start-address and the length. The AKD prepares itself for handling SyncManager 0 events. |
| (PI) | 0x01 | - |
| (PS) | 0x04 | AKD reads the SyncManager 2 & 3 configuration and verifies the value of the start-address and the length. |
| (SP) | 0x02 | - |
| (SI) | 0x01 | - |
| (SO) | 0x08 | The SyncManager 2 hardware interrupt will be enabled by the drive. |
| (OS) | 0x04 | Deactivation of SyncManager 2 hardware interrupt. |
| (OP) | 0x02 | Deactivation of SyncManager 2 hardware interrupt.. |
| (OI) | 0x01 | Deactivation of SyncManager 2 hardware interrupt. |

21.4.4 CANopen over EtherCAT (CoE) Status Machine

The status machine for the control and status words corresponds to the CANopen status machine in accordance with DS402. CANopen control and status words are captured in every instance of fixed PDO mapping (see chapter entitled 'Fixed PDO Mapping', page).



21.4.4.1 Status Description

| Status | Description |
|------------------------|--|
| Not Ready to Switch On | The drive is not ready to switch on; the controller has not indicated readiness for service. The drive is still in the boot phase or in fault status. |
| Switch On Disable | In 'Switch On Disable' status, the amplifier cannot be enabled via the EtherCAT interface, because (for example) there is no connection to a power source. |
| Ready to Switch On | In 'Ready to Switch On' status, the drive can be enabled via the control word. |
| Switched On | In 'Switched On' status, the amplifier is enabled, but the setpoints of the EtherCAT-interface are not yet transferred. The amplifier is idle, and a positive edge in bit 3 of the control word activates setpoint transfer (transition to 'Operation Enable' status). |
| Operation Enable | In this status, the drive is enabled and setpoints are transferred from the EtherCAT interface. |
| Quick Stop Active | The drive follows a quick stop ramp. |
| Fault Reaction Active | The drive responds to a fault with an emergency stop ramp. |
| Fault | A fault is pending, the drive is stopped and disabled. |

21.4.4.2 Commands in the Control Word

Bit assignment in the control word

| Bit | Name | Bit | Name |
|-----|---|-----|-----------------------|
| 0 | Switch on | 8 | Pause/halt |
| 1 | Disable Voltage | 9 | reserved |
| 2 | Quick Stop | 10 | reserved |
| 3 | Enable Operation | 11 | reserved |
| 4 | Operation mode specific | 12 | reserved |
| 5 | Operation mode specific | 13 | Manufacturer-specific |
| 6 | Operation mode specific | 14 | Manufacturer-specific |
| 7 | Reset Fault (only effective for faults) | 15 | Manufacturer-specific |

Commands in the control word

| Command | Bit 7 Fault Reset | Bit 3 Enable Operation | Bit 2 Quick Stop | Bit 1 Disable Voltage | Bit 0 Switch on | Transitions |
|-------------------|----------------------|---------------------------|---------------------|--------------------------|--------------------|--------------|
| Shutdown | X | X | 1 | 1 | 0 | 2, 6, 8 |
| Switch on | X | X | 1 | 1 | 1 | 3 |
| Disable Voltage | X | X | X | 0 | X | 7, 9, 10, 12 |
| Quick Stop | X | X | 0 | 1 | X | 7, 10, 11 |
| Disable Operation | X | 0 | 1 | 1 | 1 | 5 |
| Enable Operation | X | 1 | 1 | 1 | 1 | 4, 16 |
| Fault Reset | 1 | X | X | X | X | 15 |

Bits labeled **X** are irrelevant. **0** and **1** indicate the status of individual bits.

Mode-dependent bits in the control word

The following table shows the mode-dependent bits in the control word. Only manufacturer-specific modes are supported at present. The individual modes are set by Object 6060h Modes of operation.

| Operation mode | No. | Bit 4 | Bit 5 | Bit 6 |
|----------------------------------|-----|------------------------|------------------------|-------------------|
| Profile Position Mode (pp) | 01h | new_setpoint | change_set_immediately | absolute/relative |
| Profile Velocity Mode (pv) | 03h | reserved | reserved | reserved |
| Profile Torque Mode (tq) | 04h | reserved | reserved | reserved |
| Homing Mode (hm) | 06h | homing_operation_start | reserved | reserved |
| Interpolated Position Mode (ip) | 07h | | reserved | reserved |
| Cyclic synchronous position mode | 08h | reserved | reserved | reserved |

Description of the remaining bits in the control word

Bit 8: (Pause) If Bit 8 is set, then the drive halts (pauses) in all modes. The setpoints (speed for homing or jogging, motion task number, setpoints for digital mode) for the individual modes are retained.

Bit 9,10: These bits are reserved for the drive profile (DS402).

Bit 13, 14, 15: These bits are manufacturer-specific, and reserved at present.

21.4.4.3 Status Machine Bits (status word)

Bit assignment in the status word

| Bit | Name | Bit | Name |
|-----|--------------------|-----|------------------------------------|
| 0 | Ready to switch on | 8 | Manufacturer-specific (reserved) |
| 1 | Switched on | 9 | Remote (always 1) |
| 2 | Operation enable | 10 | Target reached |
| 3 | Fault | 11 | Internal limit active |
| 4 | Voltage enabled | 12 | Operation mode specific (reserved) |
| 5 | Quick stop | 13 | Operation mode specific (reserved) |
| 6 | Switch on disabled | 14 | Manufacturer-specific (reserved) |
| 7 | Warning | 15 | Manufacturer-specific (reserved) |

States of the status machine

| State | Bit 6 switch on disable | Bit 5 quick stop | Bit 3 fault | Bit 2 operation enable | Bit 1 switched on | Bit 0 ready to switch on |
|------------------------|-------------------------------|------------------------|----------------|------------------------------|-------------------------|--------------------------------|
| Not ready to switch on | 0 | X | 0 | 0 | 0 | 0 |
| Switch on disabled | 1 | X | 0 | 0 | 0 | 0 |
| Ready to switch on | 0 | 1 | 0 | 0 | 0 | 1 |
| Switched on | 0 | 1 | 0 | 0 | 1 | 1 |
| Operation enabled | 0 | 1 | 0 | 1 | 1 | 1 |
| Fault | 0 | X | 1 | 0 | 0 | 0 |
| Fault reaction active | 0 | X | 1 | 1 | 1 | 1 |
| Quick stop active | 0 | 0 | 0 | 1 | 1 | 1 |

Bits labeled **X** are irrelevant. **0** and **1** indicate the status of individual bits.

Description of the remaining bits in the status word

Bit 4: `voltage_enabled` The DC-link voltage is present if this bit is set.

Bit 7: `warning` There are several possible reasons for Bit 7 being set and this warning being produced. The reason for this warning can be revealed by using the Object 20subindex manufacturer warnings.

Bit 9: `remote` is always set to 1, i.e. the drive can always communicate and be influenced via the RS232 - interface.

Bit 10: `target_reached` This is set when the drive has reached the target position.

Bit 11: `internal_limit_active` This bit specifies that a movement was or is limited. In different modes, different warnings cause the bit to be set.

21.4.5 Fixed PDO Mappings

Various ready-to-use mappings can be selected for cyclic data exchange via SDO's of the object 0x1C12 and 0x1C13. Using object 0x1C12 subindex 1 (Sync Manager 2 assignment), a fixed mapping for the cyclic command values can be set with the values 0x1701, 0x1702, 0x1703, 0x1720, 0x1721 and 0x1723. Using object 0x1C13 subindex 1 (Sync Manager 3 assignment), a fixed mapping for the cyclic actual values can be set via the data 0x1B01, 0x1B02, 0x1B20 and 0x1B21.

The following sequence describes how to select the fixed command value mapping 0x1701 via SDO's:

1. SDO write access to object 0x1C12Sub0 Data:0x00
2. SDO write access to object 0x1C12Sub1 Data:0x1701
3. SDO write access to object 0x1C12Sub0 Data:0x01

The following fixed mappings are supported:

Position interface:

| | |
|--------|--|
| 0x1701 | Position command value (4 bytes), Control word (2 bytes), total (6 bytes) |
| 0x1720 | Control Word (2 bytes), Interpolated position command value (4 bytes), Latch control word (2 bytes), Torque feed forward (2 bytes), Digital outputs (2 bytes) |
| 0x1721 | Interpolated position command value (4 bytes), Control Word (2 bytes), Torque feed forward (2 bytes) |
| 0x1723 | Control word (2 bytes), Ineterpolated position command value (4 bytes), Latch control word (2 bytes), Torque feed forward (2 bytes), Digital outputs (2 bytes), Reset of changed input information (2 bytes) |
| 0x1B01 | Position actual value (4 bytes), Status word (2 bytes), total (6 bytes) |
| 0x1B21 | Position Actual Internal Value (4 bytes), Status word (2 bytes) |

Velocity interface:

| | |
|--------|---|
| 0x1702 | Velocity command value (4 bytes), Control word (2 bytes), total (6 bytes) |
| 0x1B02 | Position actual value (4 bytes), Status word (2 bytes), total (6 bytes) |

Torque interface:

| | |
|--------|--|
| 0x1703 | Torque command value (2 bytes), Control word (2 bytes) |
|--------|--|

21.4.6 Supported Cyclical Setpoint and Actual Values

Supported cyclical setpoint values

| Name | CANopen object number | Data type | Description |
|------------------------|-----------------------|-----------|--------------------------------------|
| Position command value | 0x60C1 sub 1 | INT32 | Interpolation data record in IP-mode |
| Velocity command value | 0x60FF sub 0 | INT32 | |
| CANopen control-word | 0x6040 sub 0 | UINT16 | CANopen control word. |
| Latch Control word | 0x20a4 sub 0 | UINT16 | |
| Torque feed forward | 0x60B2 sub 0 | INT16 | |
| Digital outputs | 0x60FE sub 1 | UINT32 | |

Supported cyclical actual values

| Name | CANopen object number | Data type | Description |
|--------------------------------|-----------------------|-----------|----------------------|
| Position actual internal value | 0x6063 sub 0 | INT32 | |
| Velocity actual value | 0x606c sub 0 | INT32 | |
| CANopen status-word | 0x6041 sub 0 | UINT16 | CANopen status word. |
| Second position feedback | 2050 sub 0 | INT32 | |
| Digital inputs | 60FD sub 0 | UINT32 | |
| Following error actual value | 60F4 sub 0 | INT32 | |
| Latch position positive edge | 20a0 sub 0 | INT32 | |
| Torque actual value | 6077 sub 0 | INT16 | |
| Latch status | 20A5 sub 0 | UINT16 | |
| Analog input value | 3470 sub 0 | INT16 | |

21.4.7 Supported Operation Modes

| CANopen mode of operation | AKD mode of operation | Description |
|---------------------------|---------------------------------|--|
| Profile velocity | DRV.OPMODE 2 DRC.CMDSOURCE 1 | 0x6060Sub0 Data: 3 In this mode of operation the EtherCAT master sends cyclic velocity command values to the AKD. |
| Interpolated position | DRV.OPMODE 2 DRV.CMDSOURCE 1 | 0x6060Sub0 Data: 7 In this mode of operation the EtherCAT master sends cyclic position command values to the AKD. These command values are interpolated by the AKD according to the fieldbus sample rate. |
| Homing mode | DRV.OPMODE 2 DRV.CMDSOURCE 0 | 0x6060 sub 0 data : 6 In this mode an AKD-internal homing can be done. |

21.4.8 Adjusting EtherCAT Cycle Time

The cycle time to be used in the drive for the cyclical setpoints and actual values can either be stored in the FBUS.SAMPLEPERIOD parameter in the amplifier or configured in the startup phase.

This takes place via SDO mailbox access to CANopen objects 60C2 subindex 1 and 2.

Subindex 2, known as the interpolation time index, defines the power of ten of the time value (e.g. -3 means 10⁻³ or milliseconds) while subindex 1, known as interpolation time units, gives the number of units (e.g. 4 means 4 units).

You can run a 2 ms cycle using various combinations. For example,

Index = -3, Units = 2

or

Index = -4, Units = 20 etc.

The FBUS.SAMPLEPERIOD parameter is counted in multiples of 62.5us microseconds within the device. This means, for example that 2 ms equates to FBUS.SAMPLEPERIOD value of 32.

21.4.9 Maximum Cycle Times depending on operation mode

The minimum cycle time for the drive is largely dependent on the drive configuration (second actual position value encoder latch functionality enabled and so on)

| Interface | Cycle time AKD |
|-----------|----------------------|
| Position | ≥ 0.25 ms (≥ 250 μs) |
| Velocity | ≥ 0.25 ms (≥ 250 μs) |
| Torque | ≥ 0.25 ms (≥ 250 μs) |

21.4.10 Synchronization

On all drives, the internal PLL is theoretically able to even out an average deviation of up to 4800 ppm in the cycle time provided by the master. The drive checks once per fieldbus cycle a counter within the drive internal FPGA, which is cleared by a Sync0 (Distributed clock) event. Depending of the counter value, the drive extends or decreases the 62.5 µs MTS signal within the drive by a maximum of 300 ns.

The theoretical maximum allowed deviation can be calculated by using the following formula:

$$\max_{\text{dev}} = \frac{300[\text{ns}]}{62.5[\mu\text{s}]} \cdot 1,000,000 = 4800 [\text{ppm}]$$

The synchronization functionality within the drive can be enabled via setting bit 0 of the FBUS.PARAM02 parameter to high. Therefore FBUS.PARAM02 must be set to the value of 1. Furthermore the distributed clock functionality must be enabled by the EtherCAT master in order to activate cyclic Sync0 events.

21.4.10.1 Synchronization behavior with distributed clocks (DC) enabled

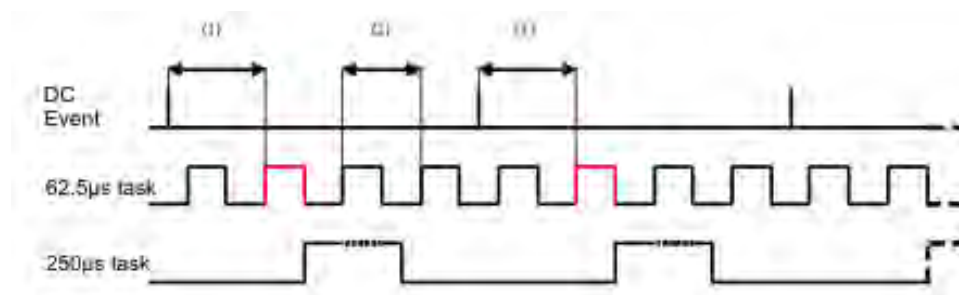
When the EtherCAT master enables distributed clocks, a distributed clock (DC) event is created in the AKD once per fieldbus cycle. An assigned 62.5 µs real-time task in the AKD monitors the elapsed time between the DC events and the AKD System time and extends or reduces the 62.5 µs strobe to the CPU as necessary.

The following fieldbus parameters are used for the synchronization feature:

1. FBUS.SYNCDIST = Expected time delay of the AKD PLL-code to the DC event.
2. FBUS.SYNCACT = Actual time delay of the AKD PLL-code to the DC event.
3. FBUS.PLLTHRESH = Number of consecutive successful synchronized PLL cycles of the AKD before the Drive is considered as synchronized.
4. FBUS.SYNCWND = Synchronization window in which the AKD is considered to be synchronized. The Drive is considered synchronized as long as the following statement is true is true for FBUS.PLLTHRESH consecutive cycles:

$$\text{FBUS.SYNCDIST} - \text{FBUS.SYNCWND} < \text{FBUS.SYNCACT} < \text{FBUS.SYNCDIST} + \text{FBUS.SYNCWND}$$

Example with a 4kHz fieldbus sample rate:



Explanation:

The red-marked 62.5[µs] real-time task displays the AKD 62.5 µs real-time task within one fieldbus cycle which is responsible for calling the AKD PLL-code. The time delay (1) shows the actual delay to the previous DC event, which is ideally close to the adjusted FBUS.SYNCDIST parameter. Depending on (1) the AKD slightly extends or reduce the 62.5[µs] IRQ generation of the high-priority real-time task in order to either increase or decrease the measured time delay to the DC event (1) for the next PLL cycle. The time distance (2) shows the 62.5[µs] ± x[ms] realtime task of the AKD.

21.4.10.2 Synchronization behavior with distributed clocks (DC) disabled

The AKD fieldbus synchronization algorithm is similar to that used by Distributed Clocks. The difference is that the AKD synchronizes to a SyncManager2 event instead of the DC event. A SyncManager2 event is created when the EtherCAT Master sends a new package of command values to the drive while the network is in the Operational state. This occurs once per fieldbus cycle.

21.4.11 Latch Control Word and Latch Status Word

Latch Control word (2 Byte)

| Bit | Value (bin) | Value (hex) | Description |
|-------|-------------------|-------------|---------------------------------------|
| 0 | 00000000 00000001 | zz01 | Enable extern latch 1 (positive rise) |
| 1 | 00000000 00000010 | zz02 | Enable extern latch 1 (negative rise) |
| 2 | 00000000 00000100 | zz04 | Enable extern latch 2 (positive rise) |
| 3 | 00000000 00001000 | zz08 | Enable extern latch 2 (negative rise) |
| 4 | | | |
| 5-7 | | | Reserve |
| 8-12 | 00000001 00000000 | 01zz | Read external latch 1 (positive rise) |
| | 00000010 00000000 | 02zz | Read external latch 1 (negative rise) |
| | 00000011 00000000 | 03zz | Read external latch 2 (positive rise) |
| | 00000100 00000000 | 04zz | Read external latch 2 (negative rise) |
| 13-15 | | | Reserve |

Latch Status word (2 Byte)

| Bit | Value (bin) | Value (hex) | Description |
|-------|-------------------|-------------|--|
| 0 | 00000000 00000001 | zz01 | External latch 1 valid (positive rise) |
| 1 | 00000000 00000010 | zz02 | External latch 1 valid (negative rise) |
| 2 | 00000000 00000100 | zz04 | External latch 2 valid (positive rise) |
| 3 | 00000000 00001000 | zz08 | External latch 2 valid (negative rise) |
| 4 | | | |
| 5-7 | | | Reserve |
| 8-11 | 00000001 00000000 | z1zz | Acknowledge value external latch 1 (positive rise) |
| | 00000010 00000000 | z2zz | Acknowledge value external latch 1 (negative rise) |
| | 00000011 00000000 | z3zz | Acknowledge value external latch 2 (positive rise) |
| | 00000100 00000000 | z4zz | Acknowledge value external latch 2 (negative rise) |
| 12-15 | 00010000 00000000 | 1zzz | Zustand Digital Input 4 |
| | 00100000 00000000 | 2zzz | Zustand Digital Input 3 |
| | 01000000 00000000 | 4zzz | Zustand Digital Input 2 |
| | 10000000 00000000 | 8zzz | Zustand Digital Input 1 |

21.4.12 Mailbox Handling

With EtherCAT, acyclical data traffic (object channel or SDO channel) is called mailbox. This system is based around the master:

Mailbox Output:

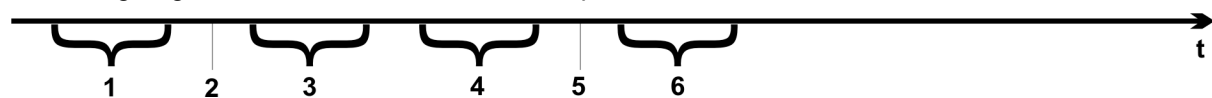
The master (EtherCAT controller) sends data to the slave (drive). This is essentially a (read/write) request from the master. Mailbox output operates via Sync Manager 0.

Mailbox Input:

The slave (drive) sends data to the master (EtherCAT controller). The master reads the slave's response. Mailbox input operates via Sync Manager 1.

Timing diagram

The timing diagram illustrates the mailbox access process:



1. The EtherCAT master writes the mailbox request to the mail-out buffer.
2. On the next interrupt, the EtherCAT interface activates a Sync Manager 0 event (mailbox output event) in the AL event register.
3. The drive reads 16 bytes from the mail-out buffer and copies them to the internal mailbox output array.
4. The drive identifies new data in the internal mailbox output array and performs an SDO access to the object requested by the EtherCAT interface. The response from the drive is written to an internal mailbox input array.
5. The drive deletes all data in the internal mailbox output array so that a new mailbox access attempt can be made.
6. The drive copies the response telegram from the internal mailbox input array to the mail-in buffer of the EtherCAT interface.

21.4.12.1 Mailbox Output

An interrupt by the EtherCAT-interface with a Sync Manager 0 - Event starts a Mailbox Output Process. A 1 in the Mail Out Event-Bit of the AL Event register signalizes the drive, that the EtherCAT-interface wants to send a Mailbox message and that it has already stored the required data in the Mail Out Buffer. Now 16 Byte data are read by the drive with the IRQ process. The bytes are defined as follows

| Address 0x1800 | | | | | | | | Address 0x180F | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| CAN over EtherCAT specific data (CoE Header) | | | | | | | | CAN specific data (standard CAN SDO) | | | | | | | |
| Byte 0 | Length of the data (Low Byte) | | | | | | | | | | | | | | |
| Byte 1 | Length of the data (High Byte) | | | | | | | | | | | | | | |
| Byte 2 | Address (Low Byte) | | | | | | | | | | | | | | |
| Byte 3 | Address (High Byte) | | | | | | | | | | | | | | |
| Byte 4 | Bit 0 to 5: Channel Bit 6 to 7: Priority | | | | | | | | | | | | | | |
| Byte 5 | Bit 0 to 3: Type | | | 1 = Reserved: ADS over EtherCAT 2 = Reserved: Ethernet over EtherCAT 3 = Can over EtherCAT... | | | | | | | | | | | |
| | Bit 4 to 7: Reserved | | | | | | | | | | | | | | |
| Byte 6 | PDO Number (with PDO transmissions only, Bit 0 = LSB of the PDO number, see Byte 7 for MSB) | | | | | | | | | | | | | | |
| Byte 7 | Bit 0: MSB of the PDO number, see Byte 6 | | | | | | | | | | | | | | |
| | Bit 1 to 3: Reserved | | | | | | | | | | | | | | |
| | Bit 4 to 7: CoE specific type | | | 0: Reserved | | | | | | | | | | | |
| | | | | 1: Emergency message | | | | | | | | | | | |
| | | | | 2: SDO request | | | | | | | | | | | |
| | | | | 3: SDO answer | | | | | | | | | | | |
| | | | | 4: TXPDO | | | | | | | | | | | |
| | | | | 5: RxPDO | | | | | | | | | | | |
| | | | | 6: Remote transmission request of a TxPDO | | | | | | | | | | | |
| | | | | 7: Remote transmission request of a RxPDO | | | | | | | | | | | |
| | | | | 8... 15: reserved | | | | | | | | | | | |
| Byte 8 | Control-Byte in the CAN telegram: | | | | | | | | | | | | | | |
| | write access: | | | 0x23=4Byte, 0x27=3Byte, 0x2B=2Byte, 0x2F=1Byte | | | | | | | | | | | |
| | read access: | | | 0x40 | | | | | | | | | | | |
| Byte 9 | Low Byte of the CAN object number (Index) | | | | | | | | | | | | | | |
| Byte 10 | High Byte of the CAN object number (Index) | | | | | | | | | | | | | | |
| Byte 11 | Subindex according to CANopen Specification for the drive | | | | | | | | | | | | | | |
| Byte 12 | Data with a write access (Low Byte) | | | | | | | | | | | | | | |
| Byte 13 | Data with a write access | | | | | | | | | | | | | | |
| Byte 14 | Data with a write access | | | | | | | | | | | | | | |
| Byte 15 | Data with a write access (High Byte) | | | | | | | | | | | | | | |

The drive answers every telegram with an answer in the Mailbox Input buffer.

21.4.12.2 Mailbox Input

The drive answers every CoE telegram with a 16 byte answer telegram in the Mailbox Input buffer. The bytes are defined as follows:

| Address 0x1C00 | | | | | | | | Address 0x1C0F | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| CAN over EtherCAT specific data (CoE Header) | | | | | | | | CAN specific data (standard CAN SDO) | | | | | | | |
| Byte 0 | Length of the data (Low Byte) | | | | | | | | | | | | | | |
| Byte 1 | Length of the data (High Byte) | | | | | | | | | | | | | | |
| Byte 2 | Address (Low Byte) | | | | | | | | | | | | | | |
| Byte 3 | Address (High Byte) | | | | | | | | | | | | | | |
| Byte 4 | Bit 0 to 5: Channel Bit 6 to 7: Priority | | | | | | | | | | | | | | |
| Byte 5 | Bit 0 to 3: Type | | | | 1 = Reserved: ADS over EtherCAT 2 = Reserved: Ethernet over EtherCAT 3 = Can over EtherCAT... | | | | | | | | | | |
| | Bit 4 to 7: Reserved | | | | | | | | | | | | | | |
| Byte 6 | PDO Number (with PDO transmissions only, Bit 0 = LSB of the PDO number, see Byte 7 for MSB) | | | | | | | | | | | | | | |
| Byte 7 | Bit 0: MSB of the PDO number, see Byte 6 | | | | | | | | | | | | | | |
| | Bit 1 to 3: Reserved | | | | | | | | | | | | | | |
| | Bit 4 to 7: CoE specific type | | | | 0: Reserved | | | | | | | | | | |
| | | | | | 1: Emergency message | | | | | | | | | | |
| | | | | | 2: SDO request | | | | | | | | | | |
| | | | | | 3: SDO answer | | | | | | | | | | |
| | | | | | 4: TXPDO | | | | | | | | | | |
| | | | | | 5: RxPDO | | | | | | | | | | |
| | | | | | 6: Remote transmission request of a TxPDO | | | | | | | | | | |
| | | | | | 7: Remote transmission request of a RxPDO | | | | | | | | | | |
| | | | | | 8...15: reserved | | | | | | | | | | |
| Byte 8 | Control-Byte in the CAN telegram: | | | | | | | | | | | | | | |
| | write access OK: | | | | 0x60 | | | | | | | | | | |
| | read access OK + length of answer: | | | | 0x43 (4 Byte), 0x47 (3 Byte), 0x4B (2Byte), 0x4F (1Byte) | | | | | | | | | | |
| | error with read- or write access: | | | | 0x80 | | | | | | | | | | |
| Byte 9 | Low Byte of the CAN object number (Index) | | | | | | | | | | | | | | |
| Byte 10 | High Byte of the CAN object number (Index) | | | | | | | | | | | | | | |
| Byte 11 | Subindex according to CANopen Specification for Kollmorgen drive | | | | | | | | | | | | | | |
| Byte 12 | Data (Low Byte) | | | | | | | | | | | | | | |
| Byte 13 | Data | | | | error code Fehlercode according to CANopen Specification in case of an error | | | | | | | | | | |
| Byte 14 | Data | | | | data value of the object in case of successfull read access | | | | | | | | | | |
| Byte 15 | Data (High Byte) | | | | | | | | | | | | | | |

21.4.12.3 Example: Mailbox Access

In the example below, PDOs 0x1704 are mapped (see Chapter # “Fixed PDO Mappings”):

The master sends this mailbox output message:

| | | |
|----------------|------|--|
| Byte 0 | 0x0A | The next 10 Bytes contain data (Byte 2 to Byte 11) |
| Byte 1 | 0x00 | The next 10 Bytes contain data (Byte 2 to Byte 11) |
| Byte 2 | 0x00 | Address 0 |
| Byte 3 | 0x00 | Address 0 |
| Byte 4 | 0x00 | Channel 0 and Priority 0 |
| Byte 5 | 0x03 | CoE Object |
| Byte 6 | 0x00 | PDO Number 0 |
| Byte 7 | 0x20 | PDO Number 0 and SDO-Request |
| Byte 8 | 0x2B | 2 Byte write access |
| Byte 9 | 0x12 | SDO-Object 0x1C12 |
| Byte 10 | 0x1C | SDO-Object 0x1C12 |
| Byte 11 | 0x01 | Subindex 1 |
| Byte 12 | 0x04 | Data value 0x00001704 |
| Byte 13 | 0x17 | Data value 0x00001704 |
| Byte 14 | 0x00 | Data value 0x00001704 |
| Byte 15 | 0x00 | Data value 0x00001704 |

The drive returns the following message:

| | | |
|----------------|------|--|
| Byte 0 | 0x0E | The next 14 Bytes contain data (Byte 2 to Byte 15) |
| Byte 1 | 0x00 | The next 14 Bytes contain data (Byte 2 to Byte 15) |
| Byte 2 | 0x00 | Address 0 |
| Byte 3 | 0x00 | Address 0 |
| Byte 4 | 0x00 | Channel 0 and Priority 0 |
| Byte 5 | 0x03 | CoE Object |
| Byte 6 | 0x00 | PDO Number 0 |
| Byte 7 | 0x20 | PDO Number 0 and SDO-Answer |
| Byte 8 | 0x60 | Successful write access |
| Byte 9 | 0x12 | SDO-Object 0x1C12 |
| Byte 10 | 0x1C | SDO-Object 0x1C12 |
| Byte 11 | 0x01 | Subindex 1 |
| Byte 12 | 0x00 | Data value 0x00000000 |
| Byte 13 | 0x00 | Data value 0x00000000 |
| Byte 14 | 0x00 | Data value 0x00000000 |
| Byte 15 | 0x00 | Data value 0x00000000 |

21.4.13 Fieldbus Parameters

The AKD holds several fieldbus-specific, general purpose parameters. Some of them contain the following EtherCAT relevant data:

- **FBUS.PARAM02:** This parameter activates the synchronization feature of the AKD. The DC feature must be activated in order to allow the AKD to get synchronized with the master. A value of 1 enables the drive internal PLL functionality, a value of 0 deactivates this feature.
- **FBUS.PARAM03:** This parameter contains the Configured Station Alias address of the AKD. An EEPROM emulation write access to the Configured Station Alias address forces the AKD to store the drive parameters automatically using the DRV.NVSAVE command.
- **FBUS.PARAM04:** This parameter enables (1) or disables(0) the synchronization supervision of the CAN-Open or EtherCAT fieldbus.

Default values for this parameter are as follows:

CANopen drive: disabled (0)

EtherCAT drive: enabled (1)

Synchronization supervision is active when FBUS.PARAM 04 = 1 and the first CANOpen Sync message or first EtherCAT frame is received. When more than three CANOpen sync messages or seven EtherCAT frames have not been received and the drive is enabled, fault F125 (“Synchronization lost“), occurs.

This page intentionally left blank.

21.5 Index

1

| | |
|------------|-----|
| 1000h | 163 |
| 1001h | 164 |
| 1002h | 165 |
| 1003h | 166 |
| 1005h | 167 |
| 1006h | 167 |
| 1008h | 168 |
| 1009h | 168 |
| 100Ah | 168 |
| 100Ch | 169 |
| 100Dh | 169 |
| 1010h | 170 |
| 1014h | 171 |
| 1016h | 171 |
| 1017h | 172 |
| 1018h | 172 |
| 1026h | 174 |
| 1400-1403h | 185 |
| 1600-1603h | 186 |
| 1800-1803h | 188 |
| 1A00-1A03h | 190 |

2

| | |
|------------|-----|
| 2014-2017h | 175 |
| 2018h | 176 |
| 2026h | 177 |
| 204Ch | 200 |
| 20A0h | 178 |
| 20A1h | 178 |
| 20A2h | 178 |
| 20A3h | 179 |
| 20A4h | 179 |
| 20A5h | 180 |
| 20A6h | 180 |
| 20B8h | 181 |

6

| | |
|-------|-----|
| 6040h | 195 |
| 6041h | 196 |
| 6060h | 198 |
| 6061h | 199 |
| 6063h | 207 |
| 6064h | 208 |
| 6065h | 208 |
| 606Ch | 203 |
| 6071h | 205 |
| 6073h | 205 |
| 6077h | 206 |
| 607Ah | 219 |
| 607Ch | 214 |

| | |
|-------|-----|
| 607Dh | 219 |
| 6081h | 220 |
| 6083h | 220 |
| 6084h | 220 |
| 608Fh | 201 |
| 6092h | 202 |
| 6098h | 214 |
| 6099h | 216 |
| 609Ah | 216 |
| 60C0h | 209 |
| 60C1h | 210 |
| 60C2h | 211 |
| 60C4h | 212 |
| 60F4h | 208 |
| 60FDh | 182 |
| 60FEh | 182 |
| 60FFh | 204 |
| 6502h | 183 |

A

| | |
|----------------------------|---------|
| Abbreviations | 60, 137 |
| Acceleration ramp | 359 |
| AL Event | 80 |
| Analog inputs | 275 |
| Analog output | 103 |
| Analog setpoints | 275 |
| Anti-resonance filter | 662 |
| Aux. supply 24V, interface | 267 |

B

| | |
|-----------------------------|-----|
| Basic data types | 147 |
| Basic drive setup | 34 |
| Basic testing | 236 |
| Blink | 362 |
| Block diagrams | |
| current loop | 282 |
| position and velocity loops | 282 |
| Brake view | 80 |

C

| | |
|-----------------------------------|--------|
| CANopen over EtherCAT | 84 |
| Capture | 81 |
| Check communications | 40, 43 |
| Command buffer | 96 |
| Command source | 366 |
| Communication Objects | 149 |
| Connect to another drive | 40 |
| Connected and disconnected states | 38 |
| Connection diagram | |
| AKD-x00306 to x00606 | 264 |
| AKD-x01206 | 265 |
| AKD-x01206 & AKD-xzzz07 | 266 |
| Connection Diagrams | 263 |
| Control word | 195 |

| | | | |
|--|-----------------|---|-----------------|
| Control Word Commands | 85 | Foldback | 77 |
| Controlled Stop | 115 | G | |
| current loop | 121 | General definitions | 163 |
| Current loop gain scheduling | 122 | General objects | 163 |
| Cycle Time | | H | |
| Adjust | 89 | Heartbeat | 158 |
| Max.Values | 89 | Homing | 134 |
| Cyclical Values | 88 | Homing Mode | 214 |
| D | | I | |
| Data types | 147 | I/O-Connection | 104, 273 |
| DC bus link, interface | 270 | Inputs | |
| deceleration | 369 | Analog | 275 |
| Device control | 192 | Digital | 99 |
| Device Not Shown | 38, 46 | Enable | 102 |
| Dig. Encoder emulation, interface | 277 | Programmable | 102 |
| Digital inputs | 99 | Interpolated position mode | 209 |
| Digital Inputs and Outputs | 88 | Interrupt Event | 80 |
| Direct DC mains | 56 | Invalid Firmware | 259 |
| Disable | 373 | J | |
| Disconnected state | 38 | Jog Move | 153 |
| Display Codes | 34 | K | |
| Downloading Firmware | 259 | KAS IDE | 77 |
| Drive Motion Status | 153 | L | |
| Dynamic Braking | 117 | Latch Words | 91 |
| E | | M | |
| Electronic gearing | 106 | Macros | 219 |
| Emergency Messages | 253, 160 | Mailbox | 92 |
| Emergency Object | 151 | Mains supply, interface | 271 |
| emergency timeout | 375 | Manufacturer specific objects | 175 |
| Emulated Encoder Output | 277 | master/slave | 74 |
| ENABLE | 102 | Mixed data types | 148 |
| Enable/Disable | 111 | Modbus | 6 |
| encoder emulation | 74 | Modbus Dynamic Mapping | 9 |
| Error Messages | | Modulo position | 127 |
| parameter and command | 250 | Motion Tasks | 144 |
| EtherCAT onboard | 65 | Motor | 64 |
| EtherCAT Profile | 78 | Motor interface | 268 |
| Examples | 236 | N | |
| Examples, setup | 236 | Network Management Object | 150 |
| Examples, special applications | 249 | Nodeguard | 157 |
| Extended data types | 149 | Non-Plug and Play Feedback Devices | 76 |
| F | | | |
| Factor Groups | 199 | | |
| Fault messages | 240 | | |
| Faults | 364 | | |
| clearing | 249 | | |
| Feedback 1 | 66 | | |
| Feedback 2 | 73 | | |
| Feedback Connection | 279 | | |
| Fieldbus | 96 | | |
| Fieldbus Parameters | 96 | | |

O

| | |
|-----------------------------------|------------|
| Object Dictionary | 223 |
| Operating mode | 198 |
| parameters and commands active in | 225, |
| | 265 |
| Operation Modes | 88 |
| Outputs | |
| Analog | 274 |
| Overview | 50 |

P

| | |
|-----------------------------------|----------------|
| Parameter Comparer | 221 |
| Parameter Load/Save | 221 |
| PDO configuration | 184 |
| PDO Mapping | 87 |
| Performance Servo Tuner | 160 |
| advanced | 164 |
| Phase run-up | 82 |
| Position Control Function | 207 |
| Position Loop | 126 |
| Power | 56 |
| Process Data Object | 155 |
| Profile Position Mode | 218 |
| Profile specific objects | 182 |
| Profile torque mode | 205 |
| Profile Velocity Mode | 203 |
| Programmable Limit Switch | 109 |
| Prohibited Use | 62, 139 |
| Pulse Direction, interface | 278 |

R

| | |
|----------------------------------|------------|
| Receive PDOs | 185 |
| Regen resistor, interface | 269 |
| Regeneration | 58 |
| Registration moves | 149 |
| Response monitoring | 169 |

S

| | |
|----------------------------------|------------|
| Safe Torque Off (STO) | 118 |
| Safety Instructions | |
| Electrical Installation | 64, 141 |
| General | 62, 139 |
| Save On Disconnect | 157 |
| Save On Exit | 156 |
| Save On Firmware Download | 158 |
| Save Options | 156 |
| scope | 205 |
| channels | 205 |
| settings and presets | 212 |
| time-base | 205 |
| SDO abort codes | 154 |
| Service Data Object | 152 |
| Service Motion | 152 |

| | |
|-------------------------------|------------|
| Settings | 52 |
| Setup | 66 |
| Signed Integer | 148 |
| Slave Register | 79 |
| Slider Tuning | 160 |
| Status machine | 192 |
| Status Machine | 84 |
| Status word | 196 |
| Status Word | 86 |
| Synchronization | 90 |
| Synchronization Object | 150 |

T

| | |
|---------------------------|----------------|
| Target group | 59, 136 |
| TCP/IP | 48 |
| Terminal | 218 |
| Time Stamp Object | 150 |
| Transmission modes | 156 |
| Transmit PDOs | 188 |
| Trigger modes | 156 |
| Trouble Shooting | 257 |
| Tuning | 189 |
| TwinCAT | 66 |

U

| | |
|-------------------------|----------------|
| Units | 84 |
| Unsigned Integer | 147 |
| Up/Down Input | 278 |
| Use as directed | 62, 139 |

V

| | |
|----------------------|------------|
| Velocity Loop | 124 |
|----------------------|------------|

W

| | |
|-------------------------------|------------|
| Wake and shake (WS) | 68 |
| Warnings | 240 |
| Watch | 52 |
| Wizard | 35 |
| Workbench over TwinCAT | 71 |

This page intentionally left blank.

This page intentionally left blank.

Sales and Service

We are committed to quality customer service. In order to serve in the most effective way, please contact your local sales representative for assistance.

If you are unaware of your local sales representative, please contact us.

Europe

Kollmorgen Customer Support Europe

Internet www.kollmorgen.com
E-Mail technik@kollmorgen.com
Tel.: +49(0)2102 - 9394 - 0
Fax: +49(0)2102 - 9394 - 3155

North America

Kollmorgen Customer Support North America

Internet www.kollmorgen.com
E-Mail support@kollmorgen.com
Tel.: +1 - 540 - 633 - 3545
Fax: +1 - 540 - 639 - 4162



22 CANopen Manual

AKD™

CANopen Communication



Edition: Revision C, April 2011

Valid for Hardware Revision C

Part Number 903-200004-00

Original Documentation

CANopen®

Keep all manuals as a product component during the life span of the product. Pass all manuals to future users and owners of the product.

KOLLMORGEN®

Record of Document Revisions:

| Revision | Remarks |
|-------------|--|
| - ,11/2009 | Beta launch version |
| - , 12/2009 | Objects 2018&60FE added, Object dictionary, formatting updated |
| A, 04/2010 | Termination connector "optional", several new objects, Object dictionary split |
| B, 07/2010 | Part number added, several new objects, object dictionary expanded |
| C, 01/2011 | HW Rev. C, new objects, object dictionary expanded |
| D, 04/2011 | Object dictionary updated, baudrate setup |

Hardware Revision (HR)

| Hardware Revision | Firmware | WorkBench | Remarks |
|-------------------|------------------|---------------|----------------|
| A | M_01-03-zz-zzz | 1.3.0.zzzzz | Start revision |
| C | ≥ M_01-03-00-011 | ≥ 1.3.0.zzzzz | STO certified |
| | | | |

EnDat is a registered trademark of Dr. Johannes Heidenhain GmbH

EtherCAT is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH

HIPERFACE is a registered trademark of Max Stegmann GmbH

WINDOWS is a registered trademark of Microsoft Corporation

AKD is a registered trademark of Kollmorgen Corporation

Current patents:

US Patent 5,646,496 (used in control card R/D and 1 Vp-p feedback interface)

US Patent 5,162,798 (used in control card R/D)

US Patent 6,118,241 (used in control card simple dynamic braking)

Technical changes which improve the performance of the device may be made without prior notice!

Printed in the United States of America

This document is the intellectual property of Kollmorgen. All rights reserved. No part of this work may be reproduced in any form (by photocopying, microfilm or any other method) or stored, processed, copied or distributed by electronic means without the written permission of Kollmorgen.

Table of Contents

| | | |
|----------|---|-----------|
| 1 | About this User Guide | 29 |
| 1.1 | About this User Guide | 29 |
| 1.2 | Abbreviations | 29 |
| 2 | AKD Models | 31 |
| 2.1 | CC Drive Models | 31 |
| 3 | Basic Drive Setup | 33 |
| 3.1 | Basic Drive Setup | 34 |
| 3.2 | Display Codes | 34 |
| 3.3 | AKD Setup Wizard | 35 |
| 4 | Connecting the Drive | 37 |
| 4.1 | Connected and Disconnected States | 38 |
| 4.2 | Disconnected | 38 |
| 4.3 | Device Not Shown | 38 |
| 4.4 | Find and Enter IP Address | 39 |
| 4.5 | Check Communications | 40 |
| 4.6 | Connect To Another Drive | 40 |
| 5 | Communicating with the Drive | 41 |
| 5.0.1 | Overview | 42 |
| 5.0.2 | Identifying the Drive IP Address | 42 |
| 5.0.2.1 | Automatic (Dynamic) IP Addressing | 42 |
| 5.0.2.2 | Static IP Addressing | 42 |
| 5.1 | Check Communications | 43 |
| 5.2 | Troubleshooting Connection and Communication Problems | 45 |
| 5.2.1 | No Drives are Shown | 46 |
| 5.2.2 | Device Not Shown | 46 |
| 5.2.3 | Find and Enter IP Address | 46 |
| 5.3 | TCP/IP Communication Protocols | 48 |
| 5.4 | IP Address | 48 |
| 5.4.1 | Port Number | 48 |
| 5.4.2 | MAC Address | 48 |
| 6 | Using WorkBench | 49 |
| 6.1 | Welcome Screen | 49 |
| 6.1.1 | Online | 49 |
| 6.1.2 | Offline | 50 |
| 6.2 | AKD Overview | 50 |
| 6.3 | Online and Offline | 51 |
| 6.3.1 | Online Drive | 51 |
| 6.3.2 | Offline Drive | 51 |
| 6.3.3 | Switching Between Online and Offline | 52 |
| 6.4 | Watch | 52 |
| 6.5 | Settings | 52 |
| 6.5.1 | Navigation Tree | 52 |

| | |
|--|-----------|
| 6.5.2 Settings View..... | 52 |
| 7 Configuring Drive Power..... | 55 |
| 7.1 Power..... | 56 |
| 7.1.1 Drive Setup for Power and Bus..... | 56 |
| 7.1.2 Direct DC Mains Operation..... | 56 |
| 7.2 Regeneration..... | 58 |
| 7.2.1 Overview..... | 58 |
| 7.2.2 Regen Resistor Options..... | 58 |
| 7.2.3 Calculating Motor Peak Energy and Regen Resistor Size..... | 58 |
| 7.2.4 Selecting a Compatible Regen Resistor..... | 61 |
| 7.2.5 Configuring Regen Parameter Values..... | 61 |
| 7.2.6 Related Parameters..... | 62 |
| 8 Configuring Motor Settings..... | 63 |
| 8.1 Motor..... | 64 |
| 8.1.1 Overview..... | 64 |
| 8.1.2 Motor Setup..... | 64 |
| 8.1.3 Using the Motor View..... | 64 |
| 8.1.4 Selecting a Motor..... | 65 |
| 8.1.4.1 Configuring Custom Motors..... | 66 |
| 8.1.4.2 Validating Motor Parameters..... | 66 |
| 8.2 Feedback 1..... | 66 |
| 8.2.1 Overview..... | 67 |
| 8.3 Using Feedback Options..... | 67 |
| 8.3.1 Auto..... | 67 |
| 8.3.2 Incremental Encoder..... | 67 |
| 8.3.3 Sine Encoder..... | 67 |
| 8.3.4 Endat 2.1, Endat 2.2..... | 67 |
| 8.3.5 BiSS..... | 67 |
| 8.3.6 Hiperface..... | 67 |
| 8.3.7 Resolver..... | 68 |
| 8.3.8 SFD..... | 68 |
| 8.3.9 Using Wake and Shake (WS)..... | 68 |
| 8.3.9.1 Overview..... | 68 |
| 8.3.9.2 Configuring WS..... | 68 |
| Wake and Shake, More View..... | 69 |
| Special Cases for WS..... | 70 |
| Operation with Motor Brake..... | 70 |
| End of Travel Limits..... | 70 |
| Large Load Inertia or High Friction System..... | 70 |
| 8.3.9.3 Using WS: Advanced..... | 70 |
| Maximizing WS Reliability..... | 71 |
| 8.3.9.4 Troubleshooting WS..... | 72 |

| | |
|--|----|
| 8.3.9.5 Related Parameters..... | 73 |
| 8.3.10 Commutation Alignment..... | 73 |
| 8.4 Feedback 2..... | 73 |
| 8.4.1 Encoder Emulation..... | 74 |
| 8.4.1.1 Overview..... | 74 |
| 8.4.1.2 Using Encoder Emulation..... | 74 |
| 8.4.1.3 Function Settings..... | 74 |
| Mode 0: Disable X9..... | 74 |
| Output Modes 1 and 2..... | 74 |
| Input Modes 3, 4, and 5..... | 75 |
| 8.4.1.4 Resolution..... | 76 |
| 8.4.1.5 Related Parameters and Commands..... | 76 |
| 8.5 Non-Plug and Play Feedback Devices..... | 76 |
| 8.6 Parameters..... | 76 |
| 8.7 Calculations..... | 76 |
| 8.7.1 Current Loop..... | 77 |
| 8.7.2 Velocity Loop..... | 77 |
| 8.7.3 Position Loop..... | 77 |
| 8.7.4 Slider Tuning..... | 77 |
| 8.7.4.1 Input - Motor Data..... | 77 |
| 8.7.4.2 Constants..... | 77 |
| 8.7.4.3 Output - Control Loop Gains..... | 77 |
| 8.8 Foldback..... | 77 |
| 8.8.1 Drive Foldback..... | 78 |
| 8.8.2 Setting up motor foldback..... | 78 |
| 8.8.3 Setting Fault and Warning Levels..... | 78 |
| 8.8.4 Motor Peak Current Time..... | 79 |
| 8.8.5 Motor Foldback Ramp..... | 79 |
| 8.8.6 Motor Recovery..... | 80 |
| 8.9 Overall Foldback..... | 80 |
| 8.10 Brake..... | 80 |
| 8.11 Using Position Capture..... | 81 |
| 8.11.1 Overview..... | 81 |
| 8.11.2 Configuring Position Capture..... | 81 |
| 8.11.2.1 Setting the Capture Source (CAP0.TRIGGER)..... | 81 |
| 8.11.2.2 Setting the Capture Mode (CAP0.MODE)..... | 81 |
| 8.11.2.3 Arming and Retrieving the Capture Value (CAP0.EN and CAP0.T)..... | 82 |
| 8.11.2.4 Setting the Capture Edge (CAP0.EDGE)..... | 82 |
| 8.11.2.5 Setting the Pre-Condition Event: (CAP0.EVENT)..... | 82 |
| 8.11.2.6 Setting up a Pre-Condition for complex capture..... | 82 |
| Setting the Precondition Edge: (Terminal Command: CAP0.PREEDGE)..... | 82 |
| Setting the Pre-Condition Select: (Terminal Command: CAP0.PRESELECT)..... | 82 |
| 8.11.3 Related Parameters..... | 82 |

| | | |
|-----------|---|-----------|
| 9 | Selecting Units for Your Application | 83 |
| 9.1 | Selecting and Saving Units | 84 |
| 9.2 | Units Example | 84 |
| 9.2.1 | Related Parameters | 86 |
| 10 | Configuring General Drive Settings | 87 |
| 10.1 | Digital Inputs and Outputs | 88 |
| 10.1.1 | Overview | 88 |
| 10.1.2 | Using Digital I/O | 88 |
| 10.1.3 | Digital Inputs | 88 |
| 10.1.4 | Digital Outputs | 94 |
| 10.2 | Command Buffer | 96 |
| 10.2.1 | Overview | 96 |
| 10.2.2 | Editing the Command Buffers | 97 |
| 10.2.3 | Behavior of the Command Buffer | 98 |
| 10.2.4 | Delays for the Buffer | 99 |
| 10.3 | Digital Inputs (X7/X8) | 99 |
| 10.3.1 | Digital Inputs 1 and 2 | 102 |
| 10.3.2 | Digital Inputs 3 to 7 | 102 |
| 10.3.3 | Digital Input 8 (ENABLE) | 102 |
| 10.4 | Analog Input | 103 |
| 10.4.1 | Analog Input | 103 |
| 10.4.2 | Related Parameters and Commands | 103 |
| 10.5 | Analog Output | 103 |
| 10.6 | I/O Connection | 104 |
| 10.6.1 | I/O Connectors (X7 and X8) | 105 |
| 10.7 | Electronic Gearing | 106 |
| 10.7.1 | Overview | 106 |
| 10.7.2 | Limits | 106 |
| 10.7.3 | Determining Maximum Cable Length | 107 |
| 10.7.4 | Related Parameters: | 108 |
| 10.7.5 | Limits | 109 |
| 10.8 | Programmable Limit Switch | 109 |
| 10.8.1 | Overview | 109 |
| 10.8.2 | Using Programmable Limit Switches | 109 |
| 10.8.3 | Single Shot Mode | 111 |
| 10.8.4 | Related Parameters and Commands | 111 |
| 10.9 | Enable/Disable | 111 |
| 10.9.1 | Enable Modes | 111 |
| 10.9.1.1 | Hardware Enable Mode | 111 |
| 10.9.1.2 | Software Enable Default | 112 |
| 10.9.2 | Disable Modes | 112 |
| 10.9.3 | Drive Status | 112 |
| 10.9.4 | Controlled Stop | 113 |
| 10.9.5 | More/Less Button | 113 |
| 10.10 | Controlled Stop | 115 |

| | | |
|-----------|--|------------|
| 10.10.0.1 | Related Parameters and Commands..... | 116 |
| 10.10.1 | Related topics:..... | 117 |
| 10.11 | Dynamic Braking..... | 117 |
| 10.11.1 | Drive Regeneration..... | 117 |
| 10.11.1.1 | AKD-x00306 to AKD-x00606..... | 117 |
| 10.11.1.2 | AKD-x01206 to AKD-x02406 and AKD-xzzz07..... | 117 |
| 10.12 | Emergency Stop..... | 118 |
| 10.13 | Safe Torque Off (STO)..... | 118 |
| 10.14 | Under Voltage Fault Behavior..... | 118 |
| 11 | Using Command Source and Operating Modes..... | 119 |
| 11.1 | Overview..... | 120 |
| 11.2 | Using Command Source and Operation Modes..... | 120 |
| 11.2.1 | Command Source..... | 120 |
| 11.2.1.1 | Service..... | 120 |
| 11.2.1.2 | Fieldbus..... | 120 |
| 11.2.1.3 | Electronic Gearing..... | 120 |
| 11.2.1.4 | Analog..... | 120 |
| 11.2.1.5 | Operation Mode..... | 120 |
| 11.2.2 | Related Parameters..... | 120 |
| 11.3 | Current Loop..... | 121 |
| 11.3.1 | Overview..... | 121 |
| 11.3.2 | Current Loop Gain..... | 121 |
| 11.3.3 | Current Loop Gain Scheduling..... | 122 |
| 11.3.3.1 | Using the Gain Scheduling View in WorkBench..... | 122 |
| 11.3.3.2 | Using the Terminal View for Gains Scheduling..... | 123 |
| 11.4 | Velocity Loop..... | 124 |
| 11.4.1 | Overview..... | 124 |
| 11.4.2 | Tabs in the Velocity Loop View..... | 124 |
| 11.4.3 | Velocity Loop Default Settings and Changes..... | 125 |
| 11.4.3.1 | Velocity Loop Changes Based on Slider Tuning..... | 125 |
| 11.4.3.2 | Velocity Loop Changes Based on PST..... | 126 |
| 11.5 | Position Loop..... | 126 |
| 11.5.1 | Overview..... | 126 |
| 11.5.2 | Tabs in the Position Loop View..... | 126 |
| 11.5.3 | Position Loop Default Behavior and Changes..... | 127 |
| 11.5.3.1 | Position Loop Changes Based on Slider Tuning..... | 127 |
| 11.5.3.2 | Position Loop Changes Based on PST..... | 127 |
| 11.5.4 | Modulo Position..... | 127 |
| 11.5.4.1 | Setting up the modulo axis in WorkBench..... | 128 |
| 11.5.4.2 | Setting up the modulo axis from the Terminal..... | 128 |
| 11.5.4.3 | Parameters affected by the modulo axis..... | 128 |
| 11.5.4.4 | Drive functions affected by modulo axis..... | 129 |
| | Software limit switch..... | 129 |
| | Programmable limit switch..... | 129 |
| | Digital output Modes 5 and 6..... | 129 |

| | |
|--|------------|
| Motion tasking to absolute target positions..... | 129 |
| 11.5.4.5 Using the modulo position feature with multiturn encoders..... | 129 |
| 12 Creating Motion..... | 133 |
| 12.1 Homing..... | 134 |
| 12.1.1 Overview..... | 134 |
| 12.1.2 Using Homing..... | 134 |
| 12.1.2.1 Home Default Window..... | 134 |
| Mode Selection:..... | 135 |
| Settings:..... | 135 |
| Controls:..... | 136 |
| 12.1.2.2 Selecting and Using Homing Modes..... | 136 |
| Homing Mode 0: Home Using Current Position..... | 136 |
| Homing Mode 1: Find Limit Input..... | 136 |
| Homing Mode 2: Find Input Limit then Find Zero Angle..... | 137 |
| Homing Mode 3: Find Input Limit then Find Index..... | 138 |
| Homing Mode 4: Find Home Input..... | 139 |
| Homing Mode 5: Find Home Input then Find Zero Angle..... | 139 |
| Homing Mode 6: Find Home Input then Find Index..... | 140 |
| Homing Mode 7: Find Zero Angle..... | 141 |
| Homing mode 8: Move Until Position Error Exceeded..... | 141 |
| Homing Mode 9: Move Until Position Error Exceeded then Find Zero Angle..... | 141 |
| 12.1.2.3 Homing Mode 10: Move Until Position Error Exceeded then Find Index..... | 142 |
| Homing Mode 11: Find Index Signal..... | 143 |
| 12.1.3 Using Homing: Advanced..... | 144 |
| 12.1.3.1 Related Parameters and Commands..... | 144 |
| 12.1.3.2 Related Topics..... | 144 |
| 12.2 Motion Tasks..... | 144 |
| 12.2.1 Overview..... | 144 |
| 12.2.2 Motion Task Input Table..... | 144 |
| 12.2.3 Using Motion Tasks..... | 145 |
| 12.2.3.1 Motion Profiles..... | 147 |
| 12.2.3.2 Motion Types..... | 147 |
| Absolute motion task..... | 148 |
| Motion task relative to command position (PL.CMD)..... | 148 |
| Motion task relative to previous target position..... | 148 |
| 12.2.4 Using Motion Tasks: Advanced..... | 148 |
| 12.2.4.1 Joining multiple tasks..... | 148 |
| 12.2.4.2 Start Conditions..... | 149 |

| | |
|---|------------|
| 12.2.4.3 Blending..... | 149 |
| 12.3 Registration Moves (Motion Task Blending)..... | 149 |
| 12.3.1 Configuring Registration Moves in WorkBench..... | 150 |
| 12.3.2 Configuring Registration Moves from the Terminal View..... | 150 |
| 12.4 Service Motion..... | 152 |
| 12.5 Jog Move..... | 153 |
| 12.6 Drive Motion Status..... | 153 |
| 13 Saving Your Drive Configuration..... | 155 |
| 13.1 Save Options..... | 156 |
| 13.2 Save On Exit..... | 156 |
| 13.3 Save On Disconnect..... | 157 |
| 13.4 Save On Firmware Download..... | 158 |
| 14 Tuning Your System..... | 159 |
| 14.1 Slider Tuning..... | 160 |
| 14.1.0.1 Gentle, Medium, and Stiff..... | 160 |
| 14.1.0.2 The Slider..... | 160 |
| 14.1.0.3 Inertia Ratio..... | 160 |
| 14.2 Load Inertia..... | 160 |
| 14.3 Using the Performance Servo Tuner..... | 160 |
| 14.3.1 Overview..... | 160 |
| 14.3.2 Using the PST..... | 161 |
| 14.3.2.1 Saving and Emailing Bode Plots..... | 162 |
| 14.3.2.2 Importing a Frequency Response..... | 163 |
| 14.3.3 Measurement Options..... | 163 |
| 14.3.3.1 Using Manual Excitation Levels..... | 163 |
| 14.3.3.2 Taking a Bode Measurement without the PST..... | 164 |
| 14.3.4 Using the Performance Servo Tuner: Advanced..... | 164 |
| 14.3.4.1 Typical Cases for Advanced PST Use..... | 165 |
| Tuning Systems with Low-Frequency Resonances..... | 165 |
| Tuning Systems with High-Frequency Resonances..... | 166 |
| Tuning systems with noisy frequency responses..... | 167 |
| 14.3.4.2 PST Options..... | 169 |
| Enable BiQuad 1 Tuner..... | 170 |
| Enable BiQuad 2 Tuner..... | 170 |
| Biquad Type..... | 170 |
| Tune Acceleration FF..... | 170 |
| Enable Stability Motion..... | 170 |
| Phase and Gain Margins..... | 170 |
| Tune Velocity Integral..... | 170 |
| Enable Low Pass Search..... | 170 |
| 14.3.4.3 Measurement Options..... | 170 |

| | |
|--|------------|
| Current Amplitude..... | 171 |
| Velocity Amplitude..... | 171 |
| Smooth %..... | 171 |
| Measurement..... | 174 |
| Injection Point..... | 174 |
| Excitation Type..... | 175 |
| FFT Points..... | 175 |
| Excite Gap..... | 175 |
| Number Points..... | 175 |
| Velocity Max..... | 176 |
| 14.3.4.4 Plot Options..... | 177 |
| Coherence..... | 177 |
| 14.3.4.5 Cursors..... | 177 |
| 14.3.4.6 Resizing Bode Plots..... | 179 |
| Reading and Understanding the Bode Plot..... | 180 |
| Using the Performance Servo Tuner to Manually Tune Systems..... | 183 |
| Using the Tuning Simulation..... | 184 |
| Using the Performance Servo Tuner Manual Tuning Interface..... | 184 |
| Simulating Modified Loop Gains with the Performance Servo Tuner..... | 185 |
| Simulating Filters with the Performance Servo Tuner..... | 188 |
| Using Filters to Reduce Noise..... | 189 |
| 14.4 Tuning Guide..... | 189 |
| 14.4.1 Overview..... | 189 |
| 14.4.2 Determining Tuning Criteria..... | 190 |
| 14.4.3 Before You Tune..... | 190 |
| 14.4.4 Closed Loop Tuning Methods..... | 190 |
| 14.4.4.1 Tuning the Velocity Loop..... | 191 |
| 14.4.4.2 Tuning the Position Loop..... | 193 |
| 14.4.5 Torque Feedforward Tuning Methods..... | 193 |
| 14.4.5.1 Shape Based Feedforward Tuning..... | 193 |
| 14.4.6 Using Anti-Resonance Filters..... | 194 |
| 14.4.6.1 Biquad Calculations..... | 202 |
| 14.4.6.2 Common Uses Of Anti Resonance Filters..... | 204 |
| 15 Scope..... | 205 |
| 15.1 Overview..... | 205 |
| 15.2 Using the Scope..... | 205 |
| 15.2.1 Scope Channels Tab..... | 205 |
| 15.2.1.1 Source Column..... | 205 |

| | | |
|-----------|--|------------|
| 15.2.1.2 | Color Column..... | 206 |
| 15.2.1.3 | Hide Column..... | 206 |
| 15.2.1.4 | Y-Axis Column..... | 206 |
| 15.2.1.5 | Filter and Filter Frequency Column..... | 206 |
| 15.2.2 | Scope Time-base and Trigger Tab..... | 206 |
| 15.2.2.1 | Scope Time-base and Trigger, More View..... | 207 |
| 15.2.2.2 | Trigger Type..... | 208 |
| 15.2.2.3 | Trigger Position..... | 208 |
| 15.2.2.4 | Trigger Value..... | 209 |
| 15.2.2.5 | Effects of Recorder Gap..... | 210 |
| 15.2.2.6 | Trigger Slope..... | 211 |
| 15.3 | Scope Settings..... | 211 |
| 15.3.1 | Load a setting (preset) to Scope screen..... | 212 |
| 15.3.2 | Create a new preset..... | 212 |
| 15.3.3 | Save or delete preset..... | 212 |
| 15.3.4 | Import preset..... | 212 |
| 15.3.5 | Export preset..... | 213 |
| 15.3.6 | Scope axis scaling and zooming..... | 214 |
| 15.3.7 | Manual range per axis..... | 215 |
| 15.3.8 | Unit display on Y axis..... | 215 |
| 16 | Using Parameters and the Terminal Screen..... | 217 |
| 16.1 | Terminal..... | 218 |
| 16.1.1 | Overview..... | 218 |
| 16.1.2 | Using the Terminal..... | 218 |
| 16.1.3 | Macros..... | 219 |
| 16.1.3.1 | Creating a Macro from Terminal commands..... | 219 |
| 16.1.3.2 | Macro Editor..... | 220 |
| 16.2 | Viewing Parameters..... | 221 |
| 16.3 | Parameter List..... | 221 |
| 16.4 | Parameter Load/Save..... | 221 |
| 16.5 | Parameter Comparer..... | 221 |
| 16.5.1 | Reference Parameter Selection..... | 222 |
| 16.5.2 | Target Parameter Selection..... | 222 |
| 16.5.3 | Display the comparison..... | 223 |
| 16.5.4 | Motion Task Comparison..... | 224 |
| | Summary of Parameters and Commands..... | 225 |
| 17 | Faults and Warnings..... | 239 |
| 17.1 | Fault and Warning Messages..... | 240 |
| 17.2 | Clearing Faults..... | 249 |
| 17.3 | Parameter and Command Error Messages..... | 250 |
| 17.4 | CANopen Emergency Messages and Error Codes..... | 253 |
| 17.5 | Unknown Fault..... | 256 |
| 17.5.0.1 | Remedies..... | 256 |
| 18 | Troubleshooting the AKD..... | 257 |
| 19 | Firmware and Firmware Updates..... | 258 |

| | | |
|---------------------------------|--|------------|
| 19.1 | Downloading Firmware..... | 259 |
| 19.2 | Firmware Compatibility..... | 259 |
| 19.3 | Invalid Firmware..... | 259 |
| 19.4 | Forcing the drive into firmware download mode..... | 260 |
| 19.5 | Forced download of AKD firmware..... | 260 |
| Connection Diagrams..... | | 263 |
| 19.6 | Connection Diagram, AKD-x00306 to x00606..... | 264 |
| 19.7 | Connection Diagram, AKD-x01206..... | 265 |
| 19.8 | Connection Diagram, AKD-x02406 and AKD-xzzz07..... | 266 |
| 19.9 | 24 V Auxiliary Supply (X1)..... | 267 |
| 19.10 | Motor Connection..... | 268 |
| 19.11 | External Regen Resistor (X3)..... | 269 |
| 19.12 | DC Bus Link (X3)..... | 270 |
| 19.13 | Mains Supply Connection (X3, X4)..... | 271 |
| 19.13.1 | Three Phase connection (all AKD types)..... | 271 |
| 19.13.2 | Single phase connection (AKD-xzzz06 only)..... | 272 |
| 19.14 | I/O Connection..... | 273 |
| 19.14.1 | I/O Connectors (X7 and X8)..... | 273 |
| 19.15 | Analog Output (X8)..... | 274 |
| 19.16 | Analog Input (X8)..... | 275 |
| 19.17 | Command encoder signal connection..... | 276 |
| 19.17.1 | Incremental encoder input 5 V (X9)..... | 276 |
| 19.17.2 | Incremental encoder input 24 V (X7)..... | 276 |
| 19.17.3 | Emulated Encoder Output (EEO) - A quad B (X9)..... | 277 |
| 19.18 | Pulse / Direction signal connection..... | 278 |
| 19.18.1 | Pulse / Direction input 5 V (X9)..... | 278 |
| 19.18.2 | Pulse / Direction input 24 V (X7)..... | 278 |
| 19.19 | Up / Down signal connection..... | 278 |
| 19.19.1 | Up / Down input 5 V (X9)..... | 278 |
| 19.19.2 | Up / Down input 24 V (X7)..... | 279 |
| 19.20 | Feedback Connector (X10)..... | 279 |
| Block Diagrams..... | | 281 |
| 19.21 | Block Diagram for Current Loop..... | 282 |
| 19.22 | Block Diagram for Position/Velocity Loop..... | 282 |
| Appendix A..... | | 1 |
| 20 | Modbus..... | 3 |
| 20.1 | Overview..... | 3 |
| 20.2 | Installation and Setup..... | 3 |
| 20.3 | Overview of Messaging..... | 3 |
| 20.4 | Supported Functions..... | 3 |
| 20.5 | Read Holding Registers (0x03)..... | 3 |
| 20.6 | Write Multiple Registers (0x10)..... | 4 |
| 20.7 | Exception Response Codes..... | 5 |
| 20.8 | Modbus..... | 6 |
| 20.8.1 | Overview..... | 6 |

| | | |
|-----------|--|-----------|
| 20.8.2 | Modbus Installation and Setup..... | 6 |
| 20.8.3 | Overview of Messaging..... | 6 |
| 20.8.4 | Supported Functions..... | 6 |
| 20.8.5 | Read Holding Registers (0x03)..... | 6 |
| 20.8.6 | Write Multiple Registers (0x10)..... | 8 |
| 20.8.7 | Exception Response Codes..... | 8 |
| 20.9 | Modbus Dynamic Mapping..... | 9 |
| 20.9.1 | Configuring Dynamic Mapping..... | 9 |
| 20.9.2 | Saving and Resetting Dynamic Mapping..... | 10 |
| 20.9.3 | Scaling parameters..... | 10 |
| 20.9.4 | Modbus specific registers (Parameters)..... | 11 |
| 20.9.5 | 32-bit versus 16-bit Values..... | 11 |
| 20.9.6 | Mapping of 64-bit Parameters to 32-bit Parameters..... | 11 |
| 20.9.7 | Fault Registers..... | 12 |
| 20.9.8 | Mapping Table..... | 12 |
| 20.10 | Modbus Parameter Table..... | 13 |
| 20.11 | Modbus 64-bit Parameters to 32-bit Mapping..... | 23 |
| 21 | Ethercat Manual..... | 27 |
| 21.1 | General..... | 58 |
| 21.1.1 | About this Manual..... | 59 |
| 21.1.2 | Target Group..... | 59 |
| 21.1.3 | Abbreviations Used..... | 60 |
| 21.2 | Safety..... | 61 |
| 21.2.1 | Safety Instructions..... | 62 |
| 21.2.2 | Use as directed..... | 62 |
| 21.2.3 | Prohibited use..... | 62 |
| 21.3 | Installation and Setup..... | 63 |
| 21.3.1 | Safety Instructions..... | 64 |
| 21.3.2 | EtherCAT Onboard..... | 65 |
| 21.3.2.1 | LED functions..... | 65 |
| 21.3.2.2 | Connection technology..... | 65 |
| 21.3.2.3 | Network Connection Example..... | 65 |
| 21.3.3 | Guide to Setup..... | 66 |
| 21.3.4 | Setup via TwinCAT NC/PTP System Manager..... | 66 |
| 21.3.4.1 | Scan devices..... | 67 |
| 21.3.4.2 | Select the device..... | 67 |
| 21.3.4.3 | Scan for boxes..... | 68 |
| 21.3.4.4 | Add Slaves to NC tasks..... | 68 |
| 21.3.4.5 | Enable the network configuration..... | 69 |
| 21.3.4.6 | Enable the axis and move the axis..... | 70 |
| 21.3.5 | Setup WorkBench over TwinCAT..... | 71 |
| 21.3.5.1 | TwinCAT and WorkBenchconfiguration..... | 72 |
| 21.3.5.2 | Connecting to a drive using WorkBench..... | 73 |
| 21.3.5.3 | Configuring and enabling a drive..... | 76 |
| 21.3.6 | Setup via KAS IDE..... | 77 |

| | |
|---|------------|
| 21.4 EtherCAT Profile..... | 78 |
| 21.4.1 Slave Register..... | 79 |
| 21.4.2 AL Event (Interrupt Event) and Interrupt Enable..... | 80 |
| 21.4.2.1 Interrupt Enable Register (Address 0x0204:0x0205)..... | 80 |
| 21.4.2.2 AL Event Request (Address 0x0220:0x0221)..... | 81 |
| 21.4.3 Phase Run-Up..... | 82 |
| 21.4.3.1 AL Control (Address 0x0120:0x0121)..... | 82 |
| 21.4.3.2 AL Status (Address 0x0130:0x0131)..... | 82 |
| 21.4.3.3 AL Status Code (Address 0x0134:0x0135)..... | 83 |
| 21.4.3.4 EtherCAT communication phases..... | 83 |
| 21.4.4 CANopen over EtherCAT (CoE) Status Machine..... | 84 |
| 21.4.4.1 Status Description..... | 84 |
| 21.4.4.2 Commands in the Control Word..... | 85 |
| 21.4.4.3 Status Machine Bits (status word)..... | 86 |
| 21.4.5 Fixed PDO Mappings..... | 87 |
| 21.4.6 Supported Cyclical Setpoint and Actual Values..... | 88 |
| 21.4.7 Supported Operation Modes..... | 88 |
| 21.4.8 Adjusting EtherCAT Cycle Time..... | 89 |
| 21.4.9 Maximum Cycle Times depending on operation mode..... | 89 |
| 21.4.10 Synchronization..... | 90 |
| 21.4.10.1 Synchronization behavior with distributed clocks (DC) enabled..... | 90 |
| 21.4.10.2 Synchronization behavior with distributed clocks (DC) disabled..... | 90 |
| 21.4.11 Latch Control Word and Latch Status Word..... | 91 |
| 21.4.12 Mailbox Handling..... | 92 |
| 21.4.12.1 Mailbox Output..... | 93 |
| 21.4.12.2 Mailbox Input..... | 94 |
| 21.4.12.3 Example: Mailbox Access..... | 95 |
| 21.4.13 Fieldbus Parameters..... | 96 |
| 21.5 Index..... | 98 |
| 22 CANopen Manual..... | 104 |
| 22.1 General..... | 135 |
| 22.1.1 About this Manual..... | 136 |
| 22.1.2 Target Group..... | 136 |
| 22.1.3 Abbreviations used..... | 137 |
| 22.2 Safety..... | 138 |
| 22.2.1 Safety Instructions..... | 139 |
| 22.2.2 Use As Directed..... | 139 |
| 22.2.3 Prohibited Use..... | 139 |
| 22.3 Installation and Setup..... | 140 |
| 22.3.1 Safety Instructions..... | 141 |
| 22.3.2 Guide to Setup..... | 142 |
| 22.4 CANopen Basics..... | 142 |
| 22.4.1 Basic Features implemented by CANopen..... | 143 |
| 22.4.1.1 Setup and general functions:..... | 143 |
| 22.4.1.2 Positioning functions:..... | 143 |

| | | |
|---|--|-----|
| 22.4.1.3 | Data transfer functions:..... | 143 |
| 22.4.2 | Transmission Rate and Procedure..... | 143 |
| 22.4.3 | Response to BUSOFF Communication Faults..... | 144 |
| 22.4.4 | Important Configuration Parameters..... | 144 |
| 22.5 | CANopen Communication Profile..... | 144 |
| 22.5.1 | General Description of CAN..... | 145 |
| 22.5.2 | Construction of the Communication Object Identifier..... | 146 |
| 22.5.3 | Definition of the Used Data Types..... | 147 |
| 22.5.3.1 | Basic data types..... | 147 |
| Unsigned Integer..... | | 147 |
| Signed Integer..... | | 148 |
| 22.5.3.2 | Mixed data types..... | 148 |
| 22.5.3.3 | Extended data types..... | 149 |
| Octet String..... | | 149 |
| Visible String..... | | 149 |
| 22.5.4 | Communication Objects..... | 149 |
| 22.5.4.1 | Network Management Objects (NMT)..... | 150 |
| 22.5.4.2 | Synchronization Object (SYNC)..... | 150 |
| 22.5.4.3 | Time-Stamp Object (TIME)..... | 150 |
| 22.5.4.4 | Emergency Object (EMCY)..... | 151 |
| Application of the Emergency Object..... | | 151 |
| Composition of the Emergency Object..... | | 151 |
| 22.5.4.5 | Service Data Objects (SDO)..... | 152 |
| Composition of the Service Data Object..... | | 152 |
| Initiate SDO Download Protocol..... | | 154 |
| Download SDO Segment Protocol..... | | 154 |
| Initiate SDO Upload Protocol..... | | 154 |
| Upload SDO Segment Protocol..... | | 154 |
| Abort SDO Protocol..... | | 154 |
| 22.5.4.6 | Process Data Object (PDO)..... | 155 |
| Transmission modes..... | | 156 |
| Trigger modes..... | | 156 |
| 22.5.4.7 | Nodeguard..... | 157 |
| 22.5.4.8 | Heartbeat..... | 158 |
| 22.6 | CANopen Drive Profile..... | 159 |
| 22.6.1 | CANopen Emergency Messages and Error Codes..... | 160 |
| 22.6.2 | General Definitions..... | 163 |
| 22.6.2.1 | General objects..... | 163 |
| Object 1000h: Device Type (DS301)..... | | 163 |

| | |
|---|-----|
| Object 1001h: Error register (DS301) | 164 |
| Object 1002h: Manufacturer Status Register (DS301) | 165 |
| Object 1003h: Predefined Error Field (DS301) | 166 |
| Object 1005h: COB-ID of the SYNC Message (DS301) | 167 |
| Object 1006h: Communication Cycle Period (DS301) | 167 |
| Object 1008h: Manufacturer Device Name (DS301) | 168 |
| Object 1009h: Manufacturer Hardware Version | 168 |
| Object 100Ah: Manufacturer Software Version (DS301) | 168 |
| Object 100Ch: Guard Time (DS301)Response monitoring | 169 |
| Object 100Dh: Lifetime Factor (DS301) | 169 |
| Object 1010h: Store Parameters (DS301) | 170 |
| Object 1014h: COB-ID for Emergency Message (DS301) | 171 |
| Object 1016h: Consumer Heartbeat Time | 171 |
| Object 1017h: Producer Heartbeat Time | 172 |
| Object 1018h: Identity Object (DS301) | 172 |
| Object 1026h: OS Prompt | 174 |
| 22.6.2.2 Manufacturer specific objects | 175 |
| Object 2014-2017h: 1st-4th Mask 1 to 4 for Transmit-PDO | 175 |
| Object 2018h: Firmware Version | 176 |
| Object 2026h: ASCII Channel | 177 |
| Object 20A0h: Latch position 1, positive edge | 178 |
| Object 20A1h: Latch position 1, negative edge | 178 |
| Object 20A2h: Latch position 2, positive edge | 178 |
| Object 20A3h: Latch position 2, negative edge | 179 |
| Object 20A4h: Latch Control Register | 179 |
| Object 20A5h: Latch Status Register | 180 |
| Object 20A6h: Latch position 1, positive or negative edge | 180 |
| Object 20B8h: Reset of changed input information | 181 |
| 22.6.2.3 Profile specific objects | 182 |
| Object 60FDh: Digital inputs (DS402) | 182 |
| Object 60FEh: Digital outputs (DS402) | 182 |
| Object 6502h: Supported drive modes (DS402) | 183 |
| 22.6.3 PDO Configuration | 184 |
| 22.6.3.1 Receive PDOs (RXPDO) | 185 |

| | |
|--|-----|
| Objects 1400-1403h: 1st - 4th RXPDO communication parameter (DS301)..... | 185 |
| Objects 1600-1603h: 1st - 4th RXPDO mapping parameter (DS301)..... | 186 |
| Default RXPDO definition..... | 187 |
| 22.6.3.2 Transmit PDOs (TXPDO)..... | 188 |
| Objects 1800-1803h: 1st - 4th TXPDO communication parameter (DS301)..... | 188 |
| Objects 1A00-1A03h: 1st - 4th TXPDO mapping parameter (DS301)..... | 190 |
| Default TXPDO definition..... | 191 |
| 22.6.4 Device Control (dc)..... | 192 |
| 22.6.4.1 Status Machine (DS402)..... | 192 |
| States of the Status Machine..... | 193 |
| Transitions of the status machine..... | 194 |
| 22.6.4.2 Object Description..... | 195 |
| Object 6040h: Control word (DS402)..... | 195 |
| Object 6041h: Status word (DS402)..... | 196 |
| Object 6060h: Modes of Operation (DS402)..... | 198 |
| Object 6061h: Modes of Operation Display (DS402)..... | 199 |
| 22.6.5 Factor Groups (fg) (DS402)..... | 199 |
| 22.6.5.1 General Information..... | 199 |
| Factors..... | 199 |
| Relationship between Physical and Internal Units..... | 199 |
| 22.6.5.2 Objects for velocity scaling..... | 200 |
| Object 204Ch: pv scaling factor..... | 200 |
| 22.6.5.3 Objects for position calculation..... | 201 |
| Object 608Fh: Position encoder resolution (DS402)..... | 201 |
| Object 6092h: Feed constant (DS402)..... | 202 |
| 22.6.6 Profile Velocity Mode (pv) (DS402)..... | 203 |
| 22.6.6.1 General Information..... | 203 |
| Objects that are defined in this section..... | 203 |
| Objects that are defined in other sections..... | 203 |
| 22.6.6.2 Object description..... | 203 |
| Object 606Ch: velocity actual value (DS402)..... | 203 |
| Object 60FFh: target velocity (DS402)..... | 204 |
| 22.6.7 Profile Torque Mode (tq) (DS402)..... | 205 |
| 22.6.7.1 General Information..... | 205 |
| Objects that are defined in this section..... | 205 |

| | |
|---|-----|
| Objects that are defined in other sections..... | 205 |
| 22.6.7.2 Object description..... | 205 |
| Object 6071h: Target torque (DS402)..... | 205 |
| Object 6073h: Max current (DS402)..... | 205 |
| Object 6077h: Torque actual value (DS402)..... | 206 |
| 22.6.8 Position Control Function (pc) (DS402)..... | 207 |
| 22.6.8.1 General Information..... | 207 |
| Objects that are defined in this section..... | 207 |
| Objects that are defined in other sections..... | 207 |
| 22.6.8.2 Object Description..... | 207 |
| Object 6063h: position actual value* (DS402)..... | 207 |
| Object 6064h: position actual value (DS402)..... | 208 |
| Object 6065h: Following error window..... | 208 |
| Object 60F4h: Following error actual value (DS402)..... | 208 |
| 22.6.9 Interpolated Position Mode (ip) (DS402)..... | 209 |
| 22.6.9.1 General information..... | 209 |
| Objects defined in this section..... | 209 |
| Objects defined in other sections..... | 209 |
| 22.6.9.2 Object description..... | 209 |
| Object 60C0h: Interpolation sub mode select..... | 209 |
| Object 60C1h: Interpolation data record..... | 210 |
| Object 60C2h: Interpolation time period..... | 211 |
| Object 60C4h: Interpolation data configuration..... | 212 |
| 22.6.10 Homing Mode (hm) (DS402)..... | 214 |
| 22.6.10.1 General information..... | 214 |
| Objects that are defined in this section..... | 214 |
| Objects that are defined in other sections..... | 214 |
| 22.6.10.2 Object Description..... | 214 |
| Object 607Ch: homing offset (DS402)..... | 214 |
| Object 6098h: homing method (DS402)..... | 214 |
| Object 6099h: homing speeds (DS402)..... | 216 |
| Object 609Ah: homing acceleration (DS402)..... | 216 |
| Homing Mode Sequence..... | 217 |
| 22.6.11 Profile Position Mode (pp)..... | 218 |
| 22.6.11.1 General Information..... | 218 |

| | |
|---|------------|
| Objects that are defined in this section..... | 218 |
| Objects that are defined in other sections..... | 218 |
| 22.6.11.2 Object Description..... | 219 |
| Object 607Ah: target position (DS402)..... | 219 |
| Object 607Dh: Software position limit (DS402)..... | 219 |
| Object 6081h: profile velocity (DS402)..... | 220 |
| Object 6083h: profile acceleration (DS402)..... | 220 |
| Object 6084h: profile deceleration (DS402)..... | 220 |
| Functional Description..... | 221 |
| 22.7 Appendix..... | 223 |
| 22.7.1 Object Dictionary..... | 223 |
| 22.7.1.1 Float Scaling..... | 223 |
| 22.7.1.2 Communication SDOs..... | 223 |
| 22.7.1.3 Manufacturer specific SDOs..... | 226 |
| 22.7.1.4 Profile specific SDOs..... | 234 |
| 22.7.2 Examples..... | 236 |
| 22.7.2.1 Examples, setup..... | 236 |
| Basic testing of the connection to the AKD controls..... | 236 |
| Example: Operating the Status Machine..... | 237 |
| Example: Jog Mode via SDO..... | 238 |
| Example: Torque Mode via SDO..... | 238 |
| Example: Jog Mode via PDO..... | 239 |
| Example: Torque Mode via PDO..... | 241 |
| Example: Homing via SDO..... | 242 |
| Example: Using the Profile Position Mode..... | 244 |
| Example: ASCII Communication..... | 247 |
| Test for SYNC telegrams..... | 248 |
| 22.7.2.2 Examples, special applications..... | 249 |
| Example: External Trajectory with Interpolated Position Mode..... | 249 |
| 22.8 Index..... | 255 |
| Appendix B..... | 261 |
| | 262 |
| About the Parameter and Command Reference Guide..... | 263 |
| Parameter and Command Naming Conventions..... | 265 |
| Summary of Parameters and Commands..... | 265 |
| AIN Parameters..... | 279 |
| AIN.CUTOFF..... | 280 |
| AIN.DEADBAND..... | 281 |

| | |
|-------------------------------------|------------|
| AIN.DEADBANDMODE..... | 283 |
| AIN.ISCALE..... | 285 |
| AIN.OFFSET..... | 286 |
| AIN.PSCALE..... | 287 |
| AIN.VALUE..... | 288 |
| AIN.VSCALE..... | 289 |
| AIN.ZERO..... | 290 |
| AIO Parameters..... | 291 |
| AIO.ISCALE..... | 291 |
| AIO.PSCALE..... | 292 |
| AIO.VSCALE..... | 293 |
| AOUT Parameters..... | 295 |
| AOUT.DEBUGADDR..... | 295 |
| AOUT.DEBUGSCALE..... | 296 |
| AOUT.ISCALE..... | 297 |
| AOUT.MODE..... | 298 |
| AOUT.OFFSET..... | 299 |
| AOUT.PSCALE..... | 300 |
| AOUT.VALUE..... | 302 |
| AOUT.VALUEU..... | 303 |
| AOUT.VSCALE..... | 304 |
| BODE Parameters..... | 305 |
| BODE.EXCITEGAP..... | 305 |
| BODE.FREQ..... | 306 |
| BODE.IAMP..... | 307 |
| BODE.IFLIMIT..... | 308 |
| BODE.IFTHRESH..... | 309 |
| BODE.INJECTPOINT..... | 310 |
| BODE.MODE..... | 311 |
| BODE.MODETIMER..... | 314 |
| BODE.PRDEPTH..... | 316 |
| BODE.VAMP..... | 317 |
| BODE.VFLIMIT..... | 318 |
| BODE.VFTHRESH..... | 319 |
| CAP Parameters..... | 320 |
| CAP0.EDGE, CAP1.EDGE..... | 320 |
| CAP0.EN, CAP1.EN..... | 321 |
| CAP0.EVENT, CAP1.EVENT..... | 322 |
| CAP0.FILTER, CAP1.FILTER..... | 325 |
| CAP0.MODE, CAP1.MODE..... | 326 |
| CAP0.PLFB, CAP1.PLFB..... | 327 |
| CAP0.PREEDGE, CAP1.PREEDGE..... | 328 |
| CAP0.PREFILTER, CAP1.PREFILTER..... | 329 |
| CAP0.PRESELECT, CAP1.PRESELECT..... | 330 |
| CAP0.STATE, CAP1.STATE..... | 331 |

| | |
|---|------------|
| CAP0.T, CAP1.T..... | 332 |
| CAP0.TRIGGER, CAP1.TRIGGER..... | 333 |
| CS Parameters..... | 334 |
| CS.DEC..... | 334 |
| CS.STATE..... | 336 |
| CS.TO..... | 337 |
| CS.VTHRESH..... | 337 |
| DIN Parameters..... | 339 |
| DIN.HCMD1 TO DIN.HCMD4..... | 340 |
| DIN.LCMD1 to DIN.LCMD4..... | 341 |
| DIN.ROTARY..... | 341 |
| DIN.STATES..... | 343 |
| DIN1.FILTER TO DIN7.FILTER..... | 344 |
| DIN1.INV TO DIN7.INV..... | 345 |
| DIN1.MODE TO DIN7.MODE..... | 346 |
| DIN1.PARAM TO DIN7.PARAM..... | 348 |
| DIN1.STATE TO DIN7.STATE..... | 349 |
| DOUT Parameters..... | 351 |
| DOUT.CTRL..... | 351 |
| DOUT.RELAYMODE..... | 352 |
| DOUT.STATES..... | 353 |
| DOUT1.MODE AND DOUT2.MODE..... | 354 |
| DOUT1.PARAM AND DOUT2.PARAM..... | 355 |
| DOUT1.STATE AND DOUT2.STATE..... | 356 |
| DOUT1.STATEU AND DOUT2.STATEU..... | 357 |
| DRV Parameters and Commands..... | 359 |
| DRV.ACC..... | 359 |
| DRV.ACTIVE..... | 361 |
| DRV.BLINKDISPLAY..... | 362 |
| DRV.CLRFAULTHIST..... | 363 |
| DRV.CLRFAULTS..... | 364 |
| DRV.CMDDELAY..... | 365 |
| DRV.CMDSOURCE..... | 366 |
| DRV.CRASHDUMP..... | 367 |
| DRV.DBILIMIT..... | 368 |
| DRV.DEC..... | 369 |
| DRV.DIR..... | 370 |
| DRV.DIS..... | 372 |
| DRV.DISMEDIA..... | 373 |
| DRV.DISSOURCES..... | 374 |
| DRV.DISTO..... | 375 |
| DRV.EMUEDIR..... | 376 |
| DRV.EMUEMODE..... | 377 |
| DRV.EMUEMTURN..... | 378 |
| DRV.EMUERES..... | 379 |

| | |
|----------------------------|------------|
| DRV.EMUEZOFFSET..... | 380 |
| DRV.EN..... | 381 |
| DRV.ENDEFAULT..... | 382 |
| DRV.FAULTHIST..... | 383 |
| DRV.FAULTS..... | 384 |
| DRV.HANDWHEEL..... | 385 |
| DRV.HELP..... | 386 |
| DRV.HELPALL..... | 387 |
| DRV.HWENMODE..... | 388 |
| DRV.ICONT..... | 389 |
| DRV.INFO..... | 390 |
| DRV.IPEAK..... | 391 |
| DRV.IZERO..... | 392 |
| DRV.LIST..... | 393 |
| DRV.LOGICVOLTS..... | 394 |
| DRV.MEMADDR..... | 395 |
| DRV.MEMDATA..... | 396 |
| DRV.MOTIONSTAT..... | 397 |
| DRV.NAME..... | 398 |
| DRV.NVLIST..... | 399 |
| DRV.NVLOAD..... | 400 |
| DRV.NVSAVE..... | 401 |
| DRV.ONTIME..... | 402 |
| DRV.OPMODE..... | 403 |
| DRV.READFORMAT..... | 404 |
| DRV.RSTVAR..... | 405 |
| DRV.RUNTIME..... | 406 |
| DRV.SETUPREQBITS..... | 407 |
| DRV.SETUPPREQLIST..... | 408 |
| DRV.STOP..... | 409 |
| DRV.TEMPERATURES..... | 410 |
| DRV.TYPE..... | 411 |
| DRV.VER..... | 412 |
| DRV.VERIMAGE..... | 413 |
| DRV.WARNINGS..... | 414 |
| DRV.ZERO..... | 415 |
| FB1 Parameters..... | 416 |
| FB1.BISSBITS..... | 416 |
| FB1.ENCRES..... | 417 |
| FB1.HALLSTATE..... | 418 |
| FB1.HALLSTATEU..... | 419 |
| FB1.HALLSTATEV..... | 420 |
| FB1.HALLSTATEW..... | 421 |
| FB1.IDENTIFIED..... | 422 |
| FB1.INITSIGNED..... | 423 |

| | |
|----------------------------------|------------|
| FB1.MECHPOS..... | 424 |
| FB1.MEMVER..... | 425 |
| FB1.OFFSET..... | 426 |
| FB1.ORIGIN..... | 427 |
| FB1.PFIND..... | 429 |
| FB1.PFINDCMDU..... | 430 |
| FB1.POLES..... | 431 |
| FB1.PSCALE..... | 432 |
| FB1.RESKTR..... | 433 |
| FB1.RESREFPHASE..... | 434 |
| FB1.SELECT..... | 435 |
| FB1.TRACKINGCAL..... | 437 |
| FB2 Parameters..... | 438 |
| FB2.ENCRES..... | 438 |
| FB2.MODE..... | 439 |
| FB2.SOURCE..... | 440 |
| FBUS Parameters..... | 441 |
| FBUS.PARAM1 TO FBUS.PARAM20..... | 441 |
| FBUS.PLLSTATE..... | 443 |
| FBUS.PLLTHRESH..... | 444 |
| FBUS.SAMPLEPERIOD..... | 445 |
| FBUS.SYNCACT..... | 446 |
| FBUS.SYNCDIST..... | 447 |
| FBUS.SYNCWND..... | 448 |
| FBUS.TYPE..... | 449 |
| GEAR Parameters..... | 450 |
| GEAR.ACCMAX..... | 450 |
| GEAR.DECMAX..... | 451 |
| GEAR.IN..... | 452 |
| GEAR.MODE..... | 453 |
| GEAR.MOVE..... | 455 |
| GEAR.OUT..... | 456 |
| GEAR.VMAX..... | 457 |
| GUI Parameters..... | 458 |
| GUI.DISPLAY..... | 458 |
| GUI.PARAM01..... | 459 |
| GUI.PARAM02..... | 459 |
| GUI.PARAM03..... | 459 |
| GUI.PARAM04..... | 460 |
| GUI.PARAM05..... | 460 |
| GUI.PARAM06..... | 460 |
| GUI.PARAM07..... | 461 |
| GUI.PARAM08..... | 461 |
| GUI.PARAM09..... | 461 |
| GUI.PARAM10..... | 462 |

| | |
|------------------------------|------------|
| HOME Parameters | 463 |
| HOME.ACC..... | 463 |
| HOME.AUTOMOVE..... | 465 |
| HOME.DEC..... | 466 |
| HOME.DIR..... | 467 |
| HOME.DIST..... | 468 |
| HOME.FEEDRATE..... | 469 |
| HOME.IPEAK..... | 470 |
| HOME.MODE..... | 471 |
| HOME.MOVE..... | 472 |
| HOME.P..... | 473 |
| HOME.PERRTHRESH..... | 474 |
| HOME.REQUIRE..... | 475 |
| HOME.SET..... | 476 |
| HOME.V..... | 477 |
| HWLS Parameters | 478 |
| HWLS.NEGSTATE..... | 478 |
| HWLS.POSSTATE..... | 479 |
| IL Parameters | 481 |
| IL.BUSFF..... | 481 |
| IL.CMD..... | 482 |
| IL.CMDU..... | 483 |
| IL.DIFOLD..... | 484 |
| IL.FB..... | 485 |
| IL.FF..... | 486 |
| IL.FOLDFTHRESH..... | 487 |
| IL.FOLDFTHRESHU..... | 488 |
| IL.FOLDWTHRESH..... | 489 |
| IL.FRCTION..... | 490 |
| IL.IFOLD..... | 491 |
| IL.IUFB..... | 492 |
| IL.IVFB..... | 493 |
| IL.KACCCFF..... | 494 |
| IL.KBUSFF..... | 495 |
| IL.KP..... | 496 |
| IL.KPDRATIO..... | 497 |
| IL.KPLOOKUPINDEX..... | 498 |
| IL.KPLOOKUPVALUE..... | 499 |
| IL.KPLOOKUPVALUES..... | 500 |
| IL.KVFF..... | 501 |
| IL.LIMITN..... | 502 |
| IL.LIMITP..... | 503 |
| IL.MFOLDD..... | 504 |
| IL.MFOLDR..... | 505 |
| IL.MFOLDT..... | 506 |

| | |
|--|------------|
| IL.MIFOLD..... | 507 |
| IL.OFFSET..... | 508 |
| IL.VCMD..... | 509 |
| IL.VUFB..... | 510 |
| IL.VVFB..... | 511 |
| 23 LOAD.INERTIA..... | 512 |
| 24 MODBUS Parameters..... | 513 |
| MODBUS.PIN AND MODBUS.POUT..... | 514 |
| MODBUS.PSCALE..... | 515 |
| MODBUS.UNITLABEL..... | 516 |
| MOTOR Parameters..... | 517 |
| MOTOR.AUTOSET..... | 517 |
| MOTOR.BRAKE..... | 518 |
| MOTOR.BRAKERLS..... | 519 |
| MOTOR.BRAKESTATE..... | 520 |
| MOTOR.CTF0..... | 521 |
| MOTOR.ICONT..... | 522 |
| MOTOR.IDDATAVALID..... | 523 |
| MOTOR.INERTIA..... | 524 |
| MOTOR.IPEAK..... | 525 |
| MOTOR.KE..... | 526 |
| MOTOR.KT..... | 527 |
| MOTOR.LQLL..... | 528 |
| MOTOR.NAME..... | 529 |
| MOTOR.PHASE..... | 530 |
| MOTOR.PITCH..... | 531 |
| MOTOR.POLES..... | 532 |
| MOTOR.R..... | 533 |
| MOTOR.RTYPE..... | 534 |
| MOTOR.TBRAKEAPP..... | 535 |
| MOTOR.TBRAKERLS..... | 536 |
| MOTOR.TEMP..... | 537 |
| MOTOR.TEMPFAULT..... | 538 |
| MOTOR.TEMPWARN..... | 539 |
| MOTOR.TYPE..... | 540 |
| MOTOR.VMAX..... | 541 |
| MOTOR.VOLTMAX..... | 542 |
| MOTOR.VOLTMIN..... | 543 |
| MOTOR.VOLTRATED..... | 544 |
| MOTOR.VRATED..... | 545 |
| MT Parameters and Commands..... | 547 |
| MT.ACC..... | 547 |
| MT.CLEAR..... | 549 |
| MT.CNTL..... | 550 |
| MT.CONTINUE..... | 553 |

| | |
|---|------------|
| MT.DEC..... | 554 |
| MT.EMERGMT..... | 556 |
| MT.HOMEREQUIRE..... | 557 |
| MT.LIST..... | 558 |
| MT.LOAD..... | 559 |
| MT.MOVE..... | 560 |
| MT.MTNEXT..... | 561 |
| MT.NUM..... | 562 |
| MT.P..... | 563 |
| MT.PARAMS..... | 564 |
| MT.SET..... | 565 |
| MT.TNEXT..... | 566 |
| MT.TNUM..... | 567 |
| MT.TNVSAVE..... | 568 |
| MT.TPOSWND..... | 569 |
| MT.TVELWND..... | 570 |
| MT.V..... | 571 |
| MT.VCMD..... | 572 |
| PL Parameters..... | 573 |
| PL.CMD..... | 573 |
| PL.ERR..... | 574 |
| PL.ERRFTHRESH..... | 575 |
| PL.ERRMODE..... | 577 |
| PL.ERRWTHRESH..... | 578 |
| PL.FB..... | 579 |
| PL.FBSOURCE..... | 580 |
| PL.INTINMAX..... | 581 |
| PL.INTOUTMAX..... | 582 |
| PL.KI..... | 583 |
| PL.KP..... | 584 |
| PL.MODP1..... | 585 |
| PL.MODP2..... | 586 |
| PL.MODPDIR..... | 587 |
| PL.MODPEN..... | 588 |
| PLS Parameters and Commands..... | 589 |
| PLS.EN..... | 589 |
| PLS.MODE..... | 590 |
| PLS.P1 TO PLS.P8..... | 591 |
| PLS.RESET..... | 592 |
| PLS.STATE..... | 593 |
| PLS.T1 TO PLS.T8..... | 594 |
| PLS.UNITS..... | 595 |
| PLS.WIDTH1 TO PLS.WIDTH8..... | 597 |
| REC Parameters and Commands..... | 599 |
| REC.ACTIVE..... | 599 |

| | |
|------------------------------|------------|
| REC.CH1 to REC.CH6..... | 600 |
| REC.DONE..... | 601 |
| REC.GAP..... | 602 |
| REC.NUMPOINTS..... | 603 |
| REC.OFF..... | 604 |
| REC.RECPRMLIST..... | 605 |
| REC.RETRIEVE..... | 606 |
| REC.RETRIEVEDATA..... | 607 |
| REC.RETRIEVEFRMT..... | 608 |
| REC.RETRIEVEHDR..... | 609 |
| REC.RETRIEVESIZE..... | 610 |
| REC.STOPTYPE..... | 611 |
| REC.TRIG..... | 612 |
| REC.TRIGPARAM..... | 613 |
| REC.TRIGPOS..... | 614 |
| REC.TRIGPRMLIST..... | 615 |
| REC.TRIGSLOPE..... | 616 |
| REC.TRIGTYPE..... | 617 |
| REC.TRIGVAL..... | 618 |
| REGEN Parameters..... | 619 |
| REGEN.POWER..... | 619 |
| REGEN.REXT..... | 620 |
| REGEN.TEXT..... | 621 |
| REGEN.TYPE..... | 622 |
| REGEN.WATTEXT..... | 623 |
| SM Parameters..... | 625 |
| SM.I1..... | 625 |
| SM.I2..... | 626 |
| SM.MODE..... | 627 |
| SM.MOVE..... | 630 |
| SM.T1..... | 631 |
| SM.T2..... | 632 |
| SM.V1..... | 633 |
| SM.V2..... | 634 |
| STO Parameters..... | 635 |
| STO.STATE..... | 635 |
| SWLS Parameters..... | 636 |
| SWLS.EN..... | 636 |
| SWLS.LIMIT0..... | 637 |
| SWLS.LIMIT1..... | 638 |
| SWLS.STATE..... | 639 |
| UNIT Parameters..... | 641 |
| UNIT.ACCLINEAR..... | 641 |
| UNIT.ACCROTARY..... | 642 |
| UNIT.LABEL..... | 643 |

| | |
|-------------------------------|------------|
| UNIT.PIN..... | 644 |
| UNIT.PLINEAR..... | 645 |
| UNIT.POUT..... | 646 |
| UNIT.PROTARY..... | 647 |
| UNIT.VLINEAR..... | 648 |
| UNIT.VROTARY..... | 649 |
| VBUS Parameters..... | 650 |
| VBUS.OVFTHRESH..... | 650 |
| VBUS.OVWTHRESH..... | 651 |
| VBUS.RMSLIMIT..... | 652 |
| VBUS.UVFTHRESH..... | 653 |
| VBUS.UVMODE..... | 654 |
| VBUS.UVWTHRESH..... | 655 |
| VBUS.VALUE..... | 656 |
| VL Parameters..... | 657 |
| VL.ARPF1 TO VL.ARPF4..... | 657 |
| VL.ARPQ1 TO VL.ARPQ4..... | 659 |
| VL.ARTYPE1 TO VL.ARTYPE4..... | 660 |
| VL.ARZF1 TO VL.ARZF4..... | 661 |
| VL.ARZQ1 TO VL.ARZQ4..... | 662 |
| VL.BUSFF..... | 663 |
| VL.CMD..... | 664 |
| VL.CMDU..... | 665 |
| VL.ERR..... | 666 |
| VL.FB..... | 667 |
| VL.FBFILTER..... | 668 |
| VL.FBSOURCE..... | 669 |
| VL.FBUNFILTERED..... | 670 |
| VL.FF..... | 671 |
| VL.GENMODE..... | 672 |
| VL.KBUSFF..... | 673 |
| VL.KI..... | 674 |
| VL.KO..... | 675 |
| VL.KP..... | 676 |
| VL.KVFF..... | 677 |
| VL.LIMITN..... | 678 |
| VL.LIMITP..... | 679 |
| VL.LMJR..... | 679 |
| VL.MODEL..... | 681 |
| VL.OBSBW..... | 682 |
| VL.OBSMODE..... | 683 |
| VL.THRESH..... | 684 |
| WS Parameters..... | 685 |
| WS.ARM..... | 685 |
| WS.DISARM..... | 686 |

| | |
|----------------------|------------|
| WS.DISTMAX..... | 687 |
| WS.DISTMIN..... | 688 |
| WS.IMAX..... | 689 |
| WS.MODE..... | 690 |
| WS.NUMLOOPS..... | 691 |
| WS.STATE..... | 692 |
| WS.T..... | 693 |
| WS.TDELAY1..... | 694 |
| WS.TDELAY2..... | 695 |
| WS.TDELAY3..... | 696 |
| WS.VTHRESH..... | 697 |
| 25 Index..... | 700 |

This page intentionally left blank.

22.1 General

22.1.1 About this Manual

This manual, *AKD CANopen Communication*, This manual describes the installation, setup, range of functions, and software protocol for the CANopen AKD product series. All AKD CANopen drives have built-in CANopen functionality; therefore an additional option card is not required.

A digital version of this manual (pdf format) is available on the CD-ROM included with your drive. Manual updates can be downloaded from the Kollmorgen website.

Related documents for the AKD series include:

- *AKD Quick Start* (also provided in hard copy). This guide provides instructions for basic drive setup and connection to a network.
- *AKD Installation Manual* (also provided in hard copy). This manual provides instructions for installation and drive setup.
- *AKD Parameter and Command Reference Guide*. This guide provides documentation for the parameters and commands used to program the AKD.
- *AKD EtherCAT Communication*. This manual includes setup information for the EtherCAT interface and describes the EtherCAT profile.

Additional documentation:

- CAN Application (CAL) for Industrial Applications (publisher CiA e.V.)
- Draft Standards 301 (from Version 4.0), 402 (publisher CiA e.V.)
- CAN Specification Version 2.0 (publisher CiA e.V.)
- ISO 11898 ... Controller Area Network (CAN) for high-speed communication

22.1.2 Target Group

This manual addresses personnel with the following qualifications:

- Installation: only by electrically qualified personnel.
- Setup : only by qualified personnel with extensive knowledge of electrical engineering and drive technology
- Programming: Software developers, project-planners

The qualified personnel must know and observe the following standards:

- ISO 12100, IEC 60364 and IEC 60664
- National accident prevention regulations

⚠ WARNING During operation there are deadly hazards, with the possibility of death, severe injury or material damage. The operator must ensure that the safety instructions in this manual are followed. The operator must ensure that all personnel responsible for working with the servo drive have read and understand the manual.

22.1.3 Abbreviations used

| Abbreviation | Meaning |
|--------------|--|
| BTB/RTO | Ready to operate (standby) |
| COB | Communication Object |
| COB-ID | Communication Object Identifier |
| EEPROM | Electrically erasable/programmable memory |
| EMC | Electromagnetic compatibility |
| EMCY | Emergency Objects |
| ISO | International Standardization Organization |
| km | 1000 m |
| LED | Light-emitting diode |
| LSB | Low significant Byte (or Bit) |
| MSB | Main significant Byte (or Bit) |
| MB | Megabyte |
| NMT | Network Management Objects |
| NSTOP | Limit switch for negative (left) rotation |
| PC | Personal Computer |
| PDO | Process Data Object |
| PSTOP | Limit switch for positive (right) rotation |
| RAM | Volatile memory |
| ROD | Incremental position encoder |
| RXPDO | Receive PDO |
| SDO | Service Data Object |
| SYNC | Synchronization Objects |
| TXPDO | Transmit PDO |

22.2 Safety

22.2.1 Safety Instructions

| | |
|------------------|--|
| ⚠ DANGER | <p>During operation there are deadly hazards, with the possibility of death, severe injury or material damage. Do not open or touch the equipment during operation. Keep all covers and cabinet doors closed during operation. Touching the equipment is allowed during installation and commissioning for properly qualified persons only.</p> <ul style="list-style-type: none"> • During operation, drives may have uncovered live components, depending on their level of enclosure protection. • Control and power connections may be live, even though the motor is not rotating. • Drives may have hot surfaces during operation. Heat sink can reach temperatures above 80°C. |
| ⚠ WARNING | <p>Electronic equipment is basically not failure-proof. The user is responsible for ensuring that, in the event of a failure of the drive, the drive is set to a state that is safe for both machinery and personnel, for instance with the aid of a mechanical brake.</p> <p>Drives with drives and CANopen expansion cards are remote-controlled machines. They can start to move at any time without previous warning. Take appropriate measures to ensure that the operating and service personnel is aware of this danger.</p> <p>Implement appropriate protective measures to ensure that any unintended start-up of the machines cannot result in dangerous situations for personnel or machinery. Software limit-switches are not a substitute for the hardware limit-switches in the machine.</p> |
| NOTICE | <p>Install the drive as described in the <i>Installation Manual</i>. The wiring for the analog setpoint input and the positioning interface, as shown in the wiring diagram in the <i>Installation Manual</i>, is not required. Never break any of the electrical connections to the drive while it is live. This action can result in destruction of the electronics</p> |

22.2.2 Use As Directed

Drives are components that are built into electrical plants or machines and can only be operated as integral components of these plants or machines. The manufacturer of the machine used with a drive must generate a hazard analysis for the machine and take appropriate measures to ensure that unforeseen movements cannot cause personnel injury or property damage.

Please observe the chapters "Use as directed" and "Prohibited use" in the *AKD Installation Manual*.

The CANopen interface serves only for the connection of the *AKD* to a master via the CAN bus.

22.2.3 Prohibited Use

Other use than that described in chapter "Use as directed" is not intended and can lead to personnel injuries and equipment damage. The drive may not be used with a machine that does not comply with appropriate national directives or standards. The use of the drive in the following environments is also prohibited:

- potentially explosive areas
- environments with corrosive and/or electrically conductive acids, alkaline solutions, oils, vapors, dusts
- ships or offshore applications

22.3 Installation and Setup

22.3.1 Safety Instructions

⚠ DANGER Never undo any electrical connections to the drive while it is live. There is a danger of electrical arcing with damage to contacts and serious personal injury. Wait at least seven minutes after disconnecting the drive from the main supply power before touching potentially live sections of the equipment (e.g. contacts) or undoing any connections. Capacitors can still have dangerous voltages present up to 7 minutes after switching off the supply power. To be sure, measure the voltage in the DC Bus link and wait until it has fallen below 40 V. Control and power connections can still be live, even if the motor is not rotating.

⚠ WARNING Electronic equipment is basically not failure-proof. The user is responsible for ensuring that, in the event of a failure of the drive, the drive is set to a state that is safe for both machinery and personnel, for instance with the aid of a mechanical brake. Drives with drives and CANopen are remote-controlled machines. They can start to move at any time without previous warning. Take appropriate measures to ensure that the operating and service personnel is aware of this danger. Implement appropriate protective measures to ensure that any unintended start-up of the machines cannot result in dangerous situations for personnel or machinery. Software limit-switches are not a substitute for the hardware limit-switches in the machine.

NOTICE Install the drive as described in the *Installation Manual*. The wiring for the analog setpoint input and the positioning interface, as shown in the wiring diagram in the *Installation Manual*, is not required. Never break any of the electrical connections to the drive while it is live. This action can result in destruction of the electronics.

NOTICE The drive's status must be monitored by the PLC to acknowledge critical situations. Wire the FAULT contact in series into the emergency stop circuit of the installation. The emergency stop circuit must operate the supply contactor.

NOTE It is permissible to use the setup software to alter the settings of the drive. Any other alterations will invalidate the warranty.

NOTE Because of the internal representation of the position-control parameters, the position controller can only be operated if the final limit speed of the drive does not exceed:

rotary

at sinusoidal² commutation: 7500 rpm

at trapezoidal commutation: 12000 rpm.

linear

at sinusoidal² commutation: 4 m/s

at trapezoidal commutation: 6.25 m/s

NOTE All the data on resolution, step size, positioning accuracy etc. refer to calculatory values. Non-linearities in the mechanism (backlash, flexing, etc.) are not taken into account. If the final limit speed of the motor must be altered, then all the parameters that were previously entered for position control and motion blocks must be adapted.

22.3.2 Guide to Setup

⚠ WARNING

Only professional personnel with extensive knowledge of control and drive technology are allowed to setup the drive.

⚠ CAUTION

Make sure that any unintended movement of the drive cannot endanger machinery or personnel.

1. Check assembly/installation. Check that all the safety instructions in the product manual for the drive and this manual have been observed and implemented. Check the setting for the station address and baud rate.
2. Connect PC, start WorkBench. Use the setup software WorkBench to set the parameters for the drive.
3. Setup basic functions. Start up the basic functions of the drive and optimize the current, speed and position controllers. This section of the setup is described in the in the online help of the setup software.
4. Save parameters. When the parameters have been optimized, save them in the drive.
5. Start up communication. The altered parameters will only become effective after a reboot (switch off 24V and switch on again). Adjust the transmission rate of the AKD to match the master.
6. Test communication. Check for the bootup-message, when you switch on the drive. Do an SDO read access on index 0x1000 subindex 0 (DeviceType).
7. Setup position controller. Setup the position controller, as described in the WorkBenchonline help.

22.4 CANopen Basics

22.4.1 Basic Features implemented by CANopen

It is assumed that the basic operating functions of the communication profile are known and available as reference documentation. When working with the position controller that is integrated in AKD, the following functions are available:

22.4.1.1 Setup and general functions:

- Homing, set reference point
- Provision of a digital setpoint for speed and torque control
- Support of the following modes of the CANopen Profile DS402:
 - Profile position mode
 - Homing mode
 - Profile torque mode
 - Interpolated position mode
 - Profile velocity mode

22.4.1.2 Positioning functions:

- Execution of a motion task from the motion block memory of the drive
- Execution of a direct motion task
- Absolute trajectory, ip-Mode

22.4.1.3 Data transfer functions:

- Transmit a motion task to the drive's motion block memory. A motion task consists of these elements:
 - Position setpoint (absolute task) or path setpoint (relative task)
 - Speed setpoint
 - Acceleration time, braking time
 - Type of motion task (absolute/relative)
 - Number of a following task (with or without pause)
- Read a motion task from the motion block memory of the drive
- Read actual values
- Read the error register
- Read the status register
- Read/write control parameters

22.4.2 Transmission Rate and Procedure

- Bus connection and bus medium: CAN-standard ISO 11898 (CAN high-speed)
- Transmission rate: max. 1Mbit/s
- Possible settings for the drive: 125 (default), 250, 500 and 1000 kbit/s
- The baudrate is set with the AKD - parameter FBUS.PARAM01. It gets effective by saving this parameter to NVRAM and re-starting the drive.

22.4.3 Response to BUSOFF Communication Faults

The communication fault BUSOFF is directly monitored and signaled by Level 2 (CAN controller). This message may have various causes. A few examples:

- Telegrams are transmitted, although there is no other CAN node connected
- CAN nodes have different transmission rates
- The bus cable is faulty
- Faulty cable termination causes reflections on the cable.

A BUSOFF is only signaled by the AKD, if another CAN node is connected and at least one object was successfully transmitted to start off with. The BUSOFF condition is signaled by the error message 702. If the output stage is enabled at the moment when this fault occurs, the output stage is disabled.

22.4.4 Important Configuration Parameters

FBUS.PARAM01 see "Transmission Rate and Procedure"

FBUS.PARAM02 0 - no PLL used for synchronization

1 - PLL used for synchronized modes, IP (7), CSP (8), generates a warning n125, when PLL is unlocked

FBUS.PARAM04 0 - arrival of SYNC-messages in cyclic-synchronized application is not supervised

1 - arrival of SYNC-messages in cyclic-synchronized application is supervised (after 3 missing SYNC-telegrams the fault F125 is generated)

FBUS.PARAM05 description for bits 0 to 3 as in AKD - command reference

Bit 4 = 0: Scaling for position, velocity and acceleration objects is done via UNIT parameters

Bit 4 = 1: Scaling is done via special DS402 - objects (independent on units)

Bit 5 used in EtherCAT, reserved for CAN

Bit 6 = 0: Bit 0 of parameter MT.CNTL (object 35D9 sub 0) is exclusively used for DS402 controlword

Bit 6 = 1: Bit 0 of parameter MT.CNTL (object 35D9 sub 0) can be accessed

22.5 CANopen Communication Profile

22.5.1 General Description of CAN

This chapter describes the basic services and communication objects of the CANopen communication profile DS 301, which are used in the AKD.

NOTE

It is assumed that the basic operating functions of the communication profile are known, and available as reference documentation.

The transmission method that is used here is defined in ISO 11898 (Controller Area Network CAN for high-speed communication).

The Layer-1/2 protocol (Physical Layer/Data Link Layer) that is implemented in all CAN modules provides, amongst other things, the requirements for data.

Data transport or data request is made by means of a data telegram (Data Frame) with up to 8 bytes of user data, or by a data request telegram (Remote Frame).

Communication objects (COBs) are labeled by an 11-bit Identifier (ID) that also determines the priority of objects.

A Layer-7 protocol (Application Layer) was developed, to decouple the application from the communication. The service elements that are provided by the Application Layer make it possible to implement an application that is spread across the network. These service elements are described in the CAN Application Layer (CAL) for Industrial Applications.

The communication profile CANopen and the drive profile are mounted on the CAL.

The basic structure of a communication object is shown in the following diagram:

| | | | | | | | |
|---|--------|---|------|--------------|-----|---|-----|
| S | COB-ID | R | CTRL | Data Segment | CRC | A | EOM |
| O | | T | | | | C | |
| M | | R | | | | K | |

| | |
|--------------|---|
| SOM | Start of message |
| COB-ID | Communication Object Identifier (11-bit) |
| RTR | Remote Transmission Request |
| CTRL | Control Field (e.g. Data Length Code) |
| Data Segment | 0 to 8byte (Data-COB) 0byte (Remote-COB) |
| CRC | Cyclic Redundancy Check |
| ACK | Acknowledge slot |
| EOM | End of message |

22.5.2 Construction of the Communication Object Identifier

The following diagram shows the layout of the COB Identifier (COB-ID). The Function Code defines the interpretation and priority of the particular object.

| | | | | | | | | | | |
|---------------|---|---|---|-----------|---|---|---|---|---|---|
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Function-Code | | | | Module-ID | | | | | | |

Bit 0 .. 6

Module ID (drive's CAN-bus address, range 1 to 127; is set up in WorkBench or the drive,)

Bit 7 to 10

Function Code (number of the communication object that is defined in the server)

NOTE

If an invalid station number (=0) is set, then the module will be set internally to 1.

The following tables show the default values for the COB Identifier after switching on the drive. The objects, which are provided with an index (Communication Parameters at Index), can have a new ID assigned after the initialization phase. The indices in brackets are optional.

Predefined broadcast objects (send to all nodes):

| Object | Function code (binary) | Resulting COB-IDs | | Communication parameters at index |
|--------|------------------------|-------------------|------|-----------------------------------|
| | | Dec. | Hex. | |
| NMT | 0000 | 0 | 0 | — |
| SYNC | 0001 | 128 | 80 | (1005) |
| TIME | 0010 | 256 | 100 | not supported |

Predefined Peer-to-Peer objects (node sends to node):

| Object | Function code (binary) | Resulting COB-IDs | | Communication parameters at index | Priority |
|-----------|------------------------|-------------------|----------|-----------------------------------|----------|
| | | Dec. | Hex. | | |
| EMERGENCY | 0001 | 129..255 | 81..FF | — | high |
| TPDO 1 | 0011 | 385..511 | 181..1FF | 1800 | |
| RPDO 1 | 0100 | 513..639 | 201..27F | 1400 | |
| TPDO 2 | 0101 | 641..767 | 281..2FF | 1801 | |
| RPDO 2 | 0110 | 769..895 | 301..37F | 1401 | |
| TPDO 3 | 0110 | 897..1023 | 381..3FF | 1802 | |
| RPDO 3 | 1000 | 1025..1151 | 401..47F | 1402 | |
| TPDO 4 | 1001 | 1153..1279 | 481..4FF | 1803 | |
| RPDO 4 | 1010 | 1281..1407 | 501..57F | 1403 | |
| SDO (tx*) | 1011 | 1409..1535 | 581..5FF | | |
| SDO (rx*) | 1100 | 1537..1663 | 601..67F | | |
| Nodeguard | 1110 | 1793..1919 | 701..77F | (100E) | low |

*tx = direction of transmission: AKD => Master

rx = direction of transmission: Master => AKD

22.5.3 Definition of the Used Data Types

This chapter defines the datatypes that are used. Each datatype can be described by bit-sequences. These bit-sequences are grouped into "Octets" (bytes). The so-called "Little-Endian" format (a.k.a. Intel format) is used for numerical data types (see also: DS301 Application Layer "General Description of Data Types and Encoding Rules").

22.5.3.1 Basic data types

Unsigned Integer

Data in the basic data type UNSIGNEDn define exclusively positive integers.

The value range is from 0 to 2^n-1 . The bit sequence $b = b_0$ to b_{n-1} defines the value

$$\text{UNSIGNEDn}(b) = b_{n-1} 2^{n-1} + \dots + b_1 2^1 + b_0 2^0$$

Example: the value 266 = 10Ah is transmitted in the data type UNSIGNED16, in the form of two octets (1st octet = 0Ah, 2nd octet = 01h).

Transmission syntax for the data type UNSIGNEDn

| Octet number | 1. | 2. | 3. | 4. |
|--------------|----------------|-------------------|----------------------|----------------------|
| UNSIGNED8 | b_7 to b_0 | | | |
| UNSIGNED16 | b_7 to b_0 | b_{15} to b_8 | | |
| UNSIGNED24 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | |
| UNSIGNED32 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |
| UNSIGNED40 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |
| UNSIGNED48 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |
| UNSIGNED56 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |
| UNSIGNED64 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |

| Octet number | 5. | 6. | 7. | 8. |
|--------------|----------------------|----------------------|----------------------|----------------------|
| UNSIGNED8 | | | | |
| UNSIGNED16 | | | | |
| UNSIGNED24 | | | | |
| UNSIGNED32 | | | | |
| UNSIGNED40 | b_{39} to b_{32} | | | |
| UNSIGNED48 | b_{39} to b_{32} | b_{47} to b_{40} | | |
| UNSIGNED56 | b_{39} to b_{32} | b_{47} to b_{40} | b_{55} to b_{48} | |
| UNSIGNED64 | b_{39} to b_{32} | b_{47} to b_{40} | b_{55} to b_{48} | b_{63} to b_{56} |

Signed Integer

Data in the basic data type INTEGER_n define both positive and negative integers.

The value range is from $-2^{n-1}-1$ to $2^{n-1}-1$. The bit sequence $b = b_0$ to b_{n-1} defines the value

$$\text{INTEGER}_n(b) = b_{n-2} 2^{n-2} + \dots + b_1 2^1 + b_0 2^0 \text{ with } b_{n-1} = 0$$

Negative numbers are represented as 2's complement, which means:

$$\text{INTEGER}_n(b) = -\text{INTEGER}_n(b) - 1 \text{ with } b_{n-1} = 1$$

Example: the value -266 = FEF6h is transmitted in the data type INTEGER16, in the form of two octets (1st octet = F6h, 2nd octet = FEh).

Transmission syntax for the data type INTEGER_n

| Octet number | 1. | 2. | 3. | 4. |
|--------------|----------------|-------------------|----------------------|----------------------|
| INTEGER8 | b_7 to b_0 | | | |
| INTEGER16 | b_7 to b_0 | b_{15} to b_8 | | |
| INTEGER24 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | |
| INTEGER32 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |
| INTEGER40 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |
| INTEGER48 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |
| INTEGER56 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |
| INTEGER64 | b_7 to b_0 | b_{15} to b_8 | b_{23} to b_{16} | b_{31} to b_{24} |

| Octet number | 5. | 6. | 7. | 8. |
|--------------|----------------------|----------------------|----------------------|----------------------|
| INTEGER8 | | | | |
| INTEGER16 | | | | |
| INTEGER24 | | | | |
| INTEGER32 | | | | |
| INTEGER40 | b_{39} to b_{32} | | | |
| INTEGER48 | b_{39} to b_{32} | b_{47} to b_{40} | | |
| INTEGER56 | b_{39} to b_{32} | b_{47} to b_{40} | b_{55} to b_{48} | |
| INTEGER64 | b_{39} to b_{32} | b_{47} to b_{40} | b_{55} to b_{48} | b_{63} to b_{56} |

22.5.3.2 Mixed data types

Mixed data types combine basic data types (INTEGER_n, UNSIGNED_n, REAL). Two types of mixed data are distinguished:

- STRUCT: This data type is composed of elements with different data types.
- ARRAY: This data type is composed of elements of the same data type.

22.5.3.3 Extended data types

Extended data types are derived from basic data types and mixed data types. The types of extended data that are supported are defined below.

Octet String

The data type OCTET_STRING is defined with the data type ARRAY. Length is the length of the octet string.

```
ARRAY[length] OF UNSIGNED8    OCTET_STRINGlength
```

Visible String

The data type VISIBLE_STRING can be defined with the data type UNSIGNED8 or the data type ARRAY. Permissible values are 00h and the range from 20h to 7Eh. The data are interpreted as 7 bit ASCII code (as per ISO 646-1973(E)). Length is the length of the visible string.

```
UNSIGNED8                VISIBLE_CHAR
ARRAY[length] OF VISIBLE_CHAR  VISIBLE_STRINGlength
```

22.5.4 Communication Objects

Communication objects are described with the help of service elements and protocols. Two basic types of service elements are used.

- Unconfirmed services PDO
- Confirmed services SDO

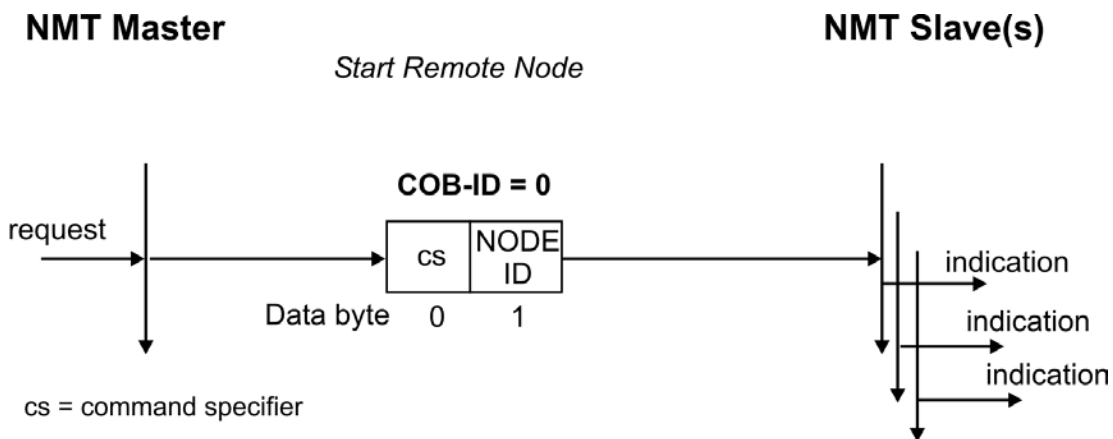
All services require faultless operation of the Data Link and Physical Layer.

AKD supports communication objects that are described in detail in the following sections:

- Network Management Objects (NMT)
- Synchronization Object (SYNC)
- Emergency Object (EMCY)
- Process Data Object (PDO)
- Service Data Object (SDO)
- Nodeguard/Heartbeat

22.5.4.1 Network Management Objects (NMT)

The following diagram describes the NMT telegram:



The drive supports the following network management functions:

cs = 129, reset node:

Causes a cold-start of the drive. This deletes all parameters saved in the RAM and loads the values stored in the EEPROM.

cs = 130, reset communication node:

Causes a stop of PDO-communication, gives a new bootup-message

cs = 1, start remote node:

Starts the CAN node. I.e. the PDOs of the drive are enabled for operation. From this moment, transmit-PDOs will be transmitted under event-control, and cyclical process data operation can commence.

cs = 2, stop remote node:

Stops the CAN node, I.e. the drive no longer responds to any received PDOs or transmits any PDOs.

22.5.4.2 Synchronization Object (SYNC)

The SYNC object usually is used as a periodic Broadcast Object and provides the basic clock for the bus. SYNC has a high priority, to ensure constant time intervals. The usage of this protocol is explained in the appendix from page . You can use the SYNC object to start motion task of several axes simultaneously for example.

22.5.4.3 Time-Stamp Object (TIME)

This communication object is not supported by the AKD.

22.5.4.4 Emergency Object (EMCY)

EMCY is event-triggered and generated by an internal fault/error situation. This object is transmitted afresh for every error. Since the error codes are device-dependent, they are described in the Chapter *“CANopen Emergency Messages and Error Codes”* (=> p. 160). The last 10 Emergency error codes can be read via object 1003.

Application of the Emergency Object

The reaction in the event of an error or fault depends on the error class and is therefore variable. For this reason, the reaction is described with the aid of an error status machine. The error conditions error-free and error occurred are distinguished. The following transitions are defined:

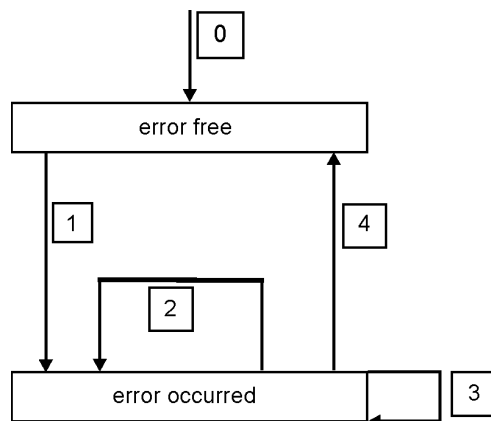
Transition 0: After initialization, the error-free status is taken up if no errors are detected. No error signal is generated in this condition.

Transition 1: The AKD detects an internal error and indicates this in the first three bytes of the emergency telegram (error code in Bytes 0,1 and error register in Byte 2).

Transition 2: One error has been reset, but not all. The EMCY telegram contains error code 0000 and the error register indicates the remaining errors that are present. The manufacture-specific area is set to zero.

Transition 3: A new error has occurred. The AKD remains in the error status and transmits an EMCY. Object with the corresponding error code. The new error code is entered into bytes 0 and 1.

Transition 4: All errors have been reset. The EMCY telegram contains the error code 0000, The error register does not indicate any other errors. The manufacture-specific area is set to zero.



Composition of the Emergency Object

The Emergency Object is composed of 8 bytes, divided as follows:

| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---------|----------------------|---|------------------------------|----------|----------|---|---|---|
| Content | Emergency error code | | Error register (object 1001) | Category | Reserved | | | |

If an Emergency Object is generated, the error condition is then signaled to the status machine (error free/error occurred) by the generation of a second Emergency Object. Only the first four bytes are relevant in this case (Emergency Error code, Error register, Category). Byte 0/1 contains the Emergency Error Code (0000) and Byte 2 indicates if a possible further error is present. If the error register contains 00, the error status is error-free. Byte 3 contains the category. The interpretations of the error numbers (error code) and the error categories are described in the section Emergency Messages. The error register is defined through object 1001. Error register’.

22.5.4.5 Service Data Objects (SDO)

SDOs are used to implement access to the Object Dictionary. The SDOs are required for parametrization and for status polling. Access to an individual object is made with a multiplexer via the Index and Subindex of the Object Dictionary. The following communication protocols are supported by AKD:

- Initiate SDO Download Protocol
- Download SDO Segment Protocol
- Initiate SDO Upload Protocol
- Upload SDO Segment Protocol
- Abort SDO Transfer Protocol

The definitions of the individual communication services and protocols can be found in DS301. Examples of the usage of SDOs can be found in the appendix from page .

NOTE

Since a SDO is a confirmed service, the system must always wait for the SDO response telegram before it is allowed to transmit a new telegram.

Composition of the Service Data Object

An SDO consists of the following components:

| Byte | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---------|-----|-------|----------|------|---|---|---|---|
| Content | R/W | Index | Subindex | Data | | | | |

1. The control byte (Byte 1):

The control byte determines whether the SDO should write or read the content of the entry in the Object Dictionary. A description of the complete Object Dictionary for AKD=> p. 223. Data exchange with the AKD is governed by the *CMS multiplexed domain protocols* standard, as described in the CAN standard DS 202.

To read data, the control byte must be written in the manner shown below:

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|-------|---|---|---|---|---|---|---|
| Content | ccs=2 | | X | X | X | X | X | X |

ccs => client command specifier (ccs = 2 => initiate upload request)

X => free data

So a value of 0100 0000 (binary) or 40h must be transmitted in the control byte.

The drive sends back a corresponding response byte:

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---------|-------|---|---|---|---|---|---|---|
| Content | scs=2 | | X | n | e | s | | |

scs => server command specifier (scs = 2 => initiate upload response)

n => only valid for e = s = 1, if this is so, n contains the number of bytes that do not contain data

X => free data

If reading is successful, the response byte always has set the bits 0 and 1 (e = s = 1).

Encoded byte length in the SDO response:

0x43 - 4 bytes

0x47 - 3 bytes

0x4B - 2 bytes

0x4F - 1 byte.

If an error occurs, scs is set to 4, the response byte is 0x80 and the error information is in the four byte data field. The decoding of the error => p. 160

To write data, the control byte must be written in the manner shown below:

| Client | <i>Initiate Domain Download</i> | | | | | | | | | | | Server | | | | |
|---------|---|---|---|---|---|---|---|---|---|---|---|--------|---|---|---|------------|
| | Byte 1 | | | | | | | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| request | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | indication |
| => | ccs=1 | | | X | n | | | e | s | m | | d | d | d | d | => |
| | => => => => => =>=> => =>=> => =>=> => =>=> => =>=> => =>=> => => | | | | | | | | | | | | | | | |

n,e and s are defined like in the reading case, m: index + Subindex, d: 4 bytes data field

The data length of an object can be taken from the object dictionary in the appendix.

The control byte should be:

0x23 for a 4-byte access

0x27 for a 3-byte access

0x2B for a 2-byte access

0x2F for a 1-byte access

| Client | <= <= <= <= <= <=<= <= <=<= <= <=<= <= <=<= <= <= | | | | | | | | | | | Server | | | | |
|---------|---|---|---|---|---|---|---|---|---|----------|---|--------|---|----|---|----------|
| | Byte 1 | | | | | | | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| confirm | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | response |
| <= | scs=3 | | | X | | | | m | | reserved | | | | <= | | |

2. Index (Bytes 2 and 3):

The Index is the main entry in the Object Dictionary, and divides the parameters into groups.

(Example: Index 1018h is the Identity Object). As for all CAN data, the Index is stored with the bytes in reverse order.

For example: Index 6040h means Byte 2 = 40h, Byte 3 = 60h)

3. Subindex (Byte 4):

The Subindex divides the parameters within a group of parameters.

4. Data field (Bytes 5 to 8):

These components are used for the exchange of user data. In read-request telegrams to the AKD they are set to 0. They have no content in a write confirmation from the AKD if the transfer was successful, but if the write operation was faulty they contain an error => p. 160.

Initiate SDO Download Protocol

The Initiate SDO Download protocol is used for write access to objects with up to 4 bytes of user data (expedited transfer) or to initiate a segment transfer (normal transfer).

Download SDO Segment Protocol

The Download SDO Segment protocol is used for write access to objects with more than 4 bytes of user data (normal transfer).

Initiate SDO Upload Protocol

The SDO Upload protocol is used for read access to objects with up to 4 bytes of user data (expedited transfer) or to initiate a segment transfer (normal transfer).

Upload SDO Segment Protocol

The Upload SDO Segment protocol is used for read access to objects with more than 4 bytes of user data (normal transfer).

Abort SDO Protocol

The Abort SDO protocol breaks off SDO transmission, and indicates the error that caused the break in transmission through an abort code (error code). The error code is in the format of an UNSIGNED32 value. The following table shows possible reasons for an abort SDO.

| Abort Code | Description |
|------------|---|
| 0504 0000h | SDO timeout |
| 0504 0001h | Command specifier invalid |
| 0504 0002h | SDO segmented: invalid blocksize |
| 0504 0004h | SDO segmented: invalid block CRC |
| 0504 0005h | SDO segmented: out of memory |
| 0601 0001h | Attempted read access to a write-only object |
| 0601 0002h | Attempted write access to a read-only object |
| 0602 0000h | Object does not exist in Object Dictionary |
| 0604 0041h | Object cannot be mapped to a PDO |
| 0604 0042h | Size and number of mapped objects exceed permissible PDO length |
| 0604 0043h | General parameter incompatibility |
| 0606 0000h | SDO hardware fault |
| 0607 0010h | Data type incompatible, length of service parameter is incompatible |
| 0609 0011h | Subindex does not exist |
| 0609 0030h | Outside value range for the parameter (only for write access) |
| 0609 0031h | Parameter value too high |
| 0609 0032h | Parameter value too low |
| 0800 0020h | Data cannot be transmitted or saved |
| 0800 0022h | Data cannot be transmitted or saved because of device status |

Abort Codes not listed above are reserved.

22.5.4.6 Process Data Object (PDO)

PDOs are used for real-time data communication. PDOs can, for instance, be used to set up controllers similar to analog drives. Instead of +/-10VDC setpoints and ROD feedback, digital speed setpoints and position feedback are attained via PDOs in this case.

Transmission is carried out unconfirmed without a protocol "overhead". This communication object uses the unconfirmed communication service.

PDOs are defined via the Object Dictionary for the AKD. Mapping is made during the configuration phase, with the help of SDOs. Length is defined with the mapped objects.

The definition of the PDO service and protocol can be found in DS301. Examples of the usage of PDOs can be found in the appendix.

Basically, two types of PDOs can be distinguished, depending on the direction of transmission:

- Transmit-PDOs (TPDOs) (AKD => Master)
The TPDOs transmit data from AKD to control system (for example actual value objects, instrument status).
- Receive-PDOs (RPDOs) (Master =>AKD)
The RPDOs receive data from control system to AKD (for example setpoints).

AKD supports four independent PDO channels for each direction of transmission. The channels are labeled by the channel numbers 1 to 4.

There are two parameter sets each for the configuration of each of the four possible PDOs, and they can be set up through the corresponding SDOs:

1. Mapping parameters, to determine which data are available (mapped) in the selected PDO and to define, which data are contained.
2. Communication parameters, that define whether the PDOs operate in synchronized mode, or event-driven (objects 1400h to 1403h, 1800h to 1803h).

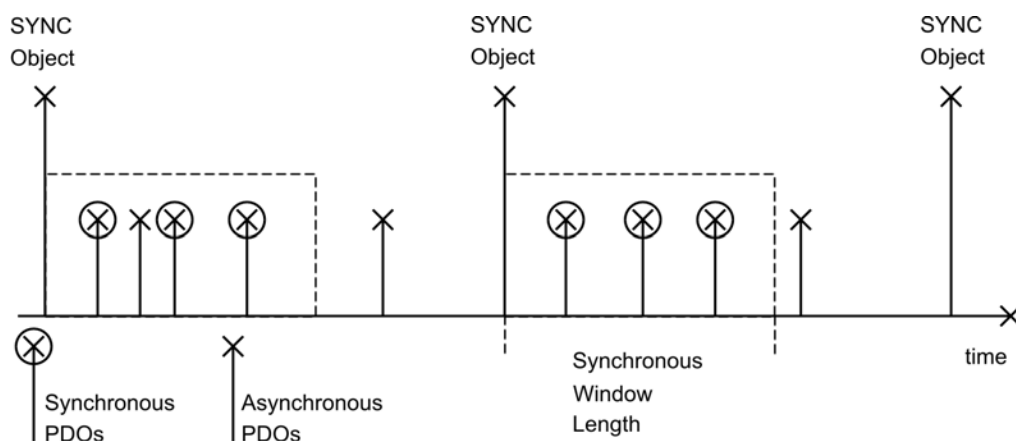
Transmission modes

The following PDO transmission modes are distinguished:

- Synchronous transmission
- Asynchronous transmission

The pre-defined SYNC Object is transmitted periodically (bus clock), to synchronize the drives. Synchronous PDOs are transmitted within a pre-defined time window immediately following the SYNC Object.

The transmission modes are set up with the aid of the PDO communication parameters.



Trigger modes

Three different trigger modes are distinguished:

- **Event driven:** The transmission of the telegrams is triggered by an object-specific event.
- **Time driven:** If event driven signals put a high strain on the bus, you can determine the period of time after which a PDO can be transmitted again via the inhibit time (Communication parameter, Subindex 03h)
- **Event Timer driven:** If a PDO shall be sent within a defined time interval, even if it doesn't change, this interval can be defined by a special SDO.

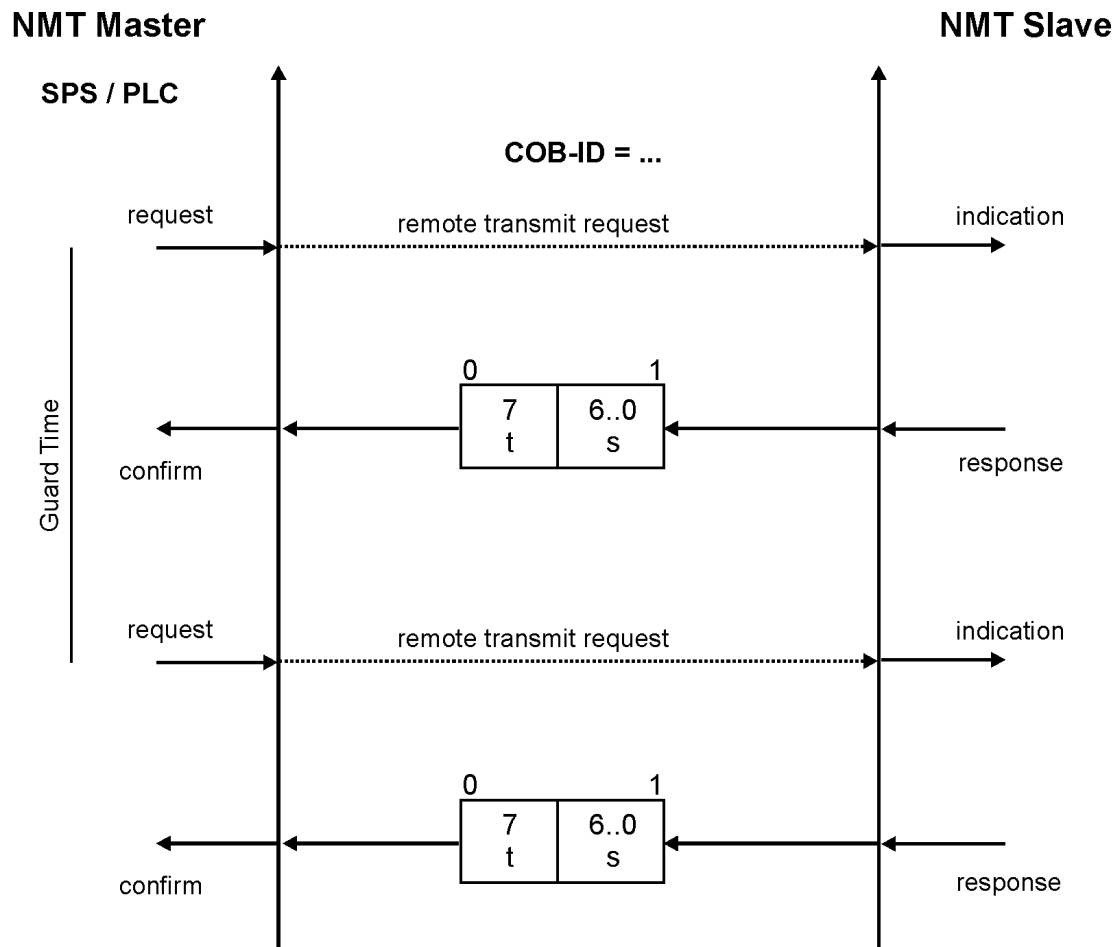
22.5.4.7 Nodeguard

The Node Guarding protocol is a functional monitoring for the drive. It requires that the drive is accessed at regular intervals by the CANopen master.

The maximum time interval that is permitted between two Nodeguard telegrams is given by the product of the Guard Time (Object 100Ch) and the Life Time Factor (Object 100Dh). If one of these two values is 0, then the response monitoring is de-activated.

If the drive is not accessed within the time defined by objects 100Ch and 100Dh, then fault F129 (response monitoring) appears on the drive, the drive is braked to a stop, and any other movement is prevented.

The time sequence for node guarding is as shown below:



t = toggle Bit, changes its status with every slave telegram

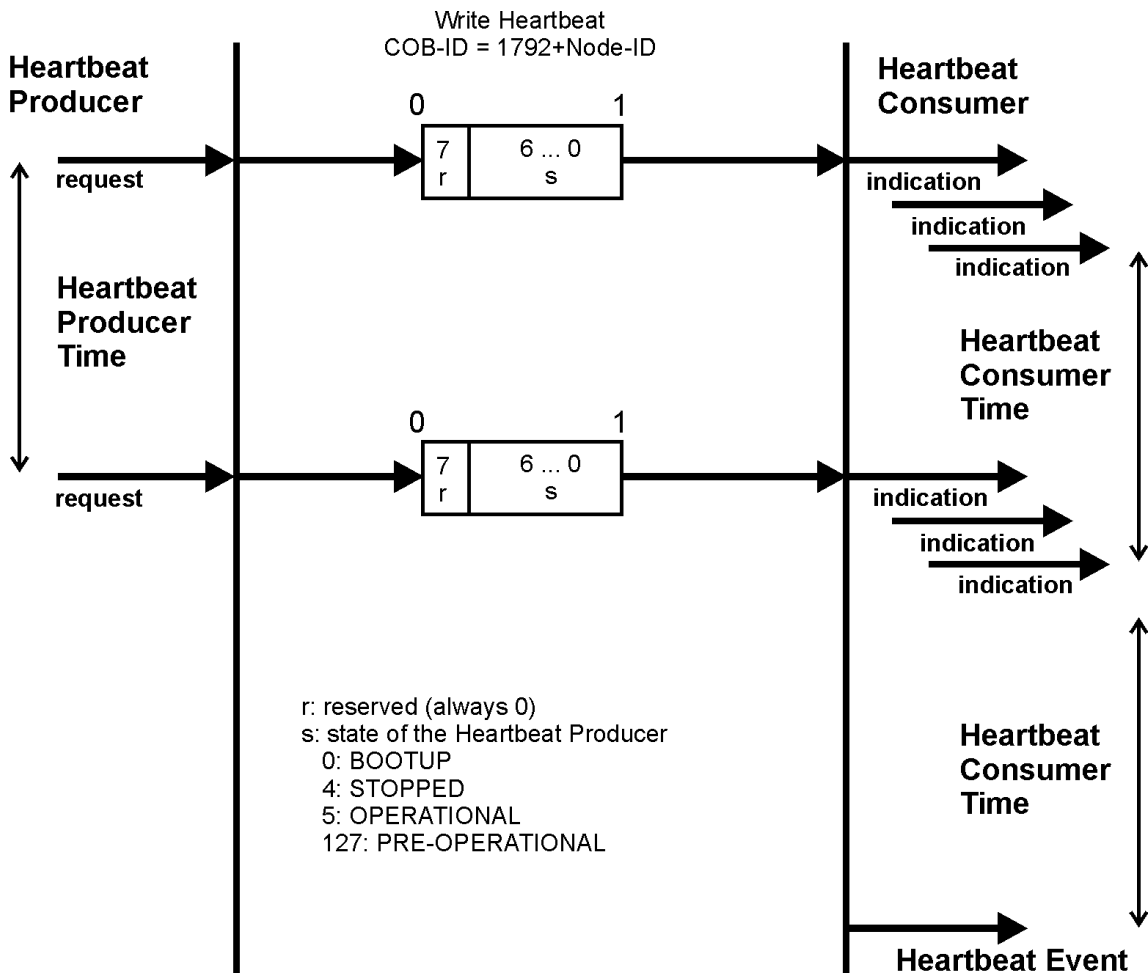
s = status of the NMT slave status machine

Node guarding is carried out by the Master through RTR telegrams with the COB-ID 700h + slave node address.

22.5.4.8 Heartbeat

The Heartbeat Protocol defines an Error Control Service without need for remote frames. A Heartbeat Producer transmits a Heartbeat message cyclically. One or more Heartbeat Consumer receive the indication. The relationship between producer and consumer is configurable via Object 1016h/1017h. The Heartbeat Consumer guards the reception of the Heartbeat within the Heartbeat Consumer Time. If the Heartbeat is not received within the Heartbeat Consumer Time a Heartbeat Event will be generated.

Heartbeat protocol:



22.6 CANopen Drive Profile

22.6.1 CANopen Emergency Messages and Error Codes

Emergency messages are triggered by internal equipment errors. They have a high ID-priority to ensure quick access to the bus. An emergency message contains an error field with pre-defined error/fault numbers (2 bytes), an error register (1byte), the error category (1 byte), and additional information.

Error numbers from 0000h to 7FFFh are defined in the communication or drive profile. Error numbers from FF00h to FFFFh have manufacturer-specific definitions. The following table describes the various error codes:

| Error Code | Fault/Warning Code | Description |
|------------|--------------------|--|
| 0x0000 | 0 | Emergency error free |
| 0x3110 | 523 | Bus over voltage |
| 0x3120 | - | Bus under voltage |
| 0x3130 | 503 | Phase failure |
| 0x3180 | 503 | Warning: Supply missing phase |
| 0x3210 | 501 | DC link over-voltage |
| 0x3220 | 502 | DC link under-voltage |
| 0x4210 | 234 | Excess temperature, device (control board) |
| 0x4310 | 235 | Excess temperature, drive (heat sink) |
| 0x5113 | 512 | Supply low voltage, U3 = supply +5 V |
| 0x5114 | 505 | Supply low voltage, manufacturer specific: supply 1V2 |
| 0x5115 | 507 | Supply low voltage, manufacturer specific: supply 2V5 |
| 0x5116 | 509 | Supply low voltage, manufacturer specific: supply 3V3 |
| 0x5117 | 514 | Supply low voltage, manufacturer specific: supply 12 V |
| 0x5118 | 516 | Supply low voltage, manufacturer specific: supply -12V |
| 0x5119 | 518 | Supply low voltage, manufacturer specific: supply 3V3 analogue |
| 0x5180 | 504 | Supply overvoltage, manufacturer specific: supply 1V2 |
| 0x5181 | 506 | Supply overvoltage, manufacturer specific: supply 2V5 |
| 0x5182 | 508 | Supply overvoltage, manufacturer specific: supply 3V3 |
| 0x5183 | 510 | Supply overvoltage, manufacturer specific: supply 5V0 |
| 0x5184 | 513 | Supply overvoltage, manufacturer specific: supply 12 V |
| 0x5185 | 104 | Supply overvoltage, manufacturer specific: supply -12V |
| 0x5186 | 517 | Supply overvoltage, manufacturer specific: supply 3V3 analogue |
| 0x5510 | 201 | Hardware memory, RAM |
| 0x5530 | 105 | Hardware memory, EEPROM |
| 0x5580 | 106 | Hardware memory, non-volatile memory data |
| 0x5581 | 202 | Hardware memory, external Ram for resident firmware failed |
| 0x5582 | 203 | Hardware memory, code integrity failed for resident firmware |
| 0x5583 | 102 | Hardware memory, resident firmware failed |
| 0x5584 | 103 | Hardware memory, resident FPGA failed |
| 0x5585 | 104 | Hardware memory, operational FPGA failed |
| 0x7180 | 301 | Motor overheat |
| 0x7182 | 305 | Motor open circuit |
| 0x7183 | 306 | Motor short circuit |
| 0x7184 | 307 | Motor brake applied |
| 0x7303 | 426 | Resolver 1 fault |
| 0x7305 | 417 | Incremental sensor 1 fault |
| 0x7380 | 402 | Feedback 1, analogue fault |
| 0x7381 | 403 | Feedback 1, EnDat communication fault |
| 0x7382 | 404 | Feedback 1, illegal hall |

| Error Code | Fault/Warning Code | Description |
|------------|--------------------|---|
| 0x7383 | 405 | Feedback 1, BiSS watchdog |
| 0x7384 | 406 | Feedback 1, BiSS multi cycle |
| 0x7385 | 407 | Feedback 1, BiSS sensor |
| 0x7386 | 408 | Feedback 1. SFD configuration |
| 0x7387 | 409 | Feedback 1, SFD UART overrun |
| 0x7388 | 410 | Feedback 1, SFD UART frame |
| 0x7389 | 412 | Feedback 1, SFD UART parity |
| 0x738A | 413 | Feedback 1. SFD transfer timeout |
| 0x738C | 415 | Feedback 1, SFD mult. corrupt position |
| 0x738D | 416 | Feedback 1, SFD Transfer incomplete |
| 0x738E | 418 | Feedback 1, power supply fault |
| 0x738F | 401 | Feedback 1, failed to set feedback |
| 0x73A0 | 424 | Feedback 2, resolver amplitude low |
| 0x73A1 | 425 | Feedback 2, resolver amplitude high |
| 0x73A2 | 425 | Feedback 2, resolver fault |
| 0x73A3 | 427 | Feedback 2, analogue low |
| 0x73A4 | 428 | Feedback 2, analogue high |
| 0x73A5 | 429 | Feedback 2, incremental low |
| 0x73A6 | 430 | Feedback 2, incremental high |
| 0x73A7 | 431 | Feedback 2, halls |
| 0x73A8 | 432 | Feedback 2, communication |
| 0x73A9 | - | Reserved |
| 0x73AA | - | Reserved |
| 0x8130 | 129 | Life Guard Error or Heartbeat Error |
| 0x8311 | 304 | Excess torque |
| 0x8331 | 524 | Torque fault |
| 0x8480 | 302 | Velocity overspeed |
| 0x8580 | 107 | Software limit switch, positive |
| 0x8581 | 108 | Software limit switch, negative |
| 0x8611 | 439 | Following error |
| 0x8686 | 151 | Not enough distance to move; Motion Exception |
| 0x8687 | 152 | Not enough distance to move; Following Motion Exception |
| 0x8688 | 153 | Velocity Limit Violation, Exceeding Max Limit |
| 0x8689 | 154 | Following Motion Failed; Check Motion Parameters |
| 0x868A | 156 | Target Position crossed due to Stop command |
| 0x86A0 | 157 | Homing Index pulse not found |
| 0x86A1 | 158 | Homing Reference Switch not found |
| 0x86A2 | 159 | Failed to set motion task parameters |
| 0x86A3 | 160 | Motion Task Activation Failed |
| 0x86A4 | 161 | Homing Procedure Failed |
| 0x8780 | 125 | Fieldbus synchronization lost |
| 0x8AF0 | 137 | Homing and Feedback mismatch |
| 0xFF00 | 701 | Fieldbus runtime fault |
| 0xFF01 | 702 | Fieldbus communication lost |
| 0xFF02 | 529 | Iu offset limit exceeded |
| 0xFF03 | 530 | Iv offset limit exceeded |
| 0xFF04 | 521 | Stored energy reached critical point |
| 0xFF05 | 527 | Iu detection stuck |

| Error Code | Fault/Warning Code | Description |
|------------|--------------------|--|
| 0xFF06 | 528 | Iv detection stuck |
| 0xFF07 | 525 | Control output over current |
| 0xFF08 | 526 | Current sensor short circuit |
| 0xFF09 | 128 | Axis dpoles |
| 0xFF0A | 531 | Power stage fault |
| 0xFF0B | 602 | Safe torque off |
| 0xFF10 | 414 | Warning: SFD single corrupted position. |
| 0x8380 | 524 | Warning: Drive foldback |
| 0x8387 | 304 | Warning: Motor foldback |
| 0xFF13 | 438 | Warning: Control position deviation |
| 0x8582 | 107 | Warning: Positive software position limit is exceeded. |
| 0x8583 | 108 | Warning: Negative software position limit is exceeded. |
| 0x8684 | 123 | Warning: Motion global warning |
| 0x7786 | 301 | Warning: Motor overheated. |
| 0x4380 | 234 | Warning: Control temperature sensor 1 high |
| 0x4387 | 235 | Warning: Power temperature sensor 1 high. |
| 0x4382 | 236 | Warning: Power temperature sensor 2 high. |
| 0x4393 | 237 | Warning: Power temperature sensor 3 high. |
| 0x4394 | 240 | Warning: Control temperature sensor 1 low. |
| 0x4385 | 241 | Warning: Control temperature sensor 1 low. |
| 0x4396 | 242 | Warning: Power temperature sensor 2 low. |
| 0x4392 | 243 | Warning: Power temperature sensor 3 low. |
| 0x3280 | 502 | Warning: Bus under voltage. |
| 0x3780 | 503 | Warning: Input phase loss. |
| 0x3287 | 521 | Warning: Dynamic Braking I ² T. |
| 0x8787 | 125 | Warning: Fieldbus PLL unlocked. |

22.6.2 General Definitions

This chapter describes objects with a general validity (e.g. Object 1000h Device Type). The next section explains the free configuration of Process Data Objects ("free mapping").

22.6.2.1 General objects

Object 1000h: Device Type (DS301)

This object describes the device type (servo drive) and device functionality (DS402 drive profile). Definition:

| | | | |
|------------------------|-------|-----------------------|---|
| MSB | | LSB | |
| Additional information | | Device profile number | |
| Mode bits | Type | 402d=192h | |
| 31 | 24 23 | 16 15 | 0 |

The device profile number is DS402, the type is 2 for drives, the mode bits 28 to 31 are manufacturer specific and may be changed from its actual value of 0. A read access delivers 0x00002192 at the moment.

| | |
|----------------------|--------------|
| Index | 1000h |
| Name | device type |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | no |

Object 1001h: Error register (DS301)

This object is an error register for the device. The device can map internal errors into this byte. It is a part of an Emergency object.

| | |
|----------------------|----------------|
| Index | 1001h |
| Name | Error register |
| Object code | VAR |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | UNSIGNED8 |
| Default value | no |

Error reasons to be signaled: If a bit is set to 1 the specified error has occurred. The generic error is signaled at any error situation.

| Bit | Description | Bit | Description |
|-----|---------------|-----|--|
| 0 | generic error | 4 | communication error (overrun, error state) |
| 1 | current | 5 | device profile specific |
| 2 | voltage | 6 | reserved (always 0) |
| 3 | temperature | 7 | manufacturer specific |

Object 1002h: Manufacturer Status Register (DS301)

The manufacturer status register contains important drive informations.

| | |
|----------------------|------------------------------|
| Index | 1002h |
| Name | Manufacturer Status Register |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Category | optional |
| Access | R/O |
| PDO mapping | possible |
| Value range | UNSIGNED32 |
| Default value | no |

The following table shows the bit assignment for the status register:

| Bit | Description | Bit | Description |
|-----|---|-----|---------------------------------------|
| 0 | 1 = Movement (positioning, homing) active | 16 | 1 = Homing move active |
| 1 | reserved | 17 | reserved |
| 2 | 1 = reference switch high (home-position) | 18 | reserved |
| 3 | 1 = In Position | 19 | 1 = Emergency stop active |
| 4 | reserved | 20 | reserved |
| 5 | reserved | 21 | reserved |
| 6 | reserved | 22 | reserved |
| 7 | reserved | 23 | 1 = Homing move finished |
| 8 | Warning active | 24 | reserved |
| 9 | 1 = target velocity reached (profile position mode) | 25 | 1 = digital input 1 set |
| 10 | reserved | 26 | 1 = digital input 2 set |
| 11 | 1 = Homing error | 27 | 1 = digital input 3 set |
| 12 | reserved | 28 | 1 = digital input 4 set |
| 13 | 1 = Safety selected | 29 | 1 = digital input hardware enable set |
| 14 | 1 = Power stage enabled | 30 | reserved |
| 15 | 1 = Error state | 31 | reserved |

Object 1003h: Predefined Error Field (DS301)

The object 1003h provides an error history with a maximum size of 10 entries.

Subindex 0 contains the number of errors which have occurred since the last reset of the error history, either by startup of the drive or resetting the error history by writing 0 to subindex 0.

A new Emergency-message is written into subindex 1 shifting the old entries one subindex higher. The old content of subindex 8 is lost.

The UNSIGNED32-information written to the subindexes is defined in the field Error Code in the description of the Emergency Messages (=> p. 160).

| | |
|----------------------|----------------------------------|
| Index | 1003h |
| Name | pre-defined Error Field |
| Object code | ARRAY |
| Data type | UNSIGNED32 |
| Category | optional |
| Subindex | 0 |
| Description | Number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/W |
| PDO mapping | not possible |
| Value range | 0 to 10 |
| Default value | 0 |
| Subindex | 1 to 10 |
| Description | Standard error field (=> p. 160) |
| Category | optional |
| Access | R/O |
| PDO mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | no |

Object 1005h: COB-ID of the SYNC Message (DS301)

This object defines the COB-Id of the synchronisation object (SYNC).

| | |
|----------------------|-----------------------------|
| Index | 1005h |
| Name | COB-ID for the SYNC message |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Category | conditional |
| Access | R/W |
| PDO mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | no |

Bit coded information:

| Bit | Value | Meaning |
|---------------|-------|----------------------------------|
| 31 (MSB) | X | — |
| 30 | 0 | Device not generate SYNC message |
| | 1 | Device generates SYNC message |
| 29 | 0 | 11 Bit ID (CAN 2.0A) |
| | 1 | 29 Bit ID (CAN 2.0B) |
| 28 to 11 | X | — |
| | 0 | if Bit 29=0 |
| 10 to 0 (LSB) | X | Bit 0 to 10 of SYNC COB-ID |

The device does not support the generation of SYNC-messages and only the 11-bit IDs. So the bits 11 to 30 are always 0.

Object 1006h: Communication Cycle Period (DS301)

This object can be used to define the period (in μs) for the transmission of the SYNC telegram.

| | |
|----------------------|-----------------------------------|
| Index | 1006h |
| Name | Period of the communication cycle |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Category | O |
| Access | R/W |
| PDO mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | 00h |

Object 1008h: Manufacturer Device Name (DS301)

The device name consists of four ASCII characters in the form Yzzz, whereby Y stands for the mains voltage (L, M, H or U, e.g. H for High Voltage) zzz stands for the power stage current.

| | |
|----------------------|--------------------------|
| Index | 1008h |
| Name | Manufacturer Device Name |
| Object code | VAR |
| Data type | Visible String |
| Category | Optional |
| Access | const |
| PDO mapping | not possible |
| Value range | |
| Default value | no |

Object 1009h: Manufacturer Hardware Version

This object will be supported in the future.

| | |
|----------------------|-------------------------------|
| Index | 1009h |
| Name | manufacturer hardware version |
| Object code | VAR |
| Data type | Visible String |
| Category | Optional |
| Access | const |
| PDO mapping | not possible |
| Value range | - |
| Default value | no |

Object 100Ah: Manufacturer Software Version (DS301)

The object contains the manufacturer software version (here: the CANopen-part of the drive firmware).

| | |
|----------------------|-------------------------------|
| Index | 100Ah |
| Name | Manufacturer Software Version |
| Object code | VAR |
| Data type | Visible String |
| Category | Optional |
| Access | const |
| PDO mapping | not possible |
| Value range | 0.01 to 9.99 |
| Default value | no |

Object 100Ch: Guard Time (DS301)Response monitoring

The arithmetical product of the Objects 100Ch Guard Time and 100Dh Lifetime Factor is the response monitoring time. The Guard Time is given in milliseconds. The response monitoring is activated with the first Nodeguard object. If the value of the object Guard Time is set to zero, then the response monitoring is inactive.

| | |
|----------------------|---|
| Index | 100Ch |
| Name | Guard Time |
| Object code | VAR |
| Data type | UNSIGNED16 |
| Category | conditional; mandatory, if heartbeat not supported |
| Access | R/W |
| PDO mapping | not possible |
| Value range | UNSIGNED16 |
| Default value | 0 |

Object 100Dh: Lifetime Factor (DS301)

The product of Guard Time and Life Time Factor gives the life time for the nodeguarding protocol. If it's 0, the protocol is not used.

| | |
|----------------------|--|
| Index | 100Dh |
| Name | Lifetime Factor |
| Object code | VAR |
| Data type | UNSIGNED8 |
| Category | conditional; (mandatory, if heartbeat not supported) |
| Access | R/W |
| PDO mapping | not possible |
| Value range | UNSIGNED8 |
| Default value | 0 |

Object 1010h: Store Parameters (DS301)

This object supports the saving of parameters to a flash EEPROM. Only the subindex 1 for saving of all parameters, which can also be saved in the parameter files via the GUI, is supported.

| | |
|----------------------|-------------------------------|
| Index | 1010h |
| Name | store parameters (DRV.NVSAVE) |
| Object code | ARRAY |
| Data type | UNSIGNED32 |
| Category | optional |
| Subindex | 0 |
| Name | number of entries |
| Object code | VAR |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO Mapping | not possible |
| Value range | 1 |
| Default value | 1 |
| Subindex | 1 |
| Name | save all parameters |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | 1 |

Data definition:

| Bit | Value | Meaning |
|---------|-------|--|
| 31 to 2 | 0 | reserved (=0) |
| 1 | 0 | Device does not save parameters autonomously |
| | 1 | Device does save parameters autonomously |
| 0 | 0 | Device does not save parameters on command |
| | 1 | Device does not save parameters on command |

By read access to subindex 1 the drive provides information about its storage functionality.

This drive provides a constant value of 1 by read access, i.e. all parameters can be saved by writing to Object 1010 sub 1. In general the drive does not save parameters autonomously with the exception of e.g. the special treatment of the homing of multiturn absolute encoders.

Storing of parameters is only done if a special signature ("save") is written to subindex 1. "save" is equivalent to the unsigned32 - number 65766173h.

Object 1014h: COB-ID for Emergency Message (DS301)

This object defines the COB-ID of the Emergency message.

| | |
|----------------------|---|
| Index | 1014h |
| Name | COB-ID emergency message |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Category | conditional; mandatory, if Emergency is supported |
| Access | R/O |
| PDO mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | 80h + Node - ID |

Object 1016h: Consumer Heartbeat Time

The consumer heartbeat time defines the expected heartbeat cycle time (ms) and must be higher than the corresponding producer heartbeat time configured on the device producing this heartbeat. Monitoring starts after the reception of the first heartbeat. If the consumer heartbeat time is 0 ms the corresponding entry is not used.

| | |
|----------------------|-------------------------|
| Index | 1016h |
| Name | consumer heartbeat time |
| Object code | ARRAY |
| Data type | UNSIGNED32 |
| Category | optional |
| Subindex | 0 |
| Description | number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO Mapping | not possible |
| Value range | 1 |
| Default value | 1 |
| Subindex | 1 |
| Description | Consumer heartbeat time |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | not possible |
| Value range | unsigned 32 |
| Default value | no |

Definition of the entry value of Subindex 1

| | | | |
|-------------------|----------------------|-----------|----------------|
| | MSB | | LSB |
| Value | reserved (value: 00) | Node-ID | heartbeat time |
| Encoded as | - | UNSIGNED8 | UNSIGNED16 |
| Bit | 31 | 24 23 | 16 15 0 |

Object 1017h: Producer Heartbeat Time

The producer heartbeat time defines the cycle time of the heartbeat in ms. If it's 0, it is not used.

| | |
|----------------------|---|
| Index | 1017h |
| Name | Producer heartbeat time |
| Object code | VAR |
| Data type | UNSIGNED16 |
| Category | conditional; mandatory, if guarding is not supported |
| Access | R/W |
| PDO mapping | not possible |
| Value range | UNSIGNED16 |
| Default value | 0 |

Object 1018h: Identity Object (DS301)

The Identity Object contains general device information.

| | |
|----------------------|-------------------|
| Index | 1018h |
| Name | Identity Object |
| Object code | RECORD |
| Data type | Identity |
| Category | mandatory |
| Subindex | 0 |
| Description | Number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | 1 to 4 |
| Default value | 4 |

Subindex 1 is a unique number for a device manufacturer.

| | |
|----------------------|------------------------|
| Subindex | 1 |
| Description | Vendor ID |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | 0x6Ah (Danaher Motion) |

Subindex 2 contains four ASCII - characters, which determine the voltage range and current class of the device. The voltage range is one character L, M or H for low, medium and high voltage. The next three characters are showing the continuous current of the drive.

| | |
|----------------------|----------------------------|
| Subindex | 2 |
| Description | Product Code |
| Category | optional |
| Access | R/O |
| PDO mapping | not possible |
| Value range | e.g. M006 for an MV6 drive |
| Default value | no |

Subindex 3 consists of two revision numbers:

- the major revision number in the upper word containing the CAN-version
- the minor revision number is not used in the AKD. The firmware version can be retrieved as a string via object 0x100A or as numbers via object 0x2018 subindex 1 to 4.

E.g. a value of 0x0014 0000 means CAN-version 0.20.

| | |
|----------------------|-----------------|
| Subindex | 3 |
| Description | Revision Number |
| Category | optional |
| Access | R/O |
| PDO mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | no |

Subindex 4 gives the serial number of the drive. This number contains the following information in it:

- bits 0..14: Board serial number (production in week of year)
- bits 15..20: week of production
- bits 21..24: year of production - 2009
- bits 25..31: ASCII-code of MFR-ID

| | |
|----------------------|---------------|
| Subindex | 4 |
| Description | Serial Number |
| Category | optional |
| Access | R/O |
| PDO mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | no |

Object 1026h: OS Prompt

The OS prompt is used to build up an ASCII - communication channel to the drive.

| | |
|--------------------|--------------|
| Index | 1026h |
| Name | OS Prompt |
| Object code | ARRAY |
| Data type | UNSIGNED8 |
| Category | optional |

| | |
|----------------------|-------------------|
| Subindex | 0 |
| Description | Number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | 2 |
| Default value | 2 |

Subindex 1 is used to send one character to the drive.

| | |
|----------------------|-----------|
| Subindex | 1 |
| Description | StdIn |
| Category | mandatory |
| Access | W |
| PDO mapping | possible |
| Value range | UNSIGNED8 |
| Default value | — |

Subindex 2 is used to receive one character from the drive.

| | |
|----------------------|-----------|
| Subindex | 2 |
| Description | StdOut |
| Category | mandatory |
| Access | R/O |
| PDO mapping | possible |
| Value range | UNSIGNED8 |
| Default value | 0 |

22.6.2.2 Manufacturer specific objects

Object 2014-2017h: 1st-4th Mask 1 to 4 for Transmit-PDO

In order to reduce the bus loading with event-triggered PDOs, masking can be used to switch off the monitoring for individual bits in the PDO. In this way it can be arranged, for instance, that actual position values are only signaled once per turn.

This Object masks the PDO-channels 1 to 4. If only two bytes have been defined in a PDO, then it masks just two bytes, although 4 bytes of mask information have been transmitted.

An activated bit in the mask means that monitoring is active for the corresponding bit in the PDO.

| | |
|----------------------|--|
| Index | 2014h 2015h 2016h 2017h |
| Name | tx_mask 1 to 4 |
| Object code | ARRAY |
| Data type | UNSIGNED32 |
| Subindex | 1 |
| Description | tx_mask1 to 4_low |
| Mode | independent |
| Access | R/W |
| PDO mapping | not possible |
| Unit | — |
| Value range | UNSIGNED32 |
| Default value | FFFFFFFFh |
| Subindex | 2 |
| Description | tx_mask1 to 4_high |
| Mode | independent |
| Access | R/W |
| PDO mapping | not possible |
| Unit | — |
| Value range | UNSIGNED32 |
| Default value | FFFFFFFFh |

Object 2018h: Firmware Version

This object gives all information regarding the firmware version.

Example: Firmware version M_01_00_01_005 would show the numbers 1, 0, 1, 5 in the sub-indices 1 to 4.

| | |
|----------------------|------------------|
| Index | 2018h |
| Name | firmware version |
| Object code | ARRAY |
| Data type | UNSIGNED16 |
| Subindex | 1 |
| Description | major version |
| Mode | independent |
| Access | R/O |
| PDO mapping | not possible |
| Unit | — |
| Value range | UNSIGNED16 |
| Default value | 0 |
| Subindex | 2 |
| Description | minor version |
| Mode | independent |
| Access | R/O |
| PDO mapping | not possible |
| Unit | — |
| Value range | UNSIGNED16 |
| Default value | 0 |
| Subindex | 3 |
| Description | revision |
| Mode | independent |
| Access | R/O |
| PDO mapping | not possible |
| Unit | — |
| Value range | UNSIGNED16 |
| Default value | 0 |
| Subindex | 4 |
| Description | branch revision |
| Mode | independent |
| Access | R/O |
| PDO mapping | not possible |
| Unit | — |
| Value range | UNSIGNED16 |
| Default value | 0 |

Object 2026h: ASCII Channel

This object is used to build up an ASCII - communication channel to the drive with 4-byte ASCII-strings.

| | |
|----------------------|-------------------|
| Index | 2026h |
| Name | ASCII Channel |
| Object code | ARRAY |
| Data type | Visible String |
| Category | optional |
| Subindex | 0 |
| Description | Number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | 2 |
| Default value | 2 |

Subindex 1 is used to send four ASCII-characters to the drive.

| | |
|----------------------|----------------|
| Subindex | 1 |
| Description | Command |
| Category | mandatory |
| Access | wo |
| PDO mapping | no |
| Value range | Visible String |
| Default value | — |

Subindex 2 is used to receive four characters from the drive.

| | |
|----------------------|----------------|
| Subindex | 2 |
| Description | Response |
| Category | mandatory |
| Access | R/O |
| PDO mapping | no |
| Value range | Visible String |
| Default value | - |

Object 20A0h: Latch position 1, positive edge

This object is used to output the position or a time, depending on CAP0.MODE, at which the first positive edge occurred on a signal, which can be configured with the command CAP0.TRIGGER. The latch enable must be active for that purpose(see object 20A4 and 20A5). With CAP0.MODE = 3 the latched position of the encoder index pulse is transferred via this object.

| | |
|----------------------|--|
| Index | 20A0h |
| Name | Latch position 1, positive edge, CAP0.PLFB Time capture, CAP0.T |
| Object code | VAR |
| Data type | INTEGER32 |
| Category | optional |
| Access | R/O |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | 0 |

Object 20A1h: Latch position 1, negative edge

This object is used to output the position or a time, depending on CAP0.MODE, at which the first negative edge occurred on a signal, which can be configured with the command CAP0.TRIGGER. The latch enable must be active for that purpose(see object 20A4 and 20A5).

| | |
|----------------------|--|
| Index | 20A1h |
| Name | Latch position 1, negative edge, CAP0.PLFB Time capture, CAP0.T |
| Object code | VAR |
| Data type | INTEGER32 |
| Category | optional |
| Access | R/O |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | 0 |

Object 20A2h: Latch position 2, positive edge

This object is used to output the position or a time, depending on CAP1.MODE, at which the first positive edge occurred on a signal, which can be configured with the command CAP1.TRIGGER. The latch enable must be active for that purpose(see object 20A4 and 20A5).

| | |
|----------------------|--|
| Index | 20A2h |
| Name | Latch position 2, positive edge, CAP1.PLFB Time capture, CAP1.T |
| Object code | VAR |
| Data type | INTEGER32 |
| Category | optional |
| Access | R/O |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | 0 |

Object 20A3h: Latch position 2, negative edge

This object is used to output the position or a time, depending on CAP0.MODE, at which the first negative edge occurred on a signal, which can be configured with the command CAP0.TRIGGER. The latch enable must be active for that purpose(see object 20A4 and 20A5).

| | |
|----------------------|--|
| Index | 20A3h |
| Name | Latch position 2, negative edge, CAP1.PLFB Time capture, CAP1.T |
| Object code | VAR |
| Data type | INTEGER32 |
| Category | optional |
| Access | R/O |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | 0 |

Object 20A4h: Latch Control Register

The latch control register is used to enable the latch monitoring of the capture engines 0 and 1. The latch is enabled with a 1 signal and disabled with a 0 signal. Whether or not a latch event has occurred can be recognised by the latch status register (object 20A5).

| | |
|----------------------|------------------------|
| Index | 20A4h |
| Name | Latch Control Register |
| Object code | VAR |
| Data type | UNSIGNED16 |
| Category | optional |
| Access | rww |
| PDO mapping | possible |
| Value range | 0 to 15 |
| Default value | 0 |

| Bit | Value (bin) | Value (hex) | Description |
|----------|-------------------|-------------|--|
| 0 | 00000000 00000001 | xx01 | Enable extern latch 1 (positive rise) |
| 1 | 00000000 00000010 | xx02 | Enable extern latch 1 (negative rise) |
| 2 | 00000000 00000100 | xx04 | Enable extern latch 2 (positive rise) |
| 3 | 00000000 00001000 | xx08 | Enable extern latch 2 (negative rise) |
| 4 | 00000000 00010000 | xx10 | Enable latch of encoder index pulse |
| 5 to 7 | | | Reserve |
| 8 | 00000001 00000000 | 01xx | Read external latch 1 (positive rise) |
| 9 | 00000010 00000000 | 02xx | Read external latch 1 (negative rise) |
| 10 | 00000011 00000000 | 03xx | Read external latch 2 (positive rise) |
| 11 | 00000100 00000000 | 04xx | Read external latch 2 (negative rise) |
| 12 | 00000101 00000000 | 05xx | Read latched position of encoder index pulse |
| 13 to 15 | | | Reserve |

Object 20A5h: Latch Status Register

The latch status register is used to look for the states of the capture engines 0 and 1.

| | |
|----------------------|-----------------------|
| Index | 20A5h |
| Name | Latch Status Register |
| Object code | VAR |
| Data type | UNSIGNED16 |
| Category | optional |
| Access | rwr |
| PDO mapping | possible |
| Value range | - |
| Default value | 0 |

| Bit | Value (bin) | Value (hex) | Description |
|----------|-------------------|-------------|--|
| 0 | 00000000 00000001 | zz01 | External latch 1 valid (positive rise) |
| 1 | 00000000 00000010 | zz02 | External latch 1 valid (negative rise) |
| 2 | 00000000 00000100 | zz04 | External latch 2 valid (positive rise) |
| 3 | 00000000 00001000 | zz08 | External latch 2 valid (negative rise) |
| 4 | 00000000 00010000 | z10 | Latched position of encoder index pulse valid (positive rise) |
| 5 to 7 | | | Reserve |
| 8 to 11 | 00000001 00000000 | z1zz | Acknowledge value external latch 1 (positive rise) |
| | 00000010 00000000 | z2zz | Acknowledge value external latch 1 (negative rise) |
| | 00000011 00000000 | z3zz | Acknowledge value external latch 2 (positive rise) |
| | 00000100 00000000 | z4zz | Acknowledge value external latch 2 (negative rise) |
| | 00000101 00000000 | z5zz | Acknowledge value of latched position of encoder index pulse (positive rise) |
| 12 to 15 | 00010000 00000000 | 1zzz | State Digital Input 4 |
| | 00100000 00000000 | 2zzz | State Digital Input 3 |
| | 01000000 00000000 | 4zzz | State Digital Input 2 |
| | 10000000 00000000 | 8zzz | State Digital Input 1 |

Object 20A6h: Latch position 1, positive or negative edge

This object is used to output the position or a time, depending on CAP0.MODE, at which the first positive or negative edge occurred on a signal, which can be configured with the command CAP0.TRIGGER.

The latch enable must be active for that purpose (see object 20A4 and 20A5).

| | |
|----------------------|---|
| Index | 20A6h |
| Name | Latch position 1, positive or negative, CAP0.PLFB |
| Object code | VAR |
| Data type | INTEGER32 |
| Category | optional |
| Access | ro |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | 0 |

Object 20B8h: Reset of changed input information

This object is used in PDOs to reset the state change information for the digital inputs shown in the Bits 24 to 30 in the object 60FD. Bit 0 to 6 are used to reset the information of the digital input 1 to 7.

| | |
|----------------------|------------------------------------|
| Index | 20B8h |
| Name | Reset of changed input information |
| Object code | VAR |
| Data type | UNSIGNED16 |
| Category | optional |
| Access | rw |
| PDO mapping | possible |
| Value range | UNSIGNED16 |
| Default value | 0 |

22.6.2.3 Profile specific objects

Object 60FDh: Digital inputs (DS402)

This index defines simple digital inputs for drives. The manufacturer bits 16 to 22 are used to mirror the digital inputs 1 to 7. The manufacturer bits 24 to 30 are used to show the change of the state of the digital inputs 1 to 7.

| | |
|----------------------|----------------|
| Index | 60FDh |
| Name | digital inputs |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Category | optional |
| Access | R/O |
| PDO mapping | possible |
| Value range | UNSIGNED32 |
| Default value | 0 |

| | | | | | | | |
|-----------------------|-----------|----|-----------|---|-------------|-------------------|-------------------|
| 31 | 16 | 15 | 4 | 3 | 2 | 1 | 0 |
| manufacturer specific | interlock | | interlock | | home switch | pos. limit switch | neg. limit switch |
| MSB | | | | | | | LSB |

Object 60FEh: Digital outputs (DS402)

This index defines simple digital outputs for drives. The manufacturer bits 16 and 17 are used to mirror the digital outputs 1 and 2.

| | |
|----------------------|-------------------|
| Index | 60FEh |
| Name | digital outputs |
| Object code | Array |
| Data type | UNSIGNED32 |
| Category | optional |
| Subindex | 0 |
| Description | number of entries |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | 2 |
| Default value | 2 |

| | | | | | | |
|-----------------------|------------------|-------|----------|----|-----------|-----|
| Subindex | 1 | | | | | |
| Description | physical outputs | | | | | |
| Category | mandatory | | | | | |
| Access | R/W | | | | | |
| PDO mapping | possible | | | | | |
| Value range | UNSIGNED32 | | | | | |
| Default value | 0 | | | | | |
| Subindex | 2 | | | | | |
| Description | bit mask | | | | | |
| Category | optional | | | | | |
| Access | R/W | | | | | |
| PDO mapping | possible | | | | | |
| Value range | UNSIGNED32 | | | | | |
| Default value | 0 | | | | | |
| 31 | 18 | 17 | 16 | 15 | 1 | 0 |
| manufacturer specific | DOUT2 | DOUT1 | reserved | | set brake | |
| MSB | | | | | | LSB |

Object 6502h: Supported drive modes (DS402)

A drive can support more than one and several distinct modes of operation. This object gives an overview of the implemented operating modes in the device. This object is read only.

| | |
|----------------------|-----------------------|
| Index | 6502h |
| Name | supported drive modes |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Category | optional |
| Access | R/O |
| PDO mapping | possible |
| Value range | UNSIGNED32 |
| Default value | 0x65 (ip hm pv pp) |

| | | | | | | | | | | |
|-----------------------|----------|----|----|----|----------|----|----|----|----|-----|
| 31 | 16 | 15 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| manufacturer specific | reserved | | ip | hm | reserved | tq | pv | vl | pp | |
| MSB | | | | | | | | | | LSB |

22.6.3 PDO Configuration

PDOs are used for process data communication. There are two types of PDOs: Receive PDOs (RPDOs) and transmit PDOs (TPDOs).

The content of the PDOs is pre-defined (see descriptions on pages => p. 185 and => p. 188). If the data content is not appropriate for a special application the data objects in the PDOs can be remapped freely.

One data entry in the PDOs looks like this:

| | | |
|----------------|------------------|-----------------------------|
| MSB | | LSB |
| index (16 bit) | Subindex (8 bit) | data length in bits (8 bit) |

The configuration procedure for a free mapping of a PDO looks like this (example for TPDO1):

1. Stop possible transmission of the PDO.

| COB-ID | Control byte | Index | | Sub- | Data | Comment |
|--------|--------------|----------|-----------|-------|-------------|-------------------|
| | | Low byte | High byte | index | | |
| 601 | 23 | 00 | 18 | 01h | 81 01 00 C0 | Switch-off COB-Id |

2. Delete the actual mapping of the PDO by writing a 0 to the subindex 0 of the mapping Object.

| COB-ID | Control byte | Index | | Sub- | Data | Comment |
|--------|--------------|----------|-----------|-------|-------------|-----------------------|
| | | Low byte | High byte | index | | |
| 601 | 2F | 00 | 1A | 00h | 00 00 00 00 | Delete actual mapping |

3. Build the mapping with object dictionary objects (see page => p. 223)) which are mappable, e.g.

| COB-ID | Control byte | Index | | Sub- | Data | Comment |
|--------|--------------|----------|-----------|-------|-------------|---|
| | | Low byte | High byte | index | | |
| 601 | 23 | 00 | 1A | 01h | 10 00 41 60 | 1st entry: CANopen statusword with 16 bits |
| 601 | 23 | 00 | 1A | 02h | 20 00 02 10 | 2nd entry: Manufacturer status with 32 bits |

3. Write the number of mapped objects to subindex 0 of the mapping Object.

| COB-ID | Control byte | Index | | Sub- | Data | Comment |
|--------|--------------|----------|-----------|-------|-------------|--|
| | | Low byte | High byte | index | | |
| 601 | 2F | 00 | 1A | 00h | 02 00 00 00 | Check for the right number of entries |

Mapping shall be done before the network management is switched to OPERATIONAL.

22.6.3.1 Receive PDOs (RXPDO)

Four Receive PDOs can be configured in the drive:

- - configuration of the communication (Objects 1400-1403h)
- - configuration of the PDO-contents (mapping, Objects 1600-1603h)

Objects 1400-1403h: 1st - 4th RXPDO communication parameter (DS301)

| | |
|--------------------|--|
| Index | 1400h to 1403h for RXPDO 1 to 4 |
| Name | receive PDO parameter |
| Object code | RECORD |
| Data type | PDO CommPar |
| Category | mandatory |

Defined sub-indices

| | |
|----------------------|--|
| Subindex | 0 |
| Name | number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO Mapping | not possible |
| Value Range | 2 |
| Default Value | 2 |
| Subindex | 1 |
| Name | COB-ID used by PDO |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | UNSIGNED32 |
| Default Value | Index 1400h: 200h + Node-ID Index 1401h: 300h + Node-ID Index 1402h: 400h + Node-ID Index 1403h: 500h + Node-ID |

Subindex 1 contains the COB-Id of the PDO as a bit coded information:

| Bit | Value | Meaning |
|----------|-------|--|
| 31 | 0 | PDO exists/is valid |
| | 1 | PDO does not exist/is not valid |
| 30 | 0 | RTR allowed on this PDO, not to be used (Can in Automation organisation) |
| | 1 | RTR not allowed on this PDO |
| 29 | 0 | 11 bit-ID (CAN 2.0A) |
| | 1 | 29 bit-ID (CAN 2.0B), not supported |
| 28 to 11 | X | Identifier-bits with 29 bit-ID, not relevant |
| 10 to 0 | X | Bits 10-0 of COB-ID |

| | |
|----------------------|-------------------|
| Subindex | 2 |
| Name | transmission type |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | UNSIGNED8 |
| Default Value | FFh |

Subindex 2 contains the transmission type of the PDO. There are two ways of setting:

- the value FFh or 255 for event-triggered PDO, which is directly interpreted by reception and taken into actions,
- values from 0 to 240, which cause a SYNC-telegram-controlled interpretation of the PDO contents. Values of 1 to 240 mean, that 0 to 239 SYNC-telegrams are ignored, before one is interpreted. The value 0 means, that only the next SYNC-telegram is interpreted.

Objects 1600-1603h: 1st - 4th RXPDO mapping parameter (DS301)

| | |
|----------------------|---|
| Index | 1600h - 1603h for RXPDO 1 .. 4 |
| Name | receive PDO mapping |
| Object Code | RECORD |
| Data Type | PDO Mapping |
| Category | mandatory |
| Subindex | 0 |
| Name | number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | 0: PDO is not active 1 - 8: PDO activated, mappings are taken only byte-wise |
| Default Value | PDO1: 1 PDO2: 2 PDO3: 2 PDO4: 2 |
| Subindex | 1 - 8 |
| Name | PDO - mapping for the n-th application object |
| Category | Conditional, depends on number and size of object be mapped |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | UNSIGNED32 |
| Default Value | See below |

Default RXPDO definition

RXPDO 1:

| Subindex | Value | Meaning |
|----------|-------------|-----------------------|
| 0 | 1 | One PDO-mapping entry |
| 1 | 60 40 00 10 | Control word |

RXPDO 2:

| Subindex | Value | Meaning |
|----------|-------------|-------------------------|
| 0 | 2 | Two PDO-mapping entries |
| 1 | 60 40 00 10 | Control word |
| 2 | 60 60 00 08 | Modes of Operation |

RXPDO 3:

| Subindex | Value | Meaning |
|----------|-------------|---------------------------|
| 0 | 2 | Two PDO-mapping entries |
| 1 | 60 40 00 10 | Control word |
| 2 | 60 7A 00 20 | Target Position (Mode PP) |

RXPDO 4:

| Subindex | Value | Meaning |
|----------|-------------|---------------------------|
| 0 | 2 | Two PDO-mapping entries |
| 1 | 60 40 00 10 | Control word |
| 2 | 60 FF 00 20 | Target Velocity (Mode PV) |

22.6.3.2 Transmit PDOs (TXPDO)

Four Transmit PDOs can be configured in the drive:

- - configuration of the communication (Objects 1800-1803h)
- - configuration of the PDO-contents (mapping, Objects 1A00-1A03h)

Objects 1800-1803h: 1st - 4th TXPDO communication parameter (DS301)

| | |
|----------------------|--|
| Index | 1800h to 1803h for TXPDO 1 to 4 |
| Name | transmit PDO parameter |
| Object code | RECORD |
| Data type | PDO CommPar |
| Category | mandatory |
| Subindex | 0 |
| Name | number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO Mapping | not possible |
| Value Range | 5 |
| Default Value | 5 |
| Subindex | 1 |
| Name | COB-ID used by PDO |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | UNSIGNED32 |
| Default Value | Index 1800h: 180h + Node-ID Index 1801h: 280h + Node-ID Index 1802h: 380h + Node-ID Index 1803h: 480h + Node-ID |
| Subindex | 2 |
| Name | transmission type |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | UNSIGNED8 |
| Default Value | FFh |

| | |
|----------------------|-----------------------------|
| Subindex | 3 |
| Name | inhibit time |
| Category | optional |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | UNSIGNED16 (n*1/10ms) |
| Default Value | 0h |
| Subindex | 4 |
| Name | reserved |
| Category | optional |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | 0 |
| Default Value | 0 |
| Subindex | 5 |
| Name | event timer |
| Category | optional |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | UNSIGNED16 (0=not used, ms) |
| Default Value | 0h |

Subindex 1 contains the COB-Id of the PDO as a bit coded information:

| Bit-Number | Value | Meaning |
|------------|-------|--|
| 31 | 0 | PDO exists/is valid |
| | 1 | PDO does not exist/is not valid |
| 30 | 0 | RTR allowed on this PDO, not supported |
| | 1 | RTR not allowed on this PDO, not supported |
| 29 | 0 | 11 bit-ID (CAN 2.0A) |
| | 1 | 29 bit-ID (CAN 2.0B), not supported |
| 28 to 11 | X | Identifier-bits with 29 bit-ID, not relevant |
| 10 to 0 | X | Bits 10-0 of COB-ID |

Subindex 2 contains the transmission type of the PDO. There are two ways of setting:

- A value of FFh or 255d for an event-triggered PDO, which is sent immediately after a change in the mapped application objects. Setting of Subindex 3 or 5 has an influence on the sending of a PDO. With Subindex 3 you can configure, in which minimal time the so configured Transmit-PDOs are sent, if PDO-data contents change (reduction of bus-load). With Subindex 5 (event time) a timer is used, which is reset with every event-triggered sending of this PDO. If there is no change of the PDO-content in this time, the PDO is sent caused by this timer event.
- Values from 0 to 240 cause a SYNC-Telegram controlled sending of the PDO. Values from 1 to 240 define how often the SYNC-telegram leads to a sending of a PDO. The value 0 means, that only the next SYNC-telegram leads to a sending of the so configured PDOs.

Objects 1A00-1A03h: 1st - 4th TXPDO mapping parameter (DS301)

| | |
|----------------------|---|
| Index | 1A00h - 1A03h for TXPDO 1 .. 4 |
| Name | transmit PDO mapping |
| Object Code | RECORD |
| Data Type | PDO Mapping |
| Category | mandatory |
| Subindex | 0 |
| Name | number of mapped application objects in PDO |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | 0: PDO is not active 1 - 8: PDO activated, mappings are taken only byte-wise |
| Default Value | PDO1: 1 PDO2: 2 PDO3: 2 PDO4: 2 |
| Subindex | 1 - 8 |
| Name | PDO - mapping for the n-th application object |
| Category | Conditional, depends on number and size of object be mapped |
| Access | R/W |
| PDO Mapping | not possible |
| Value Range | UNSIGNED32 |
| Default Value | See below |

Default TXPDO definition

TXPDO 1:

| Subindex | Value | Meaning |
|----------|-------------|-----------------------|
| 0 | 1 | One PDO-mapping entry |
| 1 | 60 41 00 10 | Status word |

TXPDO 2:

| Subindex | Value | Meaning |
|----------|-------------|----------------------------|
| 0 | 2 | Two PDO-mapping entries |
| 1 | 60 41 00 10 | Status word |
| 2 | 60 61 00 08 | Modes of Operation display |

TXPDO 3:

| Subindex | Value | Meaning |
|----------|-------------|-------------------------|
| 0 | 2 | Two PDO-mapping entries |
| 1 | 60 41 00 10 | Status word |
| 2 | 60 64 00 20 | Position actual value |

TXPDO 4:

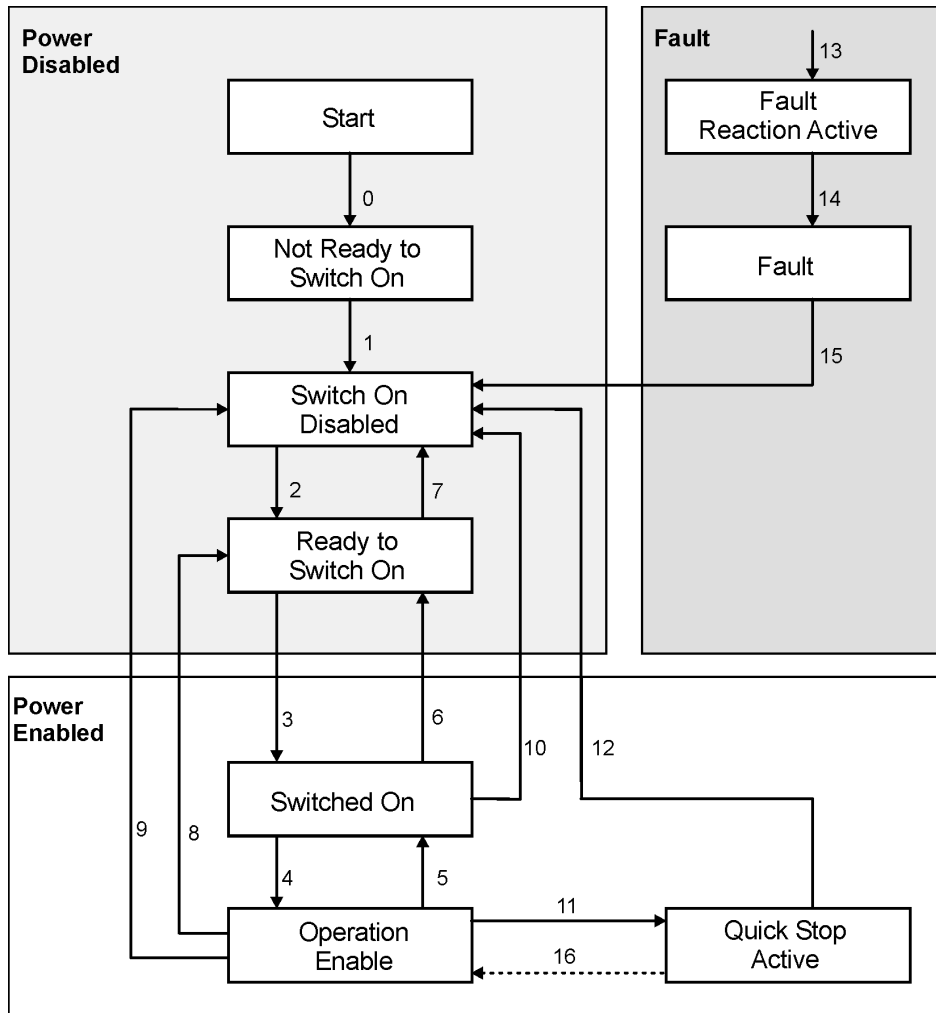
| Subindex | Value | Meaning |
|----------|-------------|-------------------------|
| 0 | 2 | Two PDO-mapping entries |
| 1 | 60 41 00 10 | Status word |
| 2 | 60 6C 00 20 | Velocity actual value |

22.6.4 Device Control (dc)

The device control of the AKD can be used to carry out all the motion functions in the corresponding modes. The control of the AKD is implemented through a mode-dependent status machine. The status machine is controlled through the control word (=> p. 195).

The mode setting is made through the object "Modes of Operation" (=> p. 198). The states of the status machine can be revealed by using the status word (=> p. 196).

22.6.4.1 Status Machine (DS402)



States of the Status Machine

| State | Description |
|-------------------------|--|
| Not Ready for Switch On | AKD is not ready to switch on, there is no operational readiness (BTB/RTO) signaled from the controller program. |
| Switch On Disable | AKD is ready to switch on, parameters can be transferred, the DC-link voltage can be switched on, motion functions cannot be carried out yet. |
| Ready to Switch On | DC-link voltage may be switched on, parameters can be transferred, motion functions cannot be carried out yet. |
| Switched On | DC-link voltage must be switched on, parameters can be transferred, motion functions cannot be carried out yet, output stage is switched on (enabled). |
| Operation Enable | No fault present, output stage is enabled, motion functions are enabled. |
| Quick Stop Active | Drive has been stopped with the emergency ramp, output stage is enabled, motion functions are not enabled. |
| Fault Reaction Active | A fault has occurred and the drive is stopped with the quickstop ramp. |
| Fault | A fault is active, the drive has been stopped and disabled. |

Transitions of the status machine

The state transitions are affected by internal events (e.g. switching off the DC-link voltage) and by the flags in the control word (bits 0,1,2,3,7).

| Transition | Event | Action |
|------------|--|--|
| 0 | Reset | Initialization |
| 1 | Initialization completed successfully. AKD is ready to operate. | none |
| 2 | Bit 1 Disable Voltage and Bit 2 Quick Stop are set in the control word (Shutdown command). DC-link voltage may be present. | none |
| 3 | Bit 0 is also set (Switch On command) | Output stage is switched on (enabled), provided that the hardware enable is present (logical AND). Drive has torque. |
| 4 | Bit 3 is also set (Enable Operation command) | Motion function is enabled, depending on the mode that is set. |
| 5 | Bit 3 is canceled (Disable Operation command) | Motion function is inhibited. Drive is stopped, using the relevant ramp (mode-dependent). The present position is maintained. |
| 6 | Bit 0 is canceled (Shutdown command) | Output stage is disabled. Drive has no torque. |
| 7 | Bits 1 and 2 are canceled (Quick Stop/Disable Voltage command) | none |
| 8 | Bit 0 is canceled (Shutdown command) | Output stage is disabled. Drive has no torque. |
| 9 | Bit 1 is canceled (Disable Voltage command) | Output stage is disabled. Drive has no torque. |
| 10 | Bits 1 and 2 are canceled (Quick Stop/Disable Voltage command) | Output stage is disabled. Drive has no torque. |
| 11 | Bit 2 is canceled (Quick Stop command) | Drive is stopped with the emergency braking ramp. The output stage remains enabled. Setpoints are canceled (motion block number, digital setpoint, speed for jogging or homing). Bit 2 must be set again before any further motion tasks can be performed. |
| 12 | Bit 1 is canceled ('Disable Voltage' command) | Output stage is disabled. Drive has no torque. |
| 13 | Fault reaction active | Execute appropriate fault reaction |
| 14 | Fault reaction is completed | Drive function is disabled. The power section may be switched off. |
| 15 | "Fault Reset" command received from host | A reset of the fault condition is carried out if no fault exists currently on the drive. After leaving the state Fault the Bit7 'Reset Fault' of the controlword must be cleared by the host |
| 16 | Bit 2 is set | Motion function is enabled again. |

NOTE

If the drive is operated through the control word/status word, then no control commands may be sent through another communication channel (ASCII channel, RS232).

22.6.4.2 Object Description

Object 6040h: Control word (DS402)

The control commands are built up from the logical combination of the bits in the control word and external signals (e.g enable output stage). The definitions of the bits are shown below:

| | |
|----------------------|--------------|
| Index | 6040h |
| Name | control word |
| Object code | VAR |
| Data type | UNSIGNED16 |
| Access | R/W |
| PDO mapping | possible |
| Unit | — |
| Value range | 0 to 65535 |
| EEPROM | no |
| Default value | 0 |

Bit assignment in control word

| Bit | Name | Bit | Name |
|-----|---|-----|-----------------------|
| 0 | Switch on | 8 | Pause/halt |
| 1 | Disable Voltage | 9 | reserved |
| 2 | Quick Stop | 10 | reserved |
| 3 | Enable Operation | 11 | reserved |
| 4 | Operation mode specific | 12 | reserved |
| 5 | Operation mode specific | 13 | Manufacturer-specific |
| 6 | Operation mode specific | 14 | Manufacturer-specific |
| 7 | Reset Fault (only effective for faults) | 15 | Manufacturer-specific |

Commands in the control word

| Command | Bit 7 Fault Reset | Bit 3 Enable Operation | Bit 2 Quick Stop | Bit 1 Disable Voltage | Bit 0 Switch on | Transitions |
|-------------------|-------------------------|------------------------------|------------------------|-----------------------------|-----------------------|--------------|
| Shutdown | X | X | 1 | 1 | 0 | 2, 6, 8 |
| Switch on | X | X | 1 | 1 | 1 | 3 |
| Disable Voltage | X | X | X | 0 | X | 7, 9, 10, 12 |
| Quick Stop | X | X | 0 | 1 | X | 7, 10, 11 |
| Disable Operation | X | 0 | 1 | 1 | 1 | 5 |
| Enable Operation | X | 1 | 1 | 1 | 1 | 4, 16 |
| Fault Reset | 1 | X | X | X | X | 15 |

Bits marked by an X are irrelevant.

Mode-dependent bits in the control word

The following table shows the mode-dependent bits in the control word. Only manufacturer-specific modes are supported at present. The individual modes are set by Object 6060_h Modes of operation.

| Operation mode | No. | Bit 4 | Bit 5 | Bit 6 |
|--|-----|------------------------|------------------------|-------------------|
| Profile Position Mode (pp) | 01h | new_setpoint | change_set_immediately | absolute/relative |
| Profile Velocity Mode (pv) | 03h | reserved | reserved | reserved |
| Profile Torque Mode (tq) | 04h | reserved | reserved | reserved |
| Homing Mode (hm) | 06h | homing_operation_start | reserved | reserved |
| Interpolated Position Mode (ip) | 07h | Enable Interpolation | reserved | reserved |
| Cyclic sync position Mode (csp) | 08h | reserved | reserved | reserved |

Description of the remaining bits in the control word

The remaining bits in the control word are described below.

Bit 8 Pause If Bit 8 is set, then the drive halts (pauses) in all modes. The setpoints (speed for homing or jogging, motion task number, setpoints for digital mode) for the individual modes are retained.

Bit 9,10 These bits are reserved for the drive profile (DS402).

Bit 13, 14, 15 These bits are manufacturer-specific, and reserved at present.

Object 6041h: Status word (DS402)

The momentary state of the status machine can be read out with the aid of the status word.

| | |
|----------------------|--------------|
| Index | 6041h |
| Name | Status word |
| Object code | VAR |
| Data type | UNSIGNED16 |
| Access | R/W |
| PDO mapping | possible |
| Unit | — |
| Value range | 0 to 65535 |
| EEPROM | yes |
| Default value | 0 |

Bit assignment in the status word

| Bit | Name | Bit | Name |
|-----|--------------------|-----|------------------------------------|
| 0 | Ready to switch on | 8 | Manufacturer-specific (reserved) |
| 1 | Switched on | 9 | Remote (always 1) |
| 2 | Operation enabled | 10 | Target reached |
| 3 | Fault | 11 | Internal limit active |
| 4 | Voltage enabled | 12 | Operation mode specific (reserved) |
| 5 | Quick stop | 13 | Operation mode specific (reserved) |
| 6 | Switch on disabled | 14 | Manufacturer-specific (reserved) |
| 7 | Warning | 15 | Manufacturer-specific (reserved) |

States of the status machine

| State | Bit 6 switch on disabled | Bit 5 quick stop | Bit 3 fault | Bit 2 operation enabled | Bit 1 switched on | Bit 0 ready to switch on |
|------------------------|--------------------------------|---------------------|----------------|-------------------------------|-------------------------|--------------------------------|
| Not ready to switch on | 0 | X | 0 | 0 | 0 | 0 |
| Switch on disabled | 1 | X | 0 | 0 | 0 | 0 |
| Ready to switch on | 0 | 1 | 0 | 0 | 0 | 1 |
| Switched on | 0 | 1 | 0 | 0 | 1 | 1 |
| Operation enabled | 0 | 1 | 0 | 1 | 1 | 1 |
| Fault | 0 | X | 1 | 0 | 0 | 0 |
| Fault reaction active | 0 | X | 1 | 1 | 1 | 1 |
| Quick stop active | 0 | 0 | 0 | 1 | 1 | 1 |

Bits marked by X are irrelevant

Description of the remaining bits in the status word

Bit 4: voltage_enabled The DC-link voltage is present if this bit is set.

Bit 7: warning There are several possible reasons for Bit 7 being set and this warning being produced. The reason of a warning can be seen by the Error code of the Emergency message, which is sent on the bus caused by this warning.

Bit 9: remote is always set to 1, i.e. the drive can always communicate and be influenced via the RS232 - interface.

Bit 10: target_reached This is set when the drive has reached the target position.

Bit 11: internal_limit_active This bit specifies that a movement was or is limited. In different modes, different warnings cause the bit to be set. The following assignments exist:

| Mode of operation | Warnings which set Bit 11 |
|-------------------|------------------------------|
| all | n04, n06, n07, n10, n11, n14 |
| 0x1 (PP), 0x88 | n03, n08, n09, n20 |

Object 6060h: Modes of Operation (DS402)

This object is used to set the mode, which can be read out by Object 6061h. Two types of operating mode are used:

- manufacturer-specific operating modes
- operating modes as per CANopen drive profile DS402

These operating modes are defined in the CANopen drive profile DS402. After the mode has been changed, the corresponding setpoint must be set once more (for instance, the homing velocity in the mode homing_setpoint). If the position or jogging mode is stored, then the Homing mode is set after a RESET of the drive.

NOTE An operating mode only becomes valid when it can be read by Object 6061h.

WARNING Never change the mode while the motor is running! The drive could move unexpectedly. When the drive is enabled, a mode change is only permissible at zero speed. Set the speed setpoint to 0 before changing over.

| | |
|----------------------|-------------------|
| Index | 6060h |
| Name | mode of operation |
| Object code | VAR |
| Data type | INTEGER8 |
| Category | mandatory |
| Access | R/W |
| PDO mapping | possible |
| Value range | 1, 3, 4, 6, 7, 8 |
| Default value | — |

Supported modes (negative values are manufacturer specific modes):

| Value (hex) | Mode |
|-------------|----------------------------|
| 1 | Profile position mode |
| 3 | Profile velocity mode |
| 4 | Profile torque mode |
| 6 | Homing mode |
| 7 | Interpolated position mode |
| 8 | Cyclic sync position mode |

Object 6061h: Modes of Operation Display (DS402)

This object can be used to read the mode that is set by Object 6060h. An operating mode only becomes valid when it can be read by Object 6061h (see also Object 6060h).

| | |
|----------------------|---------------------------|
| Index | 6061h |
| Name | mode of operation display |
| Object code | VAR |
| Data type | INTEGER8 |
| Category | mandatory |
| Access | R/O |
| PDO mapping | possible |
| Value range | 1, 3, 4, 6, 7, 8 |
| Default value | — |

22.6.5 Factor Groups (fg) (DS402)

The "factor groups" define the units of position-, velocity- and acceleration setpoints. These values are converted into drive-specific parameters.

Two types of scalings can be used depending on the configuration bit 4 in FBUS.PARAM05:

1. Scaling analog to Telnet. Then you should set the UNIT.PROTARY = 3, UNIT.VROTARY = 3 and UNIT.ACCROTARY = 3.
2. Scaling only with DS402 - scaling factors independant of the scalings used via Telnet. Therefore use the settings via the objects 204C / 6092.

NOTE

The drive parameters for the unit definitions should be set as follows:
UNIT.PROTARY= 3 (UNIT.PIN/UNIT.POUT)
UNIT.VROTARY = 3 (UNIT.PIN/UNIT.POUT/s)
UNIT.ACCROTARY = 3 (c UNIT.PIN/UNIT.POUT/s²)

22.6.5.1 General Information

Factors

You can convert between physical dimensions and sizes, and the internal units used in the device (increments). Several factors can be implemented. This chapter describes how these factors influence the system, how they are calculated and which data are necessary to build them.

Relationship between Physical and Internal Units

The factors defined in the factor group set up a relationship between device-internal units (increments) and physical units.

The factors are the result of the calculation of two parameters called dimension index and notation index. The dimension index indicates the physical dimension, the notation index indicates the physical unit and a decimal exponent for the values. These factors are directly used to normalize the physical values.

The notation index can be used in two ways:

- For a unit with decimal scaling and notation index < 64, the notation index defines the exponent/decimal place of the unit.
- For a unit with non-decimal scaling and notation index > 64, the notation index defines the subindex of the physical dimension of the unit.

22.6.5.2 Objects for velocity scaling

Object 204Ch: pv scaling factor

This object shall indicate the configured numerator and denominator of the pv scaling factor. The pv scaling factor serves to modify the resolution or directing range of the specified setpoint. It is also included in calculation of the vl velocity demand, and vl velocity actual value. It does not influence the velocity limit function and the ramp function. The value shall have no physical unit and shall be given in the range from -32 768 to +32 767, but the value of 0 shall not be used.

| | |
|----------------------|-------------------------------|
| Index | 204Ch |
| Name | pv scaling factor |
| Object code | ARRAY |
| Data type | INTEGER32 |
| Category | optional |
| Subindex | 0 |
| Description | number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | 2 |
| Default value | no |
| Subindex | 1 |
| Description | pv scaling factor numerator |
| Category | optinal |
| Access | R/W |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | +1 |
| Subindex | 2 |
| Description | pv scaling factor denominator |
| Category | optional |
| Access | R/W |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | +1 |

22.6.5.3 Objects for position calculation

Object 608Fh: Position encoder resolution (DS402)

The position encoder resolution defines the ratio of encoder increments per motor revolution.

| | |
|----------------------|-----------------------------|
| Index | 608Fh |
| Name | Position encoder resolution |
| Object Code | ARRAY |
| Data Type | UNSIGNED 32 |
| Category | optional |
| Subindex | 0 |
| Name | number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO Mapping | not possible |
| Value Range | 2 |
| Default Value | 2 |
| Subindex | 1 |
| Name | Encoder increments |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | possible |
| Value Range | UNSIGNED 32 |
| Default Value | 2 ²⁰ |
| Subindex | 2 |
| Name | Motor revolutions |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | possible |
| Value Range | UNSIGNED 32 |
| Default Value | 1 |

Object 6092h: Feed constant (DS402)

The feed constant defines the ratio of feed in position units per driving shaft revolutions. This includes the gear if present.

| | |
|----------------------|-------------------|
| Index | 6092h |
| Name | Feed constant |
| Object Code | ARRAY |
| Data Type | UNSIGNED 32 |
| Category | optional |
| Subindex | 0 |
| Name | number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO Mapping | not possible |
| Value Range | 2 |
| Default Value | 2 |
| Subindex | 1 |
| Name | Feed |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | possible |
| Value Range | UNSIGNED 32 |
| Default Value | 1 |
| Subindex | 2 |
| Name | Shaft revolutions |
| Category | mandatory |
| Access | R/W |
| PDO Mapping | possible |
| Value Range | UNSIGNED 32 |
| Default Value | 1 |

22.6.6 Profile Velocity Mode (pv) (DS402)

22.6.6.1 General Information

The profile velocity mode enables the processing of velocity setpoints and the associated accelerations.

Objects that are defined in this section

| Index | Object | Name | Type | Access |
|-------|--------|-----------------------|-----------|--------|
| 606Ch | VAR | velocity actual value | INTEGER32 | R/O |
| 60FFh | VAR | target velocity | INTEGER32 | R/W |

Objects that are defined in other sections

| Index | Object | Name | Type | Section |
|-------|--------|------------------------|------------|----------------|
| 6040h | VAR | control word | INTEGER16 | dc (=> p. 195) |
| 6041h | VAR | status word | UNSIGNED16 | dc (=> p. 196) |
| 6063h | VAR | position actual value* | INTEGER32 | pc (=> p. 207) |
| 6083h | VAR | profile acceleration | UNSIGNED32 | pp (=> p. 220) |
| 6084h | VAR | profile deceleration | UNSIGNED32 | pp (=> p. 220) |

22.6.6.2 Object description

Object 606Ch: velocity actual value (DS402)

The object velocity actual value represents the actual speed.

| | |
|----------------------|---|
| Index | 606Ch |
| Name | velocity actual value, VL.FB |
| Object code | VAR |
| Data type | INTEGER32 |
| Mode | pv |
| Access | R/O |
| PDO mapping | possible |
| Unit | velocity units (SDO is in user units and the PDO is in RPM) |
| Value range | (-2^{31}) to $(2^{31}-1)$ |
| Default value | — |
| EEPROM | no |

Object 60FFh: target velocity (DS402)

The speed setpoint (target velocity) represents the setpoint for the ramp generator.

| | |
|----------------------|-----------------------------|
| Index | 60FFh |
| Name | target velocity, VL.CMDU |
| Object code | VAR |
| Data type | INTEGER32 |
| Mode | pv |
| Access | R/W |
| PDO mapping | possible |
| Unit | increments |
| Value range | (-2^{31}) to $(2^{31}-1)$ |
| Default value | — |
| EEPROM | no |

22.6.7 Profile Torque Mode (tq) (DS402)

22.6.7.1 General Information

The profile torque mode enables the processing of torque setpoints and the associated current.

Objects that are defined in this section

| Index | Object | Name | Type | Access |
|-------|--------|---------------------|------------|--------|
| 6071h | VAR | Target torque | INTEGER16 | R/W |
| 6073h | VAR | Max current | UNSIGNED16 | R/W |
| 6077h | VAR | Torque actual value | INTEGER16 | R/O |

Objects that are defined in other sections

22.6.7.2 Object description

Object 6071h: Target torque (DS402)

This parameter is the input value for the torque controller in profile torque mode and the value is given per thousand (0.001%) of rated torque.

| | |
|----------------------|---|
| Index | 6071h |
| Name | Target torque |
| Object code | VAR |
| Data type | INTEGER16 |
| Category | conditional; mandatory, if tq supported |
| Access | R/W |
| PDO mapping | possible |
| Value range | INTEGER16 |
| Default value | 0 |

Object 6073h: Max current (DS402)

This value represents the maximum permissible torque creating current in the motor and is given per thousand (0.001%) of rated current.

| | |
|----------------------|--------------|
| Index | 6073h |
| Name | Max current |
| Object code | VAR |
| Data type | UNSIGNED16 |
| Category | optional |
| Access | R/W |
| PDO mapping | possible |
| Value range | UNSIGNED16 |
| Default value | 0 |

Object 6077h: Torque actual value (DS402)

The torque actual value corresponds to the instantaneous torque in the drive motor. The value is given per thousand (0.001%) of rated torque.

| | |
|----------------------|---------------------|
| Index | 6077h |
| Name | Torque actual value |
| Object code | VAR |
| Data type | INTEGER16 |
| Category | optional |
| Access | R/O |
| PDO mapping | possible |
| Value range | INTEGER16 |
| Default value | 0 |

22.6.8 Position Control Function (pc) (DS402)

22.6.8.1 General Information

This section describes the actual position values that are associated with the position controller of the drive. They are used for the profile position mode.

Objects that are defined in this section

| Index | Object | Name | Type | Access |
|-------|--------|------------------------|------------|--------|
| 6063h | VAR | position actual value* | INTEGER32 | r |
| 6064h | VAR | position actual value | INTEGER32 | r |
| 6065h | VAR | following error window | UNSIGNED32 | R/W |

Objects that are defined in other sections

| Index | Object | Name | Type | Section |
|-------|--------|-------------------------|------------|----------------|
| 607Ah | VAR | target position | INTEGER32 | pp (=> p. 219) |
| 607Ch | VAR | home-offset | INTEGER32 | hm (=> p. 214) |
| 607Dh | ARRAY | software position limit | INTEGER32 | pp (=> p. 219) |
| 6040h | VAR | control word | INTEGER16 | dc (=> p. 195) |
| 6041h | VAR | status word | UNSIGNED16 | dc (=> p. 196) |

22.6.8.2 Object Description

Object 6063h: position actual value* (DS402)

The object position actual value provides the momentary actual position in increments. The resolution is defined with Object 608F as power-of-two number.

| | |
|----------------------|--|
| Index | 6063h |
| Name | position actual value |
| Object code | VAR |
| Data type | INTEGER32 |
| Mode | pc, pp |
| Access | R/W |
| PDO mapping | possible |
| Unit | increments (1 turn = 2^{PRBASE}) |
| Value range | (-2^{31}) to $(2^{31}-1)$ |
| Default value | 2^{20} |
| EEPROM | no |

Object 6064h: position actual value (DS402)

The object position actual value provides the actual position. The resolution can be altered by the gearing factors of the position controller (Object 6092).

| | |
|----------------------|------------------------------|
| Index | 6064h |
| Name | position actual value, PL.FB |
| Object code | VAR |
| Data type | INTEGER32 |
| Mode | pc, pp |
| Access | R/W |
| PDO mapping | possible |
| Unit | position units |
| Value range | (-2^{31}) to $(2^{31}-1)$ |
| Default value | — |
| EEPROM | no |

Object 6065h: Following error window

The following error window defines a range of tolerated position values symmetrically to the position demand value. A following error might occur when a drive is blocked, unreachable profile velocity occurs, or at wrong closed loop coefficients. If the value of the following error window is 0, the following control is switched off.

| | |
|----------------------|------------------------|
| Index | 6065h |
| Name | Following error window |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Category | optional |
| Access | R/W |
| PDO mapping | no |
| Value range | UNSIGNED32 |
| Default value | 0 |

Object 60F4h: Following error actual value (DS402)

This object returns the current value of the following error in units defined by the user.

| | |
|----------------------|------------------------------|
| Index | 60F4h |
| Name | Following error actual value |
| Object code | VAR |
| Data type | Integer32 |
| Category | optional |
| Access | R/O |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | 0 |

22.6.9 Interpolated Position Mode (ip) (DS402)

22.6.9.1 General information

The interpolated position mode is implemented in a simple, straightforward way. Single position setpoints must be transmitted in the interpolation time period and are taken over on every defined SYNC - telegram sent. A linear interpolation is used between the setpoints.

Objects defined in this section

| Index | Object | Name | Type | Access |
|-------|--------|---|---|--------|
| 60C0h | VAR | Interpolation sub mode select | INTEGER16 | R/W |
| 60C1h | ARRAY | Interpolation data record | INTEGER32 | R/W |
| 60C2h | RECORD | Interpolation time period | Interpolation time period | R/W |
| 60C4h | RECORD | Interpolation data configuration record | Interpolation data configuration record | R/W |

Objects defined in other sections

22.6.9.2 Object description

Object 60C0h: Interpolation sub mode select

In the AKD the linear interpolation between position setpoints is supported. The only allowed value is 0.

| | |
|----------------------|-------------------------------|
| Index | 60C0h |
| Name | Interpolation sub mode select |
| Object code | VAR |
| Data type | INTEGER16 |
| Category | optional |
| Access | R/W |
| PDO mapping | possible |
| Value range | 0 |
| Default value | 0 |

Object 60C1h: Interpolation data record

In the AKD a single setpoint is supported for the interpolated position mode. For the linear interpolation mode each interpolation data record simply can be regarded as a new position setpoint.

After the last item of an interpolation data record is written to the devices input buffer, the pointer of the buffer is automatically incremented to the next buffer position.

| | |
|----------------------|---|
| Index | 60C1h |
| Name | Interpolation data record |
| Object code | ARRAY |
| Data type | INTEGER32 |
| Category | optional |
| Subindex | 0 |
| Description | number of entries |
| Data type | UNSIGNED8 |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | 1 |
| Default value | no |
| Subindex | 1 |
| Description | x1the first parameter of ip function $fip(x1, \dots, xN)$ |
| Category | mandatory |
| Access | R/W |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | no |

Object 60C2h: Interpolation time period

The interpolation time period is used for the PLL (phase locked loop) synchronized position modes. The unit (sub-index 1) of the time is given in $10^{\text{interpolation time index}}$ seconds.

Only multiples of 1 ms are allowed. The two values define the internal ASCII - parameter PTBASE (given in multiples of 250 Mikroseconds). Both values must be written to fix a new interpolation time period. PTBASE will only be updated then.

| | |
|----------------------|--|
| Index | 60C2h |
| Name | Interpolation time period |
| Object code | RECORD |
| Data type | Interpolation time period record (0080h) |
| Category | optional |
| Subindex | 0 |
| Description | number of entries, FBUS.SAMPLEPERIOD |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | 2 |
| Default value | 2 |
| Subindex | 1 |
| Description | Interpolation time units |
| Category | mandatory |
| Access | R/W |
| PDO mapping | possible |
| Value range | UNSIGNED8 |
| Default value | 1 |

Object 60C4h: Interpolation data configuration

Only a single position setpoint is supported in the AKD. Therefore only the value 1 in Subindex 5 is possible. All other subindices are set to 0.

| | |
|----------------------|---|
| Index | 60C4h |
| Name | Interpolation data configuration |
| Object code | RECORD |
| Data type | Interpolation data configuration record (0081h) |
| Category | optional |
| Subindex | 0 |
| Description | number of entries |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | 6 |
| Default value | 6 |
| Subindex | 1 |
| Description | Maximum buffer size |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | UNSIGNED32 |
| Default value | 0 |
| Subindex | 2 |
| Description | Actual buffer size |
| Category | mandatory |
| Access | R/W |
| PDO mapping | possible |
| Value range | UNSIGNED32 |
| Default value | 0 |
| Subindex | 3 |
| Description | Buffer organization |
| Category | mandatory |
| Access | R/W |
| PDO mapping | possible |
| Value range | UNSIGNED8 |
| Default value | 0 |

| | |
|----------------------|---------------------|
| Subindex | 4 |
| Description | Buffer position |
| Category | mandatory |
| Access | R/W |
| PDO mapping | possible |
| Value range | UNSIGNED16 |
| Default value | 0 |
| Subindex | 5 |
| Description | Size of data record |
| Category | mandatory |
| Access | W |
| PDO mapping | possible |
| Value range | 1 to 254 |
| Default value | 1 |
| Subindex | 6 |
| Description | Buffer clear |
| Category | mandatory |
| Access | W |
| PDO mapping | possible |
| Value range | UNSIGNED8 |
| Default value | 0 |

22.6.10 Homing Mode (hm) (DS402)

22.6.10.1 General information

This section describes the various parameters which are required to define a homing mode.

Objects that are defined in this section

| Index | Object | Name | Type | Access |
|-------|--------|---|------------|--------|
| 607Ch | VAR | HOME.P: home offset | INTEGER32 | R/W |
| 6098h | VAR | HOME.MODE, HOME.DIR:homing method | INTEGER8 | R/W |
| 6099h | ARRAY | HOME.V: homing speeds | UNSIGNED32 | R/W |
| 609Ah | VAR | HOME.ACC, HOME.DEC: homing acceleration/deceleration | UNSIGNED32 | R/W |

Objects that are defined in other sections

| Index | Object | Name | Type | Section |
|-------|--------|--------------|------------|----------------|
| 6040h | VAR | control word | INTEGER16 | dc (=> p. 195) |
| 6041h | VAR | status word | UNSIGNED16 | dc (=> p. 196) |

22.6.10.2 Object Description

Object 607Ch: homing offset (DS402)

The reference offset (home offset) is the difference between the zero position for the application and the zero point of the machine. All subsequent absolute motion tasks take account of the reference offset.

| | |
|----------------------|-----------------------------|
| Index | 607Ch |
| Name | home offset, HOME.P |
| Object code | VAR |
| Data type | INTEGER32 |
| Mode | hm |
| Access | R/W |
| PDO mapping | possible |
| Unit | user-defined |
| Value range | (-2^{31}) to $(2^{31}-1)$ |
| Default value | 0 |

Object 6098h: homing method (DS402)

| | |
|----------------------|------------------------------------|
| Index | 6098h |
| Name | homing method, HOME.MODE, HOME.DIR |
| Object code | VAR |
| Data type | INTEGER8 |
| Mode | hm |
| Access | R/W |
| PDO mapping | possible |
| Unit | position units |
| Value range | -128 to 127 |
| Default value | 0 |

Description of the homing methods

Choosing a homing method by writing a value to homing method (Object 6098h) will clearly establish:

- the homing signal (P-Stop, N-Stop, reference switch)
- the direction of actuation

and where appropriate

- the position of the index pulse.

The reference position is give by the reference offset (Object 607Ch).

A detailed description of the types of homing movement can be found in the description of WorkBench.

The following homing methods are supported:

| Method as per DS402 | Brief description: Homing | command |
|---------------------|--|----------------------------|
| -128 to -1 | reserved | — |
| 0 | reserved | — |
| 1 | homing to negative limit switch, with zeroing, negative count direction | HOME.MODE=2, HOME.DIR=0 |
| 2 | homing to positive limit switch, with zeroing, positive count direction | HOME.MODE=2, HOME.DIR=1 |
| 3 to 7 | not supported | — |
| 8 | homing to reference switch, with zeroing, positive count direction | HOME.MODE=5, HOME.DIR=1 |
| 9 to 11 | not supported | — |
| 12 | homing to reference switch, with zeroing, negative count direction | HOME.MODE=5, HOME.DIR=0 |
| 13 to 14 | not supported | — |
| 15 to 16 | reserved | — |
| 17 | homing to negative limit switch, without zeroing, negative count direction | HOME.MODE=1, HOME.DIR=0 |
| 18 | homing to negative limit switch, without zeroing, positive count direction | HOME.MODE=1, HOME.DIR=1 |
| 19 to 23 | not supported | — |
| 24 | homing to reference switch, without zeroing, positive count direction | HOME.MODE=4, HOME.DIR=1 |
| 25 to 27 | not supported | — |
| 28 | homing to reference switch, without zeroing, negative count direction | HOME.MODE=4, HOME.DIR=0 |
| 29 to 30 | not supported | — |
| 31 to 32 | reserved | — |
| 33 | homing within a single turn, negative count direction | HOME.MODE=7, HOME.DIR=0 |
| 34 | homing within a single turn, positive count direction | HOME.MODE=7, HOME.DIR=1 |
| 35 | set reference point at present position | HOME.MODE=0, HOME.DIR=0 |
| 36 to 127 | reserved | — |

Object 6099h: homing speeds (DS402)

| | |
|----------------------|---|
| Index | 6099h |
| Name | homing speeds |
| Object code | ARRAY |
| Data type | UNSIGNED32 |
| Subindex | 1 |
| Description | speed during search for switch, HOME.V |
| Mode | hm |
| Access | R/W |
| PDO mapping | possible |
| Unit | velocity units |
| Value range | 0 to $(2^{32}-1)$ |
| Default value | equivalent 60 rpm |
| Subindex | 2 |
| Description | speed during search for zero, HOME.FEEDRATE |
| Mode | hm |
| Access | R/W |
| PDO mapping | not possible |
| Unit | velocity units |
| Value range | 0 to $(2^{32}-1)$ |
| Default value | 1/8 * Object 6099 sub 1 |

Object 609Ah: homing acceleration (DS402)

| | |
|----------------------|---------------------|
| Index | 609Ah |
| Name | homing acceleration |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Mode | hm |
| Access | R/W |
| PDO mapping | possible |
| Unit | acceleration units |
| Value range | 0 to $(2^{32}-1)$ |
| Default value | 0 |

Homing Mode Sequence

The homing movement is started by setting Bit 4 (positive edge). The successful conclusion is indicated by Bit 12 in the status word (" Object 6041h: Status word (DS402)" (=> p. 196)). Bit 13 indicates that an error occurred during the homing movement. In this case, the error code must be evaluated (error register: " Object 1001h: Error register (DS301)" (=> p. 164), " Object 1003h: Predefined Error Field (DS301)" (=> p. 166), manufacturer status: " Object 1002h: Manufacturer Status Register (DS301)" (=> p. 165)).

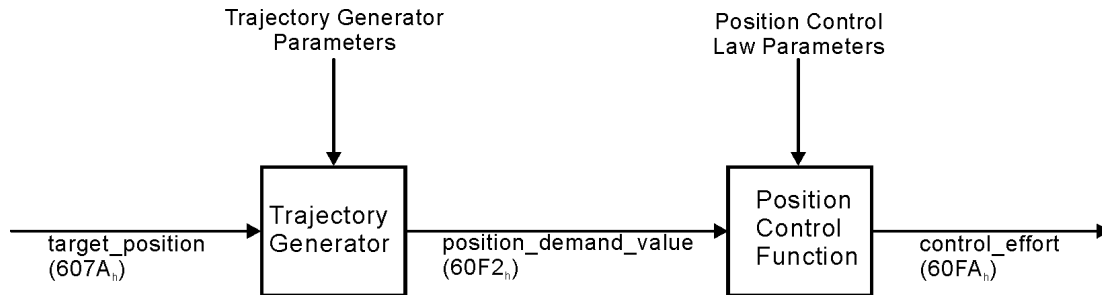
| Bit 4 | Meaning |
|--------|---------------------------------|
| 0 | homing inactive |
| 0 => 1 | start homing movement |
| 1 | homing active |
| 1 => 0 | interruption of homing movement |

| Bit 13 | Bit 12 | Meaning |
|--------|--------|---|
| 0 | 0 | reference point not set, or homing movement not yet finished |
| 0 | 1 | reference point set, homing movement finished |
| 1 | 0 | homing movement could not be successfully concluded (lag error) |
| 1 | 1 | impermissible state |

22.6.11 Profile Position Mode (pp)

22.6.11.1 General Information

The overall structure for this mode is shown in this figure:



The special handshake procedure for the control word and status word is described in "Functional Description" (=> p. 221)

Objects that are defined in this section

| Index | Object | Name | Type | Access |
|-------|--------|-------------------------|------------|--------|
| 607Ah | VAR | target position | INTEGER32 | R/W |
| 607Dh | ARRAY | software position limit | INTEGER32 | R/W |
| 6081h | VAR | profile velocity | UNSIGNED32 | R/W |
| 6083h | VAR | profile acceleration | UNSIGNED32 | R/W |
| 6084h | VAR | profile deceleration | UNSIGNED32 | R/W |

Objects that are defined in other sections

| Index | Object | Name | Type | Section |
|-------|--------|--------------|------------|----------------|
| 6040h | VAR | control word | INTEGER16 | dc (=> p. 195) |
| 6041h | VAR | status word | UNSIGNED16 | dc (=> p. 196) |

22.6.11.2 Object Description

Object 607Ah: target position (DS402)

The object target position defines the target position for the drive. The target position is interpreted as a relative distance or an absolute position, depending on Bit 6 of the control word. The type of relative movement can be further defined by the manufacturer-specific parameter 35B9h Subindex 0 and 1.

The mechanical resolution is set by the gearing factors Object 6093h Subindex 1 and 2.

| | |
|----------------------|-------------------------------|
| Index | 607Ah |
| Name | target position, MT.P |
| Object code | VAR |
| Data type | INTEGER32 |
| Mode | pp |
| Access | R/W |
| PDO mapping | possible |
| Unit | user-defined |
| Value range | $-(2^{31}-1)$ to $(2^{31}-1)$ |
| Default value | — |

Object 607Dh: Software position limit (DS402)

Software position limit contains the sub-parameters min position limit and max position limit. New target positions are checked against these limits. The limits are relative to the machine home position, which is the result of homing (including the home offset (Object 607Ch)). As default the software position limits are switched off. Changed values must be saved and the drive must be restarted to take enable the new the software limits.

| | |
|----------------------|--------------------------------------|
| Index | 607Dh |
| Name | Software position limit, SWLS.LIMIT0 |
| Object code | ARRAY |
| Data type | INTEGER32 |
| Category | optional |
| Subindex | 0 |
| Description | number of entries |
| Category | mandatory |
| Access | R/O |
| PDO mapping | not possible |
| Value range | 2 |
| Default value | 2 |
| Subindex | 1 |
| Description | min position limit, SWLS.LIMIT0 |
| Category | mandatory |
| Access | R/W |
| PDO mapping | possible |
| Value range | INTEGER32 |
| Default value | 0 (switched off) |

Object 6081h: profile velocity (DS402)

The profile velocity is the final velocity that should be reached after the acceleration phase of a motion task.

| | |
|----------------------|------------------------|
| Index | 6081h |
| Name | profile velocity, MT.V |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Mode | pp |
| Access | R/W |
| PDO mapping | possible |
| Unit | speed units |
| Value range | 0 to $(2^{32}-1)$ |
| Default value | 10 |

Object 6083h: profile acceleration (DS402)

The acceleration ramp (profile acceleration) is given in units that are defined by the user (position units per s²). They can be transformed with the acceleration factor defined by Object 6097 sub1 & 2. The type of acceleration ramp can be selected as a linear ramp or a sin² ramp (see Object 6086h).

| | |
|----------------------|------------------------------|
| Index | 6083h |
| Name | profile acceleration, MT.ACC |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Mode | pp |
| Access | R/W |
| PDO mapping | possible |
| Unit | acceleration units |
| Value range | 0 to $(2^{32}-1)$ |
| Default value | 0 |

Object 6084h: profile deceleration (DS402)

The braking/deceleration ramp is handled in the same way as the acceleration ramp (" Object 6083h: profile acceleration (DS402)" => p. 220).

| | |
|----------------------|------------------------------|
| Index | 6084h |
| Name | profile deceleration, MT.DEC |
| Object code | VAR |
| Data type | UNSIGNED32 |
| Mode | pp |
| Access | R/W |
| PDO mapping | possible |
| Unit | deceleration units |
| Value range | 0 to $(2^{32}-1)$ |
| Default value | 0 |

Functional Description

Two different ways to apply target positions to a drive are supported by this device profile.

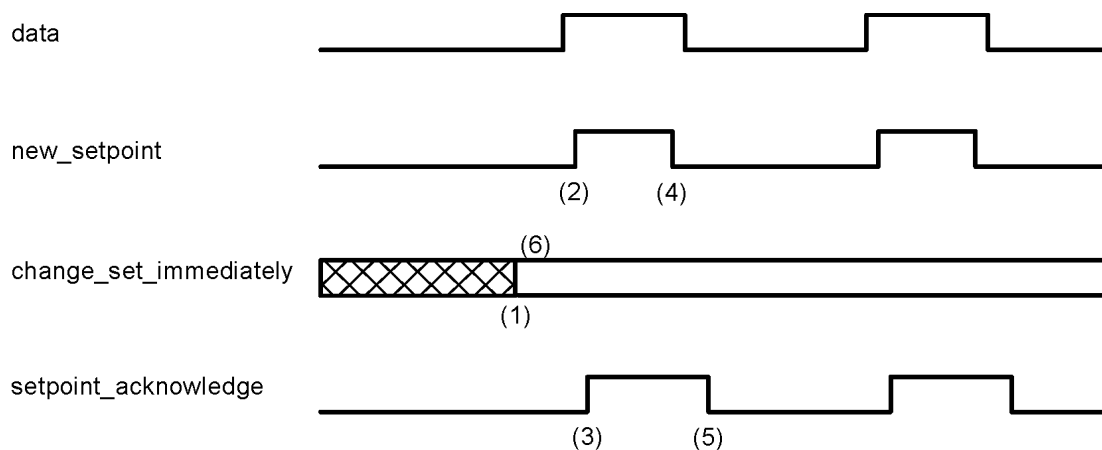
Set of setpoints:

After reaching the target_position, the drive device immediately processes the next target position, which results in a move where the velocity of the drive normally is not reduced to zero after achieving a setpoint. With AKD, this is only possible if trapezoidal ramps are used.

Single setpoints:

After reaching the target_position, the drive device signals this status to a host computer and then receives a new setpoint. After reaching a target_position, the velocity is normally reduced to zero before starting a move to the next setpoint.

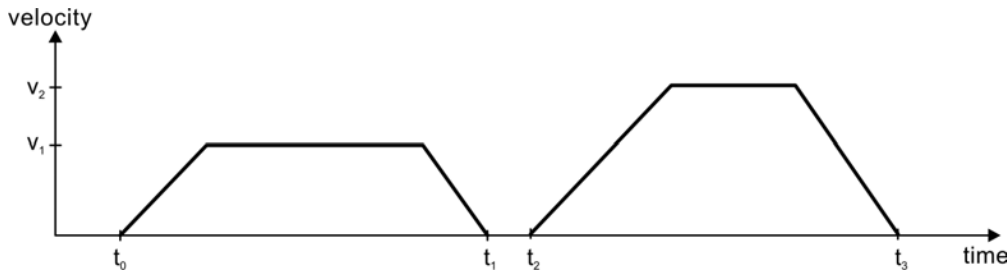
The two modes are controlled by the timing of the bits for new_setpoint and change_set_immediately in the control word, and setpoint_acknowledge in the status word. These bits allow the setting up of a request-response mechanism in order to prepare a set of setpoints while another set is still being processed in the drive unit. This minimizes reaction times within a control program on a host computer.



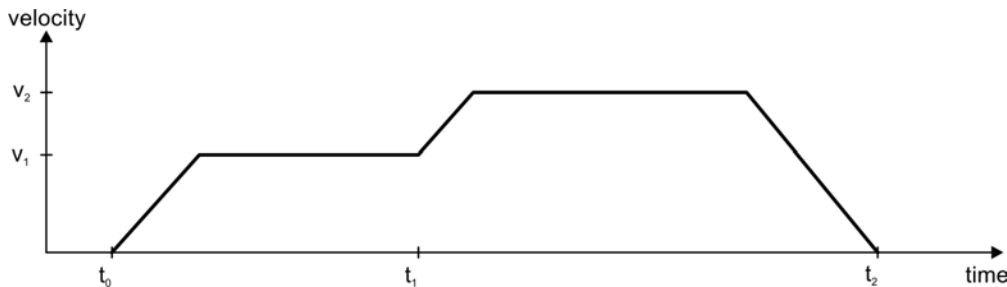
The figures show the difference between the set_of_setpoints mode and the single setpoint mode. The initial status of the bit change_set_immediately in the control word determines which mode is used. To keep these examples simple, only trapezoidal moves are used.

If the bit change_set_immediately is "0" a single setpoint is expected by the drive (1). After data is applied to the drive, a host signals that the data is valid by changing the bit new_setpoint to "1" in the control word (2). The drive responds with setpoint_acknowledge set to "1" in the status word (3) after it has recognized and buffered the new valid data. Now the host can release new_setpoint (4) and subsequently the drive will signal through setpoint_acknowledge = "0" its ability to accept new data again (5).

In the figure below this mechanism results in a velocity of zero after ramping down to reach a target_position X1 at t1. After signaling to the host, that the setpoint has been reached as described above, the next target_position is processed at t2 and reached at t3.



With change_set_immediately set to "1" (6), the host instructs the drive to apply a new setpoint immediately after reaching the previous one. The relative timing of the other signals is unchanged. This behavior causes the drive to process the next setpoint X2 in advance, and to hold its velocity when it reaches the target_position X1 at t1. The drive then moves immediately to the next target_position X2 that has already been calculated.



| Bits in the control word: | | Bits in the status word: | |
|---------------------------|-------------------------------|--------------------------|----------------------|
| Bit 4 | new_setpoint (positive edge!) | Bit 12 | setpoint acknowledge |
| Bit 5 | change_set_immediately | Bit 13 | lag/following error |
| Bit 6 | absolute/relative | | |

Notes on motion task type relative:

If Bit 6 is set, then the motion task type is relative, and activated according to the last target position or actual position. If other types of relative motion are required, these must be activated in advance through the manufacture specific object 35B9h Subindex 0 (MT.CNTL).

Notes on profile position mode:

Functional description for the profile position mode

The drive profile DS402 distinguishes between two methods of moving to a target position. These two methods are controlled by the bits for new_setpoint and change_set_immediately in the control word, and setpoint_acknowledge in the status word. These bits can be used to prepare a motion task while another is still being carried out (handshake).

Moving to several target positions without an intermediate halt

After the target position has been reached, the drive moves immediately to the next target position. This requires that new setpoints are signaled to the drive. This is done through a positive transition of the new_setpoint bit. In this case, the setpoint_acknowledge bit must not be active (=1) in the status word (see also Handshake DS402). The velocity is not reduced to zero when the first setpoint is reached.

Moving to a single target position

The drive moves to the target position, whereby the velocity is reduced to zero. Reaching the target position is signaled by the bit for target_reached in the status word.

22.7 Appendix

22.7.1 Object Dictionary

The following table describes all objects reachable via SDO or PDO. (i.p. = in preparation).

Abbreviations:

| | | | |
|--------|------------------|-------|------------------|
| U | = UNSIGNED | RO | = Read only |
| INT | = INTEGER | RW | = Read and Write |
| VisStr | = Visible String | WO | = Write only |
| | | const | = Constant |

22.7.1.1 Float Scaling

The scaling applied to objects which match floating-point parameters in WorkBench/Telnet are listed in the column "Float Scaling."

For example, index 607Ah is listed as 1:1 - this means that commanding a value of 1000 in SDO 607Ah is equivalent to entering MT.P 1000.000 in Workbench. On the other hand, index 3598h is listed as 1000:1 - this means that commanding a value of 1000 in SDO 3598h is equivalent to entering IL.KP 1.000 in Workbench.

A few parameters are listed as variable (var), because the scaling depends on other settings.

22.7.1.2 Communication SDOs

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|--|--------------|
| 1000h | 0 | U32 | | RO | no | Device type | — |
| 1001h | 0 | U8 | | RO | no | Error register | — |
| 1002h | 0 | U32 | | RO | yes | Manufacturer-specific status register | — |
| 1003h | | ARRAY | | | | Pre-defined error field | — |
| 1003h | 0 | U8 | | RW | no | Number of errors | — |
| 1003h | 1 to 10 | U32 | | RO | no | standard error field | — |
| 1005h | 0 | U32 | | RW | no | COB—ID SYNC message | — |
| 1006h | 0 | U32 | | RW | no | Communication cycle period | — |
| 1008h | 0 | VisStr | | const | no | Manufacturer device name | — |
| 1009h | 0 | VisStr | | const | no | Manufacturer hardware version | — |
| 100Ah | 0 | VisStr | | const | no | Manufacturer software version | — |
| 100Ch | 0 | U16 | | RW | no | Guard time | — |
| 100Dh | 0 | U8 | | RW | no | Lifetime factor | — |
| 1010h | | ARRAY | | | | Number of entries | — |
| 1010h | 0 | U8 | | RO | no | Number of entries | — |
| 1010h | 1 | U32 | | RW | no | Saves the drive parameters from the RAM to the NV. | DRV.NVSAVE |
| 1014h | 0 | U32 | | RW | no | COB—ID for the Emergency Object | — |
| 1016h | | RECORD | | | | Consumer heartbeat time | — |
| 1016h | 0 | U8 | | RO | no | Number of entries | — |
| 1016h | 1 | U32 | | RW | no | Consumer heartbeat time | — |
| 1017h | 0 | U16 | | RW | no | Producer heartbeat time | — |
| 1018h | | RECORD | | | | Identity Object | — |
| 1018h | 0 | U8 | | RO | no | Number of entries | — |
| 1018h | 1 | U32 | | RO | no | Vendor ID | — |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|-------------------------------------|--------------|
| 1018h | 2 | U32 | | RO | no | Product Code | — |
| 1018h | 3 | U32 | | RO | no | Revision number | — |
| 1018h | 4 | U32 | | RO | no | Serial number | |
| 1026h | | ARRAY | | | | OS prompt | — |
| 1026h | 0 | U8 | | RO | no | Number of entries | — |
| 1026h | 1 | U8 | | WO | no | StdIn | — |
| 1026h | 2 | U8 | | RO | no | StdOut | — |
| 1400h | | RECORD | | | | RXPDO1 communication parameter | — |
| 1400h | 0 | U8 | | RO | no | Number of entries | — |
| 1400h | 1 | U32 | | RW | no | RXPDO1 COB—ID | — |
| 1400h | 2 | U8 | | RW | no | Transmission type RXPDO1 | — |
| 1401h | | RECORD | | | | RXPDO2 communication parameter | — |
| 1401h | 0 | U8 | | RO | no | Number of entries | — |
| 1401h | 1 | U32 | | RW | no | RXPDO2 COB—ID | — |
| 1401h | 2 | U8 | | RW | no | Transmission type RXPDO2 | — |
| 1402h | | RECORD | | | | RXPDO3 communication parameter | — |
| 1402h | 0 | U8 | | RO | no | Number of entries | — |
| 1402h | 1 | U32 | | RW | no | RXPDO3 COB—ID | — |
| 1402h | 2 | U8 | | RW | no | Transmission type RXPDO3 | — |
| 1403h | | RECORD | | | | RXPDO4 communication parameter | — |
| 1403h | 0 | U8 | | RO | no | Number of entries | — |
| 1403h | 1 | U32 | | RW | no | RXPDO4 COB—ID | — |
| 1403h | 2 | U8 | | RW | no | Transmission type RXPDO4 | — |
| 1600h | | RECORD | | | | RXPDO1 mapping parameter | — |
| 1600h | 0 | U8 | | RO | no | Number of entries | — |
| 1600h | 1 to 8 | U32 | | RW | no | Mapping for n—th application object | — |
| 1601h | | RECORD | | | | RXPDO2 mapping parameter | — |
| 1601h | 0 | U8 | | RO | no | Number of entries | — |
| 1601h | 1 to 8 | U32 | | RW | no | Mapping for n—th application object | — |
| 1602h | | RECORD | | | | RXPDO3 mapping parameter | — |
| 1602h | 0 | U8 | | RO | no | Number of entries | — |
| 1602h | 1 to 8 | U32 | | RW | no | Mapping for n—th application object | — |
| 1603h | | RECORD | | | | RXPDO4 mapping parameter | — |
| 1603h | 0 | U8 | | RO | no | Number of entries | — |
| 1603h | 1 to 8 | U32 | | RW | no | Mapping for n—th application object | — |
| 1800h | | RECORD | | | | TXPDO1 communication parameter | — |
| 1800h | 0 | U8 | | RO | no | Number of entries | — |
| 1800h | 1 | U32 | | RW | no | TXPDO1 COB—ID | — |
| 1800h | 2 | U8 | | RW | no | Transmission type TXPDO1 | — |
| 1800h | 3 | U16 | | RW | no | Inhibit time | — |
| 1800h | 4 | U8 | | const | no | reserved | — |
| 1800h | 5 | U16 | | RW | no | Event timer | — |
| 1801h | | RECORD | | | | TXPDO2 communication parameter | — |
| 1801h | 0 | U8 | | RO | no | Number of entries | — |
| 1801h | 1 | U32 | | RW | no | TXPDO2 COB—ID | — |
| 1801h | 2 | U8 | | RW | no | Transmission type TXPDO2 | — |
| 1801h | 3 | U16 | | RW | no | Inhibit time | — |
| 1801h | 4 | U8 | | const | no | reserved | — |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|--------------------------------------|--------------|
| 1801h | 5 | U16 | | RW | no | Event timer | — |
| 1802h | | RECORD | | | | TXPDO3 communication parameter | — |
| 1802h | 0 | U8 | | RO | no | Number of entries | — |
| 1802h | 1 | U32 | | RW | no | TXPDO3 COB—ID | — |
| 1802h | 2 | U8 | | RW | no | Transmission type TXPDO3 | — |
| 1802h | 3 | U16 | | RW | no | Inhibit time | — |
| 1802h | 4 | U8 | | const | no | reserved | — |
| 1802h | 5 | U16 | | RW | no | Event timer | — |
| 1803h | | RECORD | | | | TXPDO4 communication parameter | — |
| 1803h | 0 | U8 | | RO | no | Number of entries | — |
| 1803h | 1 | U32 | | RW | no | TXPDO4 COB—ID | — |
| 1803h | 2 | U8 | | RW | no | Transmission type TXPDO4 | — |
| 1803h | 3 | U16 | | RW | no | Inhibit time | — |
| 1803h | 4 | U8 | | const | no | reserved | — |
| 1803h | 5 | U16 | | RW | no | Event timer | — |
| 1A00h | | RECORD | | | | Mapping parameter TXPDO1 | — |
| 1A00h | 0 | U8 | | RO | no | Number of entries | — |
| 1A00h | 1 to 8 | U32 | | RW | no | Mapping for n—th application object | — |
| 1A01h | | RECORD | | | | Mapping parameter TXPDO2 | — |
| 1A01h | 0 | U8 | | RO | no | Number of entries | — |
| 1A01h | 1 to 8 | U32 | | RW | no | Mapping for n—th application object | — |
| 1A02h | | RECORD | | | | Mapping parameter TXPDO3 | — |
| 1A02h | 0 | U8 | | RO | no | Number of entries | — |
| 1A02h | 1 to 8 | U32 | | RW | no | Mapping for n—th application object | — |
| 1A03h | | RECORD | | | | Mapping parameter TXPDO4 | — |
| 1A03h | 0 | U8 | | RO | no | Number of entries | — |
| 1A03h | 1 to 8 | U32 | | RW | no | Mapping for n—the application object | — |

22.7.1.3 Manufacturer specific SDOs

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|---|-------------------|
| 2014h | | ARRAY | | | | Mask TxPDO Channel 1 | — |
| 2014h | 1 | U32 | | RW | no | Mask (Byte 0..3) | — |
| 2014h | 2 | U32 | | RW | no | Mask (Byte 4..7) | — |
| 2015h | | ARRAY | | | | Mask TxPDO Channel 2 | — |
| 2015h | 1 | U32 | | RW | no | Mask (Byte 0..3) | — |
| 2015h | 2 | U32 | | RW | no | Mask (Byte 4..7) | — |
| 2016h | | ARRAY | | | | Mask TxPDO Channel 3 | — |
| 2016h | 1 | U32 | | RW | no | Mask (Byte 0..3) | — |
| 2016h | 2 | U32 | | RW | no | Mask (Byte 4..7) | — |
| 2017h | | ARRAY | | | | Mask TxPDO Channel 4 | — |
| 2017h | 1 | U32 | | RW | no | Mask (Byte 0..3) | — |
| 2017h | 2 | U32 | | RW | no | Mask (Byte 4..7) | — |
| 2018h | | ARRAY | | | | Firmware version | — |
| 2018h | 0 | U16 | | const | no | Number of entries | — |
| 2018h | 1 | U16 | | const | no | Major version | — |
| 2018h | 2 | U16 | | const | no | Minor version | — |
| 2018h | 3 | U16 | | const | no | Revision | — |
| 2018h | 4 | U16 | | const | no | Branch version | — |
| 204Ch | | ARRAY | | | | pv scaling factor | — |
| 204Ch | 0 | U8 | | RO | no | Number of entries | — |
| 204Ch | 1 | INT32 | | RW | yes | pv scaling factor numerator | — |
| 204Ch | 2 | INT32 | | RW | yes | pv scaling factor denominator | — |
| 2050h | 0 | INT32 | 1:1 | RO | no | Position, secondary feedback | DRV.HANDWHEEL |
| 20A0h | 0 | INT32 | var | RO | yes | Latch position 1, positive edge | CAP0.PLFB, CAP0.T |
| 20A1h | 0 | INT32 | var | RO | yes | Latch position 1, negative edge | CAP0.PLFB, CAP0.T |
| 20A2h | 0 | INT32 | var | RO | yes | Latch position 2, positive edge | CAP1.PLFB, CAP1.T |
| 20A3h | 0 | INT32 | var | RO | yes | Latch position 2, negative edge | CAP1.PLFB, CAP1.T |
| 20A4h | 0 | U16 | | RW | yes | Latch control register | — |
| 20A5h | 0 | U16 | | RW | yes | Latch status register | — |
| 20A6h | 0 | INT32 | var | RO | yes | Sets captured position value | CAP0.PLFB |
| 20B8h | 0 | U16 | | RW | no | Clear changed digital input information | — |
| 3405h | | ARRAY | | | | VL.ARTYPE | — |
| 3405h | 0 | U8 | | RO | no | Number of entries | — |
| 3405h | 1 | U8 | | RW | no | Calculation method for BiQuad filter 1 | VL.ARTYPE1 |
| 3405h | 2 | U8 | | RW | no | Calculation method for BiQuad filter 2 | VL.ARTYPE2 |
| 3405h | 3 | U8 | | RW | no | Calculation method for BiQuad filter 3 | VL.ARTYPE3 |
| 3405h | 4 | U8 | | RW | no | Calculation method for BiQuad filter 4 | VL.ARTYPE4 |
| 3406h | | ARRAY | | | | VL BiQuad | — |
| 3406h | 0 | U8 | | RO | no | Number of entries | — |
| 3406h | 1 | U32 | 1000:1 | RW | no | Natural frequency of pole of anti-resonance (AR) filter 1 | VL.ARPF1 |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|--|--------------------------------|
| 3406h | 2 | U32 | 1000:1 | RW | no | Natural frequency of pole of anti-resonance (AR) filter 2 | VL.ARPF2 |
| 3406h | 3 | U32 | 1000:1 | RW | no | Natural frequency of pole of anti-resonance (AR) filter 3 | VL.ARPF3 |
| 3406h | 4 | U32 | 1000:1 | RW | no | Natural frequency of pole of anti-resonance (AR) filter 4 | VL.ARPF4 |
| 3406h | 5 | U32 | 1000:1 | RW | no | Q of pole of anti-resonance (AR) filter 1 | VL.ARPQ1 |
| 3406h | 6 | U32 | 1000:1 | RW | no | Q of pole of anti-resonance (AR) filter 2 | VL.ARPQ2 |
| 3406h | 7 | U32 | 1000:1 | RW | no | Q of pole of anti-resonance (AR) filter 3 | VL.ARPQ3 |
| 3406h | 8 | U32 | 1000:1 | RW | no | Q of pole of anti-resonance (AR) filter 4 | VL.ARPQ4 |
| 3406h | 9 | U32 | 1000:1 | RW | no | Natural frequency of zero of anti-resonance (AR) filter 1 | VL.ARZF1 |
| 3406h | A | U32 | 1000:1 | RW | no | Natural frequency of zero of anti-resonance (AR) filter 2 | VL.ARZF2 |
| 3406h | B | U32 | 1000:1 | RW | no | Natural frequency of zero of anti-resonance (AR) filter 3 | VL.ARZF3 |
| 3406h | C | U32 | 1000:1 | RW | no | Natural frequency of zero of anti-resonance (AR) filter 4 | VL.ARZF4 |
| 3406h | D | U32 | 1000:1 | RW | no | Q of zero of anti-resonance filter 1 | VL.ARZQ1 |
| 3406h | E | U32 | 1000:1 | RW | no | Q of zero of anti-resonance filter 2 | VL.ARZQ2 |
| 3406h | F | U32 | 1000:1 | RW | no | Q of zero of anti-resonance filter 3 | VL.ARZQ3 |
| 3406h | 10 | U32 | 1000:1 | RW | no | Q of zero of anti-resonance filter 4 | VL.ARZQ4 |
| 3407h | | STRUCT | | | | Velocity Filter | — |
| 3407h | 0 | U8 | | RO | no | Number of entries | — |
| 3407h | 1 | INT32 | 1000:1 | RW | no | 10 Hz filtered VL.FB | VL.FBFILTER |
| 3407h | 2 | U32 | 1000:1 | RW | no | Gain for the velocity feedforward | VL.KVFF |
| 3407h | 3 | U32 | | RW | no | Gain for the acceleration feedforward | VL.KBUSFF |
| 3407h | 4 | U32 | 1:1 | RW | no | Sets the velocity error | VL.ERR |
| 3412h | 0 | INT8 | | RW | no | Type of regen resistor | REGEN.TYPE |
| 3414h | 0 | U8 | | RW | | Returns and sets the regen resistor fault level temperature. | REGEN.WATTEXT REGEN.WATTEXT |
| 3415h | 0 | U32 | 1000:1 | RO | no | Thermal regen resistor time constant | REGEN.TEXT |
| 3416h | 0 | U32 | | RO | no | Gets regen resistor's calculated power | REGEN.POWER |
| 3420h | 0 | U16 | 1000:1 | RW | no | Sets the foldback fault level. | IL.FOLDFTHRESH |
| 3421h | 0 | U32 | 1000:1 | RW | no | Sets the user value for the foldback fault level. | IL.FOLDFTHRESHU |
| 3422h | 0 | U32 | 1000:1 | | no | Sets friction compensation value. | IL.FRICITION |
| 3423h | 0 | INT32 | 1000:1 | | no | A constant current command added to compensate for gravity. | IL.OFFSET |
| 3424h | 0 | U16 | | | no | Enables/disables the integrator part of the PI loop. | IL.INTEN (Password Protected) |
| 3425h | 0 | U32 | 1000:1 | RO | no | Reads the overall foldback current limit | IL.IFOLD |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|---|-----------------|
| 3426h | 0 | U32 | 1000:1 | RW | no | Sets current loop acceleration feedforward gain value | IL.KACCCFF |
| 3430h | 0 | U8 | | RW | no | Sets the direction for absolute motion tasks. | PL.MODPDIR |
| 3431h | 0 | U16 | | RW | no | Sets the motion task in the drive | MT.SET |
| 3440h | | ARRAY | | | | Controlled stop parameters | — |
| 3440h | 0 | U8 | | RO | no | Number of entries | — |
| 3440h | 1 | U32 | 1:1 | RW | no | Sets the deceleration value for a controlled stop. | CS.DEC |
| 3440h | 2 | U32 | 1:1 | RW | no | Sets the velocity threshold for a controlled stop. | CS.VTHRESH |
| 3440h | 3 | U32 | | RW | no | Sets the time value for the drive velocity to be within CS.VTHRESH. | CS.TO |
| 3441h | 0 | U8 | | RO | no | Controlled stop state | CS.STATE |
| 3443h | 0 | U16 | | RO | no | Returns the possible reason for a drive disable | DRV.DIS |
| 3444h | 0 | U16 | 1000:1 | RO | no | Maximum current for dynamic braking | DRV.DBILIMIT |
| 3445h | 0 | U32 | | RO | no | Emergency timeout for braking | DRV.DISTO |
| 3450h | 0 | U8 | | WO | no | Release or enable brake | MOTOR.BRAKERLS |
| 3451h | 0 | U8 | | RW | no | Determines which drive parameters are calculated automatically. | MOTOR.AUTASET |
| 3452h | 0 | U16 | | RW | no | Sets the motor maximum voltage | MOTOR.VOLTMAX |
| 3453h | 0 | U32 | | RW | no | Sets the motor temperature warning level | MOTOR.TEMPWARN |
| 3454h | 0 | U32 | 1000:1 | RW | no | Sets the thermal constant of the motor coil | MOTOR.CTF0 |
| 3455h | 0 | U32 | 1000:1 | RW | no | Sets the line-to-line motor Lq | MOTOR.LQLL |
| 3456h | 0 | U32 | 1000:1 | RW | no | Sets the stator winding resistance phase-phase in ohms | MOTOR.R |
| 3457h | | RECORD | | | | Induction Motor parameter | — |
| 3457h | 0 | U8 | | RO | no | Number of entries | — |
| 3457h | 1 | INT32 | 1000:1 | RW | no | Configuration of induction motor's rated velocity. | MOTOR.VRATED |
| 3457h | 2 | U16 | | RW | no | Configuration of induction motor's rated voltage. | MOTOR.VOLTRATED |
| 3457h | 3 | U16 | | RW | no | Sets the minimum voltage for V/f Control. | MOTOR.VOLTMIN |
| 3460h | | RECORD | | | | Capture engines parameters | — |
| 3460h | 0 | U8 | | RO | no | Number of entries | — |
| 3460h | 1 | U8 | | RW | no | Specifies the trigger source for the position capture. | CAP0.TRIGGER |
| 3460h | 2 | U8 | | RW | no | Specifies the trigger source for the position capture. | CAP1.TRIGGER |
| 3460h | 3 | U8 | | RW | no | Selects the captured value. | CAP0.MODE |
| 3460h | 4 | U8 | | RW | no | Selects the captured value. | CAP1.MODE |
| 3460h | 5 | U8 | | RW | no | Controls the precondition logic. | CAP0.EVENT |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|---|----------------------------|
| 3460h | 6 | U8 | | RW | no | Controls the precondition logic. | CAP1.EVENT |
| 3460h | 7 | U8 | | RW | no | Selects the capture precondition edge. | CAP0.PREEDGE |
| 3460h | 8 | U8 | | RW | no | Selects the capture precondition edge. | CAP1.PREEDGE |
| 3460h | 9 | U8 | | RW | no | Sets the precondition trigger. | CAP0.PRESELECT |
| 3460h | A | U8 | | RW | no | Sets the precondition trigger. | CAP1.PRESELECT |
| 3470h | | RECORD | | | | | — |
| 3470h | 0 | U8 | | RO | no | Number of entries | — |
| 3470h | 1 | INT8 | | RW | no | Sets the analog output mode. | AOUT.MODE |
| 3470h | 2 | INT16 | 1000:1 | RW | yes | Reads the analog output value. | AOUT.VALUE |
| 3470h | 3 | INT16 | 1000:1 | RW | yes | Reads and writes the analog output value. | AOUT.VALUEU |
| 3470h | 4 | INT16 | 1000:1 | RO | yes | Reads the value of the analog input signal. | AIN.VALUE |
| 3470h | 5 | U32 | 1000:1 | RW | no | Sets velocity scale factor for analog output | AOUT.VSCALE |
| 3471h | 0 | U32 | 1:1 | RW | no | Sets the analog position scale factor | "AOUT.PSCALE" (=> p. 300) |
| 3472h | 0 | U32 | 1:1 | RW | no | Sets analog pscale factor | "AIN.PSCALE" (=> p. 287) |
| 3480h | 0 | U32 | 1000:1 | RW | no | Integral gain of position regulator PID loop | PL.KI |
| 3481h | | ARRAY | | | | PL.INTMAX | — |
| 3481h | 0 | U8 | | RO | no | Number of entries | — |
| 3481h | 1 | U32 | 1:1 | RW | no | Input saturation | PL.INTINMAX |
| 3481h | 2 | U32 | 1:1 | RW | no | Output saturation | PL.INTOUTMAX |
| 3482h | 0 | INT32 | 1:1 | RO | no | Maximum value of following error in homing | HOME.PERRTHRESH |
| 3483h | 0 | INT32 | 1:1 | RW | no | Sets the position error warning level | PL.ERRWTHRESH |
| 3490h | 0 | INT32 | 1:1 | RO | no | Position feedback offset | FB1.OFFSET |
| 3491h | 0 | U32 | | RO | no | Location of index pulse on EEO | DRV.EMUEMTURN |
| 3492h | 0 | U32 | | RO | no | Motion status of the drive | DRV.MOTIONSTAT |
| 3493h | 0 | U8 | | RO | no | Direction of EEO (emulated encoder output) | DRV.EMUEDIR DRV.EMUEDIR |
| 3494h | | RECORD | | | | WS parameters | — |
| 3494h | 0 | U8 | | RO | no | Number of entries | — |
| 3494h | 1 | INT16 | 1000:1 | RW | no | Sets maximum current used for wake and shake | WS.IMAX |
| 3494h | 2 | INT32 | 1:1 | RW | no | Sets the maximum movement required for wake and shake | WS.DISTMAX |
| 3494h | 3 | U16 | | RW | no | Sets the delay for wake and shake between loops in mode 0 | WS.TDELAY3 |
| 3494h | 4 | INT32 | 1:1 | RW | no | Defines the maximum allowed velocity for Wake & Shake | WS.VTHRESH |
| 3494h | 5 | U8 | | RO | no | Reads wake and shake status | WS.STATE |
| 3494h | 6 | U8 | | RW | no | Arm Wake and Shake to start | WS.ARM |
| 34A0h | | ARRAY | | | | PLS Position | |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|---|--------------|
| 34A0h | 0 | U8 | | RO | no | Number of entries | — |
| 34A0h | 1 | INT32 | 1:1 | RW | no | Limit switch 1 compare value | PLS.P1 |
| 34A0h | 2 | INT32 | 1:1 | RW | no | Limit switch 2 compare value | PLS.P2 |
| 34A0h | 3 | INT32 | 1:1 | RW | no | Limit switch 3 compare value | PLS.P3 |
| 34A0h | 4 | INT32 | 1:1 | RW | no | Limit switch 4 compare value | PLS.P4 |
| 34A0h | 5 | INT32 | 1:1 | RW | no | Limit switch 5 compare value | PLS.P5 |
| 34A0h | 6 | INT32 | 1:1 | RW | no | Limit switch 6 compare value | PLS.P6 |
| 34A0h | 7 | INT32 | 1:1 | RW | no | Limit switch 7 compare value | PLS.P7 |
| 34A0h | 8 | INT32 | 1:1 | RW | no | Limit switch 8 compare value | PLS.P8 |
| 34A1h | | ARRAY | | | | PLS Width | — |
| 34A1h | 0 | U8 | | RO | no | Number of entries | — |
| 34A1h | 1 | INT32 | 1:1 | RW | no | Sets Limit Switch1 Width | PLS.WIDTH1 |
| 34A1h | 2 | INT32 | 1:1 | RW | no | Sets Limit Switch 2 Width | PLS.WIDTH2 |
| 34A1h | 3 | INT32 | 1:1 | RW | no | Sets Limit Switch 3 Width | PLS.WIDTH3 |
| 34A1h | 4 | INT32 | 1:1 | RW | no | Sets Limit Switch 4 Width | PLS.WIDTH4 |
| 34A1h | 5 | INT32 | 1:1 | RW | no | Sets Limit Switch 5 Width | PLS.WIDTH5 |
| 34A1h | 6 | INT32 | 1:1 | RW | no | Sets Limit Switch 6 Width | PLS.WIDTH6 |
| 34A1h | 7 | INT32 | 1:1 | RW | no | Sets Limit Switch 7 Width | PLS.WIDTH7 |
| 34A1h | 8 | INT32 | 1:1 | RW | no | Sets Limit Switch 8 Width | PLS.WIDTH8 |
| 34A2h | | ARRAY | | | | PLS Time | — |
| 34A2h | 0 | U8 | | RO | no | Number of entries | — |
| 34A2h | 1 | U16 | | RW | no | Sets limit switch 1 time | PLS.T1 |
| 34A2h | 2 | U16 | | RW | no | Sets limit switch 2 time | PLS.T2 |
| 34A2h | 3 | U16 | | RW | no | Sets limit switch 3 time | PLS.T3 |
| 34A2h | 4 | U16 | | RW | no | Sets limit switch 4 time | PLS.T4 |
| 34A2h | 5 | U16 | | RW | no | Sets limit switch 5 time | PLS.T5 |
| 34A2h | 6 | U16 | | RW | no | Sets limit switch 6 time | PLS.T6 |
| 34A2h | 7 | U16 | | RW | no | Sets limit switch 7 time | PLS.T7 |
| 34A2h | 8 | U16 | | RW | no | Sets limit switch 8 time | PLS.T8 |
| 34A3h | | ARRAY | | | | PLS Configuration | — |
| 34A3h | 0 | U8 | | RO | no | Number of entries | — |
| 34A3h | 1 | U16 | | RW | no | Enables the limit switches | PLS.EN |
| 34A3h | 2 | U16 | | RW | no | Resets limit switches | PLS.RESET |
| 34A3h | 3 | U16 | | RW | no | Selects limit switch mode | PLS.MODE |
| 34A3h | 4 | U16 | | RW | no | Reads the limit switch state | PLS.STATE |
| 34A4h | 0 | U8 | | RW | no | Sets limit switch units | PLS.UNITS |
| 3501h | 0 | INT32 | 1:1 | RW | no | Acceleration ramp | DRV.ACC |
| 3502h | 0 | INT32 | 1:1 | RW | no | Acceleration ramp for homing/jog modes | HOME.ACC |
| 3506h | 0 | INTEGER | | | no | Action that hardware enable digital input will perform. | DRV.HWENMODE |
| 3509h | 0 | INT32 | 1000:1 | RO | no | Analog input voltage | AIN.VALUE |
| 3522h | 0 | INT32 | 1:1 | RW | no | Deceleration rate | DRV.DEC |
| 3524h | 0 | INT32 | 1:1 | RW | no | Deceleration ramp for homing/jog modes | HOME.DEC |
| 352Ah | 0 | INT32 | | RW | no | Direction of movements | DRV.DIR |
| 3533h | 0 | U32 | | RO | no | Resolution of motor encoder | FB1.ENCRES |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|---|-----------------|
| 3534h | 0 | U32 | | RO | no | Mode of EEO connector | DRV.EMUEMODE |
| 3535h | 0 | U32 | | RO | no | Resolution of EEO | DRV.EMUERES |
| 3537h | 0 | U32 | | RO | no | Location of EEO index pulse | DRV.EMUEZOFFSET |
| 353Bh | 0 | INT32 | | RO | no | Selection of the feedback type | FB1.SELECT |
| 3542h | 0 | U32 | 1000:1 | RW | no | Position Control Loop: Proportional Gain | PL.KP |
| 3548h | 0 | U32 | 1000:1 | RW | no | Velocity Control Loop: Proportional Gain | VL.KP |
| 354Bh | 0 | INT32 | 1000:1 | RW | no | Sets the velocity loop velocity feedforward gain value | VL.KVFF |
| 354Dh | 0 | INT32 | 1000:1 | RW | no | Velocity Control Loop: I-Integration Time | VL.KI |
| 3558h | 0 | INT32 | 1000:1 | RO | no | Current Monitor | IL.FB |
| 3559h | 0 | INT32 | 1000:1 | RO | no | Drive Ifold | IL.DIFOLD |
| 355Ah | 0 | INT32 | 1000:1 | RW | no | I2T Warning | IL.FOLDWTHRESH |
| 3562h | 0 | INT32 | | RW | no | Function of Digital Input 1 | DIN1.MODE |
| 3565h | 0 | INT32 | | RW | no | Function of Digital Input 2 | DIN2.MODE |
| 3568h | 0 | INT32 | | RW | no | Function of Digital Input 3 | DIN3.MODE |
| 356Bh | 0 | INT32 | | RW | no | Function of Digital Input 4 | DIN4.MODE |
| 356Eh | 0 | INT32 | 1000:1 | RW | no | Application Peak Current, positive direction | IL.LIMITP |
| 356Fh | 0 | INT32 | 1000:1 | RW | no | Application Peak Current, negative direction | IL.LIMITN |
| 3586h | 0 | U32 | | RW | no | Sets the motor temperature fault level | MOTOR.TEMPFAULT |
| 3587h | 0 | INT32 | | RW | no | Select Motor Holding Brake | MOTOR.BRAKE |
| 358Eh | 0 | U32 | 1000:1 | RW | no | Motor Continuous Current Rating | MOTOR.ICONT |
| 358Fh | 0 | U32 | 1000:1 | RW | no | Motor Peak Current Rating | MOTOR.IPEAK |
| 3593h | 0 | U32 | 1000:1 | RW | no | Sets the torque constant of the motor | MOTOR.KT |
| 3596h | 0 | U32 | 1000:1 | RO | no | Sets the proportional gain of the d-component current PI-regulator as a percentage of IL.KP | IL.KPDRATIO |
| 3598h | 0 | INT32 | 1000:1 | RW | no | Absolute Gain of Current Control loop | IL.KP |
| 359Ch | 0 | U32 | | RW | no | Sets the motor phase. | MOTOR.PHASE |
| 359Dh | 0 | U32 | | RW | no | Sets the number of motor poles | MOTOR.POLES |
| 35A3h | 0 | U32 | | RW | no | Sets the maximum motor speed | MOTOR.VMAX |
| 35A4h | 0 | INT32 | 1000:1 | RW | no | Maximum motor current | IL.MIFOLD |
| 35ABh | 0 | U32 | 1000:1 | RW | no | Sets the motor inertia | MOTOR.INERTIA |
| 35AFh | 0 | U32 | | RW | no | Sets the digital output 1 mode | MT.CNTL |
| 35B2h | 0 | U32 | | RW | no | Sets the digital output 2 mode | MT.MTNEXT |
| 35B4h | 0 | INT32 | | RW | no | Operating Mode | DRV.OPMODE |
| 35BCh | 0 | INT32 | | RW | no | Next Task Number for Motion Task 0 | MT.MTNEXT |
| 35C2h | 0 | INT32 | | RW | no | Select regen resistor | REGEN.REXT |
| 35C5h | 0 | INT32 | 1:1 | RO | no | Actual Following Error | PL.ERR |
| 35C6h | 0 | INT32 | 1:1 | RW | no | In-Position Window | MT.TPOSWND |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|---|------------------|
| 35C7h | 0 | INT32 | 1:1 | RW | no | Max. Following Error | PL.ERRFTHRESH |
| 35CAh | 0 | INT32 | | RW | no | Position Resolution (Numerator) | UNIT.PIN |
| 35CBh | 0 | INT32 | | RW | no | Position Resolution (Denominator) | UNIT.POUT |
| 35CFh | 0 | INT32 | | RW | no | reserved | PL.MODPEN |
| 35E2h | 0 | U32 | 1:1 | RW | no | Sets the current limit during homing procedure to a mechanical stop | HOME.IPEAK |
| 35EBh | 0 | INT32 | | WO | no | Save Data in EEPROM | DRV.NVSAVE |
| 35F0h | 0 | INT32 | | WO | no | Set Reference Point | HOME.SET |
| 35FEh | 0 | INT32 | | WO | no | Stop Motion Task | DRV.STOP |
| 35FFh | 0 | U32 | | RW | no | Selects between disable immediately or stop and then disable | DRV.DISMODE |
| 3610h | 0 | INT32 | | RO | no | Ambient Temperature | DRV.TEMPERATURES |
| 3611h | 0 | INT32 | | RO | no | Heat Sink Temperature | DRV.TEMPERATURES |
| 3612h | 0 | INT32 | | RO | no | Motor Temperature | MOTOR.TEMP |
| 3618h | 0 | INT32 | 1:1 | RO | no | Actual Velocity | VL.FB |
| 361Ah | 0 | INT32 | | RO | no | DC-bus voltage | VBUS.VALUE |
| 3622h | 0 | INT32 | 1:1 | RW | no | Max. Velocity | VL.LIMITP |
| 3623h | 0 | INT32 | 1:1 | RW | no | Max. Negative Velocity | VL.LIMITN |
| 3627h | 0 | INT32 | 1:1 | RW | no | Overspeed | VL.THRESH |
| 3629h | 0 | INT32 | 1000:1 | RW | no | SW1 Velocity Scaling Factor | AIN.VSCALE |
| 3637h | 0 | INT32 | 1:1 | RW | no | reserved | PL.MODP1 |
| 3638h | 0 | INT32 | 1:1 | RW | no | reserved | PL.MODP2 |
| 3656h | 0 | U64 | 1:1 | RW | no | Initial feedback position | FB1.ORIGIN |
| 3659h | 0 | INT32 | | RW | no | Type of acceleration setpoint for the system | UNIT.ACCROTARY |
| 365Bh | 0 | INT32 | | RW | no | Presetting for motion task that is processed later | MT.NUM |
| 365Fh | 0 | INT32 | | RW | no | Systemwide Definition of Velocity/Speed | UNIT.VROTARY |
| 3660h | 0 | INT32 | | RW | no | Set Resolution of the Position | UNIT.PROTARY |
| 366Eh | 0 | INT32 | | RW | no | Disable Delaytime with Holding Brake | MOTOR.TBRAKEAPP |
| 366Fh | 0 | INT32 | | RW | no | Enable Delaytime with Holding Brake | MOTOR.TBRAKERLS |
| 3683h | 0 | U16 | | RW | no | Delay for wake and shake timing | WS.TDELAY1 |
| 3685h | 0 | U16 | | RW | no | Sets delay for wake and shake timing | WS.TDELAY2 |
| 36D0h | 0 | U16 | | RW | no | Sets wake and shake current-vector appliance time | WS.T |
| 36D1h | 0 | U32 | 1:1 | RW | no | Sets the minimum movement required for wake and shake | WS.DISTMIN |
| 36E2h | 0 | U8 | | RW | no | Sets the number of repetitions for wake and shake | WS.NUMLOOPS |
| 36E5h | 0 | U32 | | RW | no | CAN baud rate selection | FBUS.PARAM01 |
| 36E6h | 0 | U32 | | RW | no | pll synchronization | FBUS.PARAM02 |
| 36E7h | 0 | U32 | | RW | no | - | FBUS.PARAM03 |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|---|--------------|
| 36E8h | 0 | U32 | | RW | no | SYNC surveillance | FBUS.PARAM04 |
| 36E9h | 0 | U32 | | RW | no | - | FBUS.PARAM05 |
| 36EAh | 0 | U32 | | RW | no | - | FBUS.PARAM06 |
| 36EBh | 0 | U32 | | RW | no | - | FBUS.PARAM07 |
| 36ECh | 0 | U32 | | RW | no | - | FBUS.PARAM08 |
| 36EDh | 0 | U32 | | RW | no | - | FBUS.PARAM09 |
| 36EEh | 0 | U32 | | RW | no | - | FBUS.PARAM10 |
| 36F6h | 0 | INT32 | | RW | no | Function of Digital Input 5 | DIN5.MODE |
| 36F9h | 0 | INT32 | | RW | no | Function of Digital Input 6 | DIN6.MODE |
| 36FCh | 0 | U32 | | RW | no | Function of Digital Input 7 | DIN7.MODE |
| 3856h | 0 | INT32 | 1:1 | RW | no | velocity window for profile position mode | MT.TVELWND |

22.7.1.4 Profile specific SDOs

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|----------------|-----------|-----------|---------------|--------|-----------|--|------------------------|
| 6040h | 0 | U16 | | WO | yes | Control word | — |
| 6041h | 0 | U16 | | RO | yes | Status word | — |
| 6060h | 0 | INT8 | | RW | yes | Modes of Operation | — |
| 6061h | 0 | INT8 | | RO | yes | Modes of Operation Display | — |
| 6063h | 0 | INT32 | | RO | yes | Position actual value (increments) | — |
| 6064h 6064h | 0 | INT32 | 1:1 | RO | yes | Position actual value (position units) | PL.FB |
| 6065h | 0 | U32 | 1:1 | RW | no | Following error window | PL.ERRFTHRESH |
| 606Bh | 0 | INT32 | 1:1 | RO | yes | Velocity demand value | VL.CMD |
| 606Ch | 0 | INT32 | 1000:1 | RO | yes | Velocity actual value (PDO in RPM) | VL.FB |
| 606Dh | 0 | U16 | | RW | yes | Velocity window | |
| 606Eh | 0 | U16 | | RW | yes | Velocity window time | |
| 6071h | 0 | INT16 | | RW | yes | Target torque | — |
| 6072h | 0 | U16 | | RW | yes | Max torque | — |
| 6073h | 0 | U16 | | RW | no | Max current | |
| 6077h | 0 | INT16 | | RO | yes | Torque actual value | — |
| 607Ah | 0 | INT32 | 1:1 | RW | yes | Target position | MT.P |
| 607Ch | 0 | INT32 | 1:1 | RW | no | Reference offset | HOME.P |
| 607Dh | | ARRAY | | | | Software position limit | |
| 607Dh | 0 | U8 | | RO | no | Number of entries | |
| 607Dh | 1 | INT32 | 1:1 | RW | no | Software position limit 1 | SWLS.LIMIT0 |
| 607Dh | 2 | INT32 | 1:1 | RW | no | Software position limit 2 | SWLS.LIMIT1 |
| 6081h 6081h | 0 | U32 | 1:1 | RW | yes | Profile Velocity | MT.V |
| 6083h | 0 | U32 | 1:1 | RW | yes | Profile Acceleration | MT.ACC |
| 6084h | 0 | U32 | 1:1 | RW | yes | Profile Deceleration | MT.DEC |
| 6086h | 0 | INT16 | | RW | yes | Motion profile type | MT.CNTL |
| 608Fh | | ARRAY | | | | Position encoder resolution | — |
| 608Fh | 0 | U8 | | RO | no | Number of entries | — |
| 608Fh | 1 | U32 | | RW | no | Encoder increments | — |
| 608Fh | 2 | U32 | | RW | no | Motor revolutions | |
| 6092h | | ARRAY | | | | Feed constant | — |
| 6092h | 0 | U8 | | RO | no | Number of entries | — |
| 6092h | 1 | U32 | | RW | no | Feed | UNIT.PIN |
| 6092h | 2 | U32 | | RW | no | Shaft revolutions | UNIT.POUT |
| 6098h 6098h | 0 | INT8 | | RW | no | Homing type | HOME.MODE HOME.DIR |
| 6099h | | ARRAY | | | | Homing velocity | — |
| 6099h | 0 | U8 | | RO | no | Number of entries | — |
| 6099h | 1 | U32 | 1:1 | RW | no | Speed while searching for limit switch | HOME.V |
| 6099h | 2 | U32 | | RW | no | Speed while searching for zero mark | HOME.FEEDRATE |
| 609Ah | 0 | U32 | 1:1 | RW | no | Homing acceleration | HOME.ACC HOME.DEC |
| 60B1h | 0 | INT32 | 1:1 | RW | yes | Velocity offset | "VL.BUSFF" (=> p. 663) |
| 60B2h | 0 | INT16 | | RW | yes | Torque offset (PDO only) | |

| Index | Sub-index | Data Type | Float Scaling | Access | PDO mapp. | Description | ASCII object |
|-------|-----------|-----------|---------------|--------|-----------|------------------------------------|---------------------------|
| 60C0h | 0 | INT8 | | RW | no | Interpolation submode select | — |
| 60C1h | | ARRAY | | | | Interpolation data record | — |
| 60C1h | 0 | U8 | | RO | no | Number of entries | — |
| 60C1h | 1 | INT32 | | RW | yes | x1, first parameter of ip function | — |
| 60C2h | | RECORD | | | | Interpolation time period | — |
| 60C2h | 0 | U8 | | RO | no | Number of entries | FBUS.SAMPLEPERIOD |
| 60C2h | 1 | U8 | | RW | no | Interpolation time units | — |
| 60C2h | 2 | INT16 | | RW | no | Interpolation time index | — |
| 60C4h | | RECORD | | | | Interpolation data configuration | — |
| 60C4h | 0 | U8 | | RO | no | Number of entries | — |
| 60C4h | 1 | U32 | | RO | no | Maximum buffer size | — |
| 60C4h | 2 | U32 | | RW | no | Actual buffer size | — |
| 60C4h | 3 | U8 | | RW | no | Buffer organization | — |
| 60C4h | 4 | U16 | | RW | no | Buffer position | — |
| 60C4h | 5 | U8 | | WO | no | Size of data record | — |
| 60C4h | 6 | U8 | | WO | no | Buffer clear | — |
| 60F4h | 0 | INT32 | | RO | yes | Following error actual value | PL.ERR |
| 60FDh | 0 | U32 | | RO | yes | Digital inputs | DIN1.MODE TO DIN7.MODE |
| 60FEh | | ARRAY | | | | Digital outputs | |
| 60FEh | 0 | U8 | | RO | no | Number of entries | |
| 60FEh | 1 | U32 | | RW | yes | Physical outputs | |
| 60FEh | 2 | U32 | | RW | no | Bit mask | |
| 60FFh | 0 | INT32 | | RW | yes | Target velocity | VL.CMDU |
| 6502h | 0 | U32 | | RO | no | Supported drive modes | — |

22.7.2 Examples

22.7.2.1 Examples, setup

All examples are valid for the AKD. All values are hexadecimal.

Basic testing of the connection to the AKD controls

When the AKD is switched on, a boot-up message is transmitted over the bus. The telegram continues to be transmitted, as long as it has not yet found a suitable receiver in the bus system.

If a CAN master is unable to recognize this message, then the following measures can be taken to test communication:

- Check the bus cable: correct characteristic impedance, correct termination resistors at both ends?
- With a multimeter: check the quiescent level of the bus cables CAN-H and CAN-L against CAN-GND (approx. 2.5 V).
- With an oscilloscope: check the output signals on CAN-H and CAN-L at the AKD. Are signals being transmitted on the bus? The voltage difference between CAN-H and CAN-L for a logical "0" is approx. 2-3 V.
- Does signal transmission stop if the master is connected?
- Check the master hardware.
- Check the master software!

Example: Operating the Status Machine**NOTE**

The status machine must be used sequentially during boot-up period. Leaving out a state (except for state "switched on") is not possible.

When the AKD is switched on and the boot-up message has been detected, communication via SDOs can be initiated. For example: all the parameters can be read out or written to, or the status machine for the drive can be controlled.

The state of the status machine can be obtained through the query of Object 6041h Sub 0.

Directly after switch-on, a value will be returned, such as 0240h. This corresponds to the status "Switch on disabled".

The following data would then be visible on the CAN bus:

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|-----------------|----------|-----------|-------|-------------|-------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 40 | 41 | 60 | 00h | 40 00 00 00 | |
| 583 | 4B | 41 | 60 | 00h | 40 02 00 00 | response telegram |
| | 2 bytes of data | | | | status | |

If the supply power is present and the hardware enable is at the High level (24 V to DGND) then you can try to switch the drive to the state "Switched on" by writing the Control word (Object 6040 Sub 0). If this is successful, there will be a positive acknowledgement in the SDO reply (control byte 0 in the data field = 60h).

Switch on

The messages then appear as follows:

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|-------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 2B | 40 | 60 | 00h | 06 00 00 00 | Shut down |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 07 00 00 00 | Switch on |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |

control word = 0x0007 meaning:

Bit 0, Bit 1, Bit 2 set => Switch On,
Disable Voltage off, Quick Stop off

Status query 2

The new status can then be queried again, and returns the following result:

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|-------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 40 | 41 | 60 | 00h | — | query status |
| 583 | 4B | 41 | 60 | 00h | 33 02 00 00 | response telegram |

Status = 0x0233 meaning:

Bit 0, Bit 1, Bit 5 set => ready to Switch On,
Bit 9 set => remote, operation possible via RS232

Example: Jog Mode via SDO

The motor shall work with constant velocity.

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|--------------------------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 2F | 60 | 60 | 00h | 03 00 00 00 | Mode of operation "Profile Velocity" |
| 583 | 60 | 60 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | FF | 60 | 00h | 00 00 00 00 | setpoint=0 |
| 583 | 60 | FF | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 06 00 00 00 | shutdown |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 07 00 00 00 | switch on |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 0F 00 00 00 | enable operation |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | FF | 60 | 00h | 00 41 00 00 | velocity setpoint |
| 583 | 60 | FF | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 0F 01 00 00 | Intermediate Stop |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |

Example: Torque Mode via SDO

The motor shall work with constant torque. CAN data:

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|----------------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 2F | 60 | 60 | 00h | 04 00 00 00 | Mode of operation "Torque" |
| 583 | 60 | 60 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 71 | 60 | 00h | 00 00 00 00 | setpoint=0 |
| 583 | 60 | 71 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 06 00 00 00 | shutdown |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 07 00 00 00 | switch on |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 0F 00 00 00 | enable operation |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 71 | 60 | 00h | 90 01 00 00 | setpoint 400 mA |
| 583 | 60 | 71 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 0F 01 00 00 | intermediate Stop |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |

Example: Jog Mode via PDO

It is useful to disable unused PDOs. In Operation Mode "Digital Velocity" a digital speed setpoint is transmitted via RXPDO. Actual position and actual speed is read via a TXPDO triggered by SYNC.

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|--|
| | byte | Low byte | High byte | index | | |
| 603 | 2F | 60 | 60 | 00h | 03 00 00 00 | mode of operation "Profile Velocity" |
| 583 | 60 | 60 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 16 | 00h | 00 00 00 00 | delete entries for the first RXPDO |
| 583 | 60 | 00 | 16 | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 00 | 16 | 01h | 20 00 FF 60 | mapping RXPDO1, Object 60FF, Subindex 0 speed setpoint, data length 32bit |
| 583 | 60 | 00 | 16 | 01h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 16 | 00h | 01 00 00 00 | confirm number of mapped objects |
| 583 | 60 | 00 | 16 | 00h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 1A | 00h | 00 00 00 00 | delete entries for the first TXPDO |
| 583 | 60 | 00 | 1A | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 00 | 1A | 01h | 20 00 64 60 | mapping TXPDO1/1, Object6064, Subindex 0 current position value in SI units, data length 32bit |
| 583 | 60 | 00 | 1A | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 00 | 1A | 02h | 20 00 6C 60 | mapping TXPDO1/2, Object606C, Subindex 0 current speed value, data length 32bit |
| 583 | 60 | 00 | 1A | 02h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 1A | 00h | 02 00 00 00 | check number of mapped objects |
| 583 | 60 | 00 | 1A | 00h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 18 | 02h | 01 00 00 00 | set TXPDO1 to synchronous, transmission with every SYNC |
| 583 | 60 | 00 | 18 | 02h | 00 00 00 00 | response telegram |
| 603 | 23 | 01 | 18 | 01h | 83 02 00 80 | disable TPDO2, set bit 31 (80h) |
| 583 | 60 | 01 | 18 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 02 | 18 | 01h | 83 03 00 80 | disable TPDO3 |
| 583 | 60 | 02 | 18 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 03 | 18 | 01h | 83 04 00 80 | disabled TPDO4 |
| 583 | 60 | 03 | 18 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 01 | 14 | 01h | 03 03 00 80 | disabled RPDO2 |
| 583 | 60 | 01 | 14 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 02 | 14 | 01h | 03 04 00 80 | disabled RPDO3 |
| 583 | 60 | 02 | 14 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 03 | 14 | 01h | 03 05 00 80 | disabled RPDO4 |
| 583 | 60 | 03 | 14 | 01h | 00 00 00 00 | response telegram |
| 000 | | | | | 01 03 | enable NMT |
| 603 | 2B | 40 | 60 | 00h | 06 00 00 00 | shutdown |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 07 00 00 00 | switch on |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 0F 00 00 00 | enable operation |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 203 | | | | | 00 40 | velocity setpoint |
| 080 | | | | | | send SYNC |

| COB-ID | Control byte | Index | | Sub- index | Data | Comment |
|--------|-----------------|----------|-----------|---------------|----------------------------|-------------------|
| | | Low byte | High byte | | | |
| 183 | | | | | FE 45 01 00 A6 AB 1A 00 | response |
| 603 | 2B | 40 | 60 | 00h | 0F 01 00 00 | intermediate stop |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |

Example: Torque Mode via PDO

It is useful to disable unused PDOs. The first TX_PDO shall transmit the actual current value with every SYNC.

| COB-ID | Control byte | Index | | Sub- index | Data | Comment |
|--------|-----------------|----------|-----------|---------------|-------------|---|
| | | Low byte | High byte | | | |
| 603 | 2F | 60 | 60 | 00h | 04 00 00 00 | Mode of operation "Torque" |
| 583 | 60 | 60 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 16 | 00h | 00 00 00 00 | delete entry for the first RXPDO |
| 583 | 60 | 00 | 16 | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 00 | 16 | 01h | 10 00 71 60 | mapping RXPDO1, Object6071, Subindex 0 current setpoint, data length 16bit |
| 583 | 60 | 00 | 16 | 01h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 16 | 00h | 01 00 00 00 | check number of mapped objects |
| 583 | 60 | 00 | 16 | 00h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 1A | 00h | 00 00 00 00 | delete entry for TXPDO1 |
| 583 | 60 | 00 | 1A | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 00 | 1A | 01h | 10 00 77 60 | mapping TXPDO1, Object6077, Subindex 0 actual current value, Data length 16bit |
| 583 | 60 | 00 | 1A | 01h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 1A | 00h | 01 00 00 00 | number of mapped objects |
| 583 | 60 | 00 | 1A | 00h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 18 | 02h | 01 00 00 00 | set TXPDO1 to synchronous, transmission with every SYNC |
| 583 | 60 | 00 | 18 | 02h | 00 00 00 00 | response telegram |
| 603 | 23 | 01 | 18 | 01h | 83 02 00 80 | disable TPDO2, set bit 31 (80h) |
| 583 | 60 | 01 | 18 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 02 | 18 | 01h | 83 03 00 80 | disable TPDO3 |
| 583 | 60 | 02 | 18 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 03 | 18 | 01h | 83 04 00 80 | disabled TPDO4 |
| 583 | 60 | 03 | 18 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 01 | 14 | 01h | 03 03 00 80 | disabled RPDO2 |
| 583 | 60 | 01 | 14 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 02 | 14 | 01h | 03 04 00 80 | disabled RPDO3 |
| 583 | 60 | 02 | 14 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 03 | 14 | 01h | 03 05 00 80 | disabled RPDO4 |
| 583 | 60 | 03 | 14 | 01h | 00 00 00 00 | response telegram |
| 000 | | | | | 01 03 | enable NMT |
| 603 | 2B | 40 | 60 | 00h | 06 00 00 00 | shutdown |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 07 00 00 00 | switch on |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 0F 00 00 00 | enable operation |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 203 | | | | | 12 02 | setpoint 530 mA |
| 080 | | | | | | send SYNC |
| 183 | | | | | 19 02 | actual value 537 mA |
| 603 | 2B | 40 | 60 | 00h | 0F 01 00 00 | intermediate stop |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |

Example: Homing via SDO

When the AKD is operated as a linear axis, a reference/homing point must be defined before positioning tasks can be executed. This must be done by executing a homing run in the Homing mode (0x6).

This example shows the procedure in the Homing mode.

Now some of the parameters that affect the homing movement are set via the bus. If you can be absolutely certain that no-one has altered the parameters in the servoamplifier, then this part can be omitted, since the servoamplifier save the data in non-volatile memory. The inputs must be configured as limit switches.

Because the dimension parameters are not finally defined in DS402, you must select these units:

UNIT.PROTARY = 3

UNIT.VROTARY = 3

UNIT.ACCROTARY = 3

The basic setup of the servoamplifier must be done with the help of the setup software before starting the homing run. The resolution has been set to 10000 $\mu\text{m}/\text{turn}$ in this example.

| COB-ID | Control | Index | | Sub-index | Data | Comment |
|--------|---------|----------|-----------|-----------|-------------|---|
| | byte | Low byte | High byte | | | |
| 703 | 00 | | | | | boot-up message |
| 603 | 40 | 41 | 60 | 00h | 00 00 00 00 | read profile status |
| 583 | 4B | 41 | 60 | 00h | 40 02 00 00 | response : 0240h |
| 603 | 23 | 99 | 60 | 01h | 10 27 00 00 | $v_{\text{ref}}=10000$ counts/s until limit switch is reached |
| 583 | 60 | 99 | 60 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 99 | 60 | 02h | 88 13 00 00 | $v_{\text{ref}}=5000$ counts/s from limit switch to zero mark |
| 583 | 60 | 99 | 60 | 02h | 00 00 00 00 | response telegram |
| 603 | 23 | 9A | 60 | 00h | 10 27 00 00 | Decel. and Accel. ramp 1000counts/s ² |
| 583 | 60 | 9A | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 7C | 60 | 00h | A8 61 00 00 | Reference offset 25000counts |
| 583 | 60 | 7C | 60 | 00h | 00 00 00 00 | response telegram |

Homing type (6098h)

| COB-ID | Control | Index | | Sub-index | Data | Comment |
|--------|---------|----------|-----------|-----------|-------------|--|
| | | Low byte | High byte | | | |
| 603 | 2F | 60 | 60 | 00h | 06 00 00 00 | mode of operation = homing |
| 583 | 60 | 60 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 40 | 41 | 60 | 00h | 00 00 00 00 | read profile status, response: 0250h Voltage Enabled |
| 583 | 4B | 41 | 60 | 00h | 40 02 00 00 | response : 0240h |
| 603 | 2B | 40 | 60 | 00h | 06 00 00 00 | Control word Transition_2,"ready to switch on". Shutdown |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 07 00 00 00 | Transition_3, "switch on". switch on |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 0F 00 00 00 | Transition_4,"operation enable" |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 40 | 41 | 60 | 00h | 00 00 00 00 | read profile status |
| 583 | 4B | 41 | 60 | 00h | 37 02 00 00 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 1F 00 00 00 | Homing_operation_start |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 40 | 41 | 60 | 00h | 00 00 00 00 | read profile status, response: homing not finished |
| 583 | 4B | 41 | 60 | 00h | 37 02 00 00 | response: homing not finished |
| 603 | 40 | 41 | 60 | 00h | 00 00 00 00 | read profile status, response: homing finished |
| 583 | 4B | 41 | 60 | 00h | 37 16 00 00 | response:homing finished |

Bit 12 in SDO 6041 indicates, whether homing is finished. Reading of the profile status is not necessary.

Example: Using the Profile Position Mode

This example shows the operation of the Profile position mode. For this, the PDOs are set as follows:
 First RPDO. No special mapping necessary, because the default mapping enters the controlword RXPDO1.
 Second RPDO

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|-------------------------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 2F | 01 | 16 | 00h | 00 00 00 01 | RPDO2: delete mapping |
| 583 | 60 | 01 | 16 | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 01 | 16 | 01h | 20 00 7A 60 | RPDO2, entry 1: target_position |
| 583 | 60 | 01 | 16 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 01 | 16 | 02h | 20 00 81 60 | RPDO2, entry 2: profile_velocity |
| 583 | 60 | 01 | 16 | 02h | 00 00 00 00 | response telegram |
| 603 | 2F | 01 | 16 | 00h | 02 00 00 00 | enter number of mapped objects |
| 583 | 60 | 01 | 16 | 00h | 00 00 00 00 | response telegram |

First TPDO

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|---------------------------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 2F | 00 | 1A | 00h | 00 00 00 01 | TPDO1: delete mapping |
| 583 | 60 | 00 | 1A | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 00 | 1A | 01h | 10 00 41 60 | TPDO1, entry 1: profile statusword |
| 583 | 60 | 00 | 1A | 01h | 00 00 00 00 | response telegram |
| 603 | 2F | 00 | 1A | 00h | 01 00 00 00 | enter number of mapped objects |
| 583 | 60 | 00 | 1A | 00h | 00 00 00 00 | response telegram |

Second TPDO

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|--|
| | byte | Low byte | High byte | index | | |
| 603 | 2F | 01 | 1A | 00h | 00 00 00 01 | TPDO2: delete mapping |
| 583 | 60 | 01 | 1A | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 01 | 1A | 01h | 20 00 64 60 | TPDO2, entry 1: position_actual_value |
| 583 | 60 | 01 | 1A | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 01 | 1A | 02h | 20 00 6C 60 | TPDO2, entry 2: velocity_actual_value |
| 583 | 60 | 01 | 1A | 02h | 00 00 00 00 | response telegram |
| 603 | 2F | 01 | 1A | 00h | 02 00 00 00 | enter number of mapped objects |
| 583 | 60 | 01 | 1A | 00h | 00 00 00 00 | response telegram |

The second TPDO should be sent with every SYNC by the servoamplifier.

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|-----------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 2F | 01 | 18 | 02h | 01 00 00 00 | TPDO2 with every SYNC |
| 583 | 60 | 01 | 18 | 02h | 00 00 00 00 | response telegram |

Disable unused TPDOs

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|-------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 23 | 02 | 18 | 01h | 83 03 00 80 | disable TPDO3 |
| 583 | 60 | 02 | 18 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 03 | 18 | 01h | 83 04 00 80 | disable TPDO4 |
| 583 | 60 | 03 | 18 | 01h | 00 00 00 00 | response telegram |

Disable unused RPDOs

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|-------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 23 | 02 | 14 | 01h | 03 04 00 80 | disable RPDO3 |
| 583 | 60 | 02 | 14 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 03 | 14 | 01h | 03 05 00 80 | disable RPDO4 |
| 583 | 60 | 03 | 14 | 01h | 00 00 00 00 | response telegram |

Define mechanical resolution via Object 6092h, Subindex 01h and 02h. Default values are the motion specific factors PGEARI and PGEARO:

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|-------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 23 | 93 | 60 | 01h | 00 00 10 00 | 2E20 increments |
| 583 | 60 | 93 | 60 | 01h | 00 00 00 00 | response telegram |
| 603 | 23 | 93 | 60 | 02h | A0 8C 00 00 | 3600 user units |
| 583 | 60 | 93 | 60 | 02h | 00 00 00 00 | response telegram |

After defining the PDOs they can be released with the NMT:

| COB-ID | Data | Comment |
|--------|-------|----------------|
| 000 | 01 03 | enable NMT |
| 183 | 40 02 | profile status |

Now the homing can be set and started.

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|--|
| | byte | Low byte | High byte | index | | |
| 603 | 2F | 60 | 60 | 00h | 06 00 00 00 | Operation mode = homing |
| 583 | 60 | 60 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 2F | 98 | 60 | 00h | 0C 00 00 00 | homing type 12, negative direction (DS402) |
| 583 | 60 | 98 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 99 | 60 | 01h | 40 19 01 00 | homing speed 72000 units/s=2s-1 |
| 583 | 80 | 99 | 60 | 01h | 31 00 09 06 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 06 00 00 00 | Transition_2,"ready to switch on".Shutdown |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 183 | | | | | 21 02 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 07 00 00 00 | Transition_3,"switch on".Switch on |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 183 | | | | | 33 02 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 0F 00 00 00 | Control word: Operation Enable |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 183 | | | | | 37 02 | response telegram |
| 603 | 2B | 40 | 60 | 00h | 1F 00 00 00 | start homing response telegram response: target reached response: homing attained |
| 583 | 60 | 40 | 60 | 00h | 00 00 00 00 | response telegram |
| 183 | | | | | 37 06 | |
| 183 | | | | | 37 16 | |

Finish homing with Control word 1_RPDO

| COB-ID | Data | Comment |
|--------|-------|---------|
| 203 | 0F 00 | |

Switch to Profile Position Mode and set ramps for positioning

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|--------------------------|
| | byte | Low byte | High byte | index | | |
| 603 | 2F | 60 | 60 | 00h | 01 00 00 00 | Profile Positioning Mode |
| 583 | 60 | 60 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 83 | 60 | 00h | 32 00 00 00 | 50ms acceleration time |
| 583 | 60 | 83 | 60 | 00h | 00 00 00 00 | response telegram |
| 603 | 23 | 84 | 60 | 00h | 32 00 00 00 | 50ms deceleration time |
| 583 | 60 | 84 | 60 | 00h | 00 00 00 00 | response telegram |

Setpoint

| COB-ID | | Data | Comment |
|--------|--|-------------|-----------------------------------|
| 303 | | 20 4E 00 00 | Pos 8CA0 =36000µm ; V= 20000 µm/s |
| 080 | | | send a SYNC |
| 283 | | BB F8 FF FF | response telegram |

Set controlword with „new setpoint“ by bit (bit 4)

| COB-ID | | Data | Comment |
|--------|--|-------|---------|
| 203 | | 1F 00 | |

Wait

| COB-ID | | Data | Comment |
|--------|--|-------|----------------------|
| 183 | | 37 12 | setpoint acknowledge |

Reset controlword with „new setpoint“ by bit (bit 4) reset

| COB-ID | | Data | Comment |
|--------|--|-------|----------------------------|
| 203 | | 0F 00 | |
| 183 | | 37 02 | reset Setpoint acknowledge |

Wait

| COB-ID | | Data | Comment |
|--------|--|-------------|--|
| 183 | | 37 06 | response: target reached |
| 080 | | | SYNC |
| 283 | | 92 FC FF FF | response: 92 FC position , FF FF speed |

Example: ASCII Communication

The following example reads the active faults from the drive (ASCII command DRV.FAULTS).

| COB-ID | Control | Index | | Sub-index | Data | Comment |
|--------|---------|----------|-----------|-----------|-------------|-----------------------------|
| | | Low byte | High byte | | | |
| 601 | 23 | 26 | 20 | 01h | 44 52 56 2E | send ASCII code "DRV." |
| 581 | 60 | 26 | 20 | 01h | 00 00 00 00 | response telegram |
| 601 | 23 | 26 | 20 | 01h | 46 41 55 4C | send ASCII code "FAUL" |
| 581 | 60 | 26 | 20 | 01h | 00 00 00 00 | response telegram |
| 601 | 23 | 26 | 20 | 01h | 54 53 0D 0A | send ASCII code "TS\r\n" |
| 581 | 60 | 26 | 20 | 01h | 00 00 00 00 | response telegram |
| 601 | 40 | 26 | 20 | 02h | 00 00 00 00 | read response |
| 581 | 43 | 26 | 20 | 02h | 3E 4E 6F 20 | read ASCII code ">No" |
| 601 | 40 | 26 | 20 | 02h | 00 00 00 00 | read response |
| 581 | 43 | 26 | 20 | 02h | 66 61 75 6C | read ASCII code "FAUL" |
| 601 | 40 | 26 | 20 | 02h | 00 00 00 00 | read response |
| 581 | 43 | 26 | 20 | 02h | 74 73 20 61 | read ASCII code "ts a" |
| 601 | 40 | 26 | 20 | 02h | 00 00 00 00 | read response |
| 581 | 43 | 26 | 20 | 02h | 63 64 69 76 | read ASCII code "ctiv" |
| 601 | 40 | 26 | 20 | 02h | 00 00 00 00 | read response |
| 581 | 43 | 26 | 20 | 02h | 66 0A 0D 0A | read ASCII code "e\r\n\r\n" |

Test for SYNC telegrams

Configuration

- Assign Target Position and Profile Velocity to a PDO (2nd receive-PDO)
- Assign Actual Position to a PDO (1st transmit-PDO), generated with every 2nd SYNC.
- Assign Status word and Manufacturer Status to a PDO (2nd transmit-PDO), generated with every 3rd SYNC.

Telegrams with the corresponding responses:

| COB-ID | Control byte | Index | | Sub-index | Data | Comment |
|--------|--------------|----------|-----------|-----------|-------------|---------------------------------------|
| | | Low byte | High byte | | | |
| 603 | 2F | 01 | 16 | 00h | 00 00 00 00 | RPDO2: delete mapping |
| 583 | 60 | 01 | 16 | 00h | 00 00 00 00 | |
| 603 | 23 | 01 | 16 | 01h | 20 00 7A 60 | RPDO2, entry 1: target position |
| 583 | 60 | 01 | 16 | 01h | 00 00 00 00 | |
| 603 | 23 | 01 | 16 | 02h | 20 00 81 60 | RPDO2, entry 2: profile velocity |
| 583 | 60 | 01 | 16 | 02h | 00 00 00 00 | |
| 603 | 2F | 01 | 16 | 00h | 02 00 00 00 | RPDO2: enter number of mapped objects |
| 583 | 60 | 01 | 16 | 00h | 00 00 00 00 | |
| 603 | 2F | 00 | 1A | 00h | 00 00 00 00 | TPDO1: delete mapping |
| 583 | 60 | 00 | 1A | 00h | 00 00 00 00 | |
| 603 | 23 | 00 | 1A | 01h | 20 00 64 60 | TPDO1: entry 1: Actual Position |
| 583 | 60 | 00 | 1A | 01h | 00 00 00 00 | |
| 603 | 2F | 00 | 1A | 00h | 01 00 00 00 | TPDO1: enter number of mapped objects |
| 583 | 60 | 00 | 1A | 00h | 00 00 00 00 | |
| 603 | 2F | 00 | 18 | 02h | 02 00 00 00 | TPDO1: send with every 2nd SYNC |
| 583 | 60 | 00 | 18 | 02h | 00 00 00 00 | |
| 603 | 2F | 01 | 1A | 00h | 00 00 00 00 | TPDO2: delete mapping |
| 583 | 60 | 01 | 1A | 00h | 00 00 00 00 | |
| 603 | 23 | 01 | 1A | 01h | 10 00 41 60 | TPDO2: entry 1: Status word |
| 583 | 60 | 01 | 1A | 01h | 00 00 00 00 | |
| 603 | 23 | 01 | 1A | 02h | 20 00 02 10 | TPDO2: entry 2: Manufacturer Status |
| 583 | 60 | 01 | 1A | 02h | 00 00 00 00 | |
| 603 | 2F | 01 | 16 | 00h | 02 00 00 00 | TPDO2: enter number of mapped objects |
| 583 | 60 | 01 | 16 | 00h | 00 00 00 00 | |
| 603 | 2F | 01 | 18 | 02h | 03 00 00 00 | TPDO2: send with every 3rd SYNC |
| 583 | 60 | 01 | 18 | 02h | 00 00 00 00 | |

SYNC-Object

| COB-ID | Comment |
|--------|--|
| 080 | Object 181 (TPDO 1) appears at every 2 nd SYNC Object 281 (TPDO 2) appears at every 3 rd SYNC |

Emergency-Object

If, for instance, the resolver connector is disconnected, a serious error will be caused in the controller. This results in an Emergency telegram.

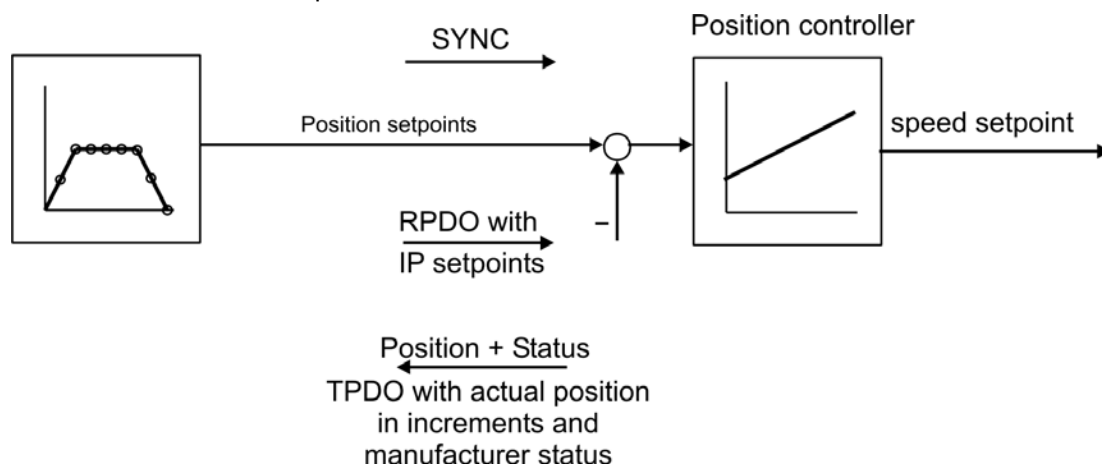
| COB-ID | Emergency error | | Error register | Error register | |
|--------|-----------------|------|----------------|----------------|---|
| | Low | High | | | |
| 081 | 10 | 43 | 08 | 00 00 00 00 | motor temperature, temperature, manufacturer specific |
| 081 | 00 | 00 | 88 | 00 00 00 00 | |

22.7.2.2 Examples, special applications

Example: External Trajectory with Interpolated Position Mode

This example shows the possible application for giving two axes position setpoints within one PDO.

Controller structure for the position controller within the drive:



All data are hexadecimal. In the example, the two axes in the system have the station addresses 1 and 2.

Before you begin this procedure, the axes should be homed (just for this example).

The common PDO contains 2 IP (interpolated position) – setpoints and can be transmitted simultaneously to two stations, whereby each station can extract the relevant data. The other data can be made ignored by using dummy entries (Object 2100 sub 0). For this purpose both axes must react on the same RPDO-COB-ID.

Action

Do the RPDO2-mapping for both axis:

Axis 1:

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|--|
| | byte | Low byte | High byte | index | | |
| 601 | 2F | 01 | 16 | 00h | 00 00 00 00 | RPDO2: delete mapping |
| 581 | 60 | 01 | 16 | 00h | 00 00 00 00 | |
| 601 | 23 | 01 | 16 | 01h | 20 01 C1 60 | RPDO2, entry 1: IP setpoint axis 1 |
| 581 | 60 | 01 | 16 | 01h | 00 00 00 00 | |
| 601 | 23 | 01 | 16 | 02h | 20 00 00 21 | RPDO2, entry 2: Dummy entry 4 bytes |
| 581 | 60 | 01 | 16 | 02h | 00 00 00 00 | |
| 601 | 2F | 01 | 16 | 00h | 02 00 00 00 | RPDO2, enter number of mapped objects |
| 581 | 60 | 01 | 16 | 00h | 00 00 00 00 | |

Axis 2:

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|--|
| | byte | Low byte | High byte | index | | |
| 602 | 2F | 01 | 16 | 00h | 00 00 00 00 | RPDO2: delete mapping |
| 582 | 60 | 01 | 16 | 00h | 00 00 00 00 | |
| 602 | 23 | 01 | 16 | 02h | 20 00 00 21 | RPDO2, entry 1: Dummy entry 4 bytes |
| 582 | 60 | 01 | 16 | 02h | 00 00 00 00 | |
| 602 | 23 | 01 | 16 | 01h | 20 01 C1 60 | RPDO2, entry 2: IP setpoint axis 2 |
| 582 | 60 | 01 | 16 | 01h | 00 00 00 00 | |
| 602 | 2F | 01 | 16 | 00h | 02 00 00 00 | RPDO2, enter number of mapped objects |
| 582 | 60 | 01 | 16 | 00h | 00 00 00 00 | |
| 602 | 23 | 01 | 16 | 01h | 01 03 00 00 | RPDO2: Set COB-ID identical to axis 1 |
| 582 | 60 | 01 | 16 | 01h | 00 00 00 00 | |

Now both axis react to the same COB-identifier 0x301, axis 1 takes byte 0 to 3 as IP setpoint, axis 2 takes byte 4 to 7. The second TPDOs shall contain the actual position in increments and the manufacturer status.

Mapping configuration for axis 1:

| COB-ID | Control | Index | | Sub-index | Data | Comment |
|--------|---------|----------|-----------|-----------|-------------|--|
| | byte | Low byte | High byte | | | |
| 601 | 2F | 01 | 1A | 00h | 00 00 00 00 | TPDO2: delete mapping |
| 581 | 60 | 01 | 1A | 00h | 00 00 00 00 | |
| 601 | 23 | 01 | 1A | 01h | 20 00 63 60 | TPDO2, entry 1: actual position in increments |
| 581 | 60 | 01 | 1A | 01h | 00 00 00 00 | |
| 601 | 23 | 01 | 1A | 02h | 20 00 02 10 | TPDO2, entry 2: Dummy entry 4 bytes |
| 581 | 60 | 01 | 1A | 02h | 00 00 00 00 | |
| 601 | 2F | 01 | 1A | 00h | 02 00 00 00 | TPDO2, enter number of mapped objects |
| 581 | 60 | 01 | 1A | 00h | 00 00 00 00 | |

The same must be done for axis 2.

Here it is assumed that both drives accept new trajectory values with every SYNC command, and must return their incremental position and manufacturer status values. The communication parameters must be set accordingly.

Axis 1:

| COB-ID | Control | Index | | Sub-index | Data | Comment |
|--------|---------|----------|-----------|-----------|-------------|---|
| | byte | Low byte | High byte | | | |
| 601 | 2F | 01 | 14 | 02h | 01 00 00 00 | RPDO2 axis 1, reaction on every sync |
| 581 | 60 | 01 | 14 | 02h | 00 00 00 00 | |
| 602 | 2F | 01 | 14 | 02h | 01 00 00 00 | RPDO2 axis 2, reaction on every sync |
| 582 | 60 | 01 | 14 | 02h | 00 00 00 00 | |
| 601 | 2F | 01 | 18 | 02h | 01 00 00 00 | TPDO2 axis 1, reaction on every sync |
| 581 | 60 | 01 | 18 | 02h | 00 00 00 00 | |
| 602 | 2F | 01 | 18 | 02h | 01 00 00 00 | TPDO2 axis 2, reaction on every sync |
| 582 | 60 | 01 | 18 | 02h | 00 00 00 00 | |

The other Tx-PDOs 3 and 4 should be switched off to minimize bus-load:

| COB-ID | Control | Index | | Sub-index | Data | Comment |
|--------|---------|----------|-----------|-----------|-------------|------------------|
| | byte | Low byte | High byte | | | |
| 601 | 23 | 02 | 18 | 01h | 81 03 00 80 | Switch off TPDO3 |
| 581 | 60 | 02 | 18 | 01h | 00 00 00 00 | |
| 601 | 23 | 03 | 18 | 01h | 81 04 00 80 | Switch off TPDO4 |
| 581 | 60 | 03 | 18 | 01h | 00 00 00 00 | |

The same must be done for axis 2.

In order to be able to make trajectory movements, both drives must be operating in the appropriate mode. This is set through Index 6060h:

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|------------------------|
| | byte | Low byte | High byte | index | | |
| 601 | 2F | 60 | 60 | 00h | 07 00 00 00 | Set IP mode for axis 1 |
| 581 | 60 | 60 | 60 | 00h | 00 00 00 00 | |
| 602 | 2F | 60 | 60 | 00h | 07 00 00 00 | Set IP mode for axis 2 |
| 582 | 60 | 60 | 60 | 00h | 00 00 00 00 | |

The cycle interval for the IP-mode shall be 1 ms. This must be defined with Object 60C1 sub 1 and 2:

| COB-ID | Control | Index | | Sub- | Data | Comment |
|--------|---------|----------|-----------|-------|-------------|--|
| | byte | Low byte | High byte | index | | |
| 601 | 2F | C2 | 60 | 01h | 01 00 00 00 | Interpolation time unit 1 |
| 581 | 60 | C2 | 60 | 01h | 00 00 00 00 | |
| 601 | 2F | C2 | 60 | 02h | FD 00 00 00 | Interpolation time index -3 -> Cycle time = $1 * 10^{-3}$ s |
| 581 | 60 | C2 | 60 | 02h | 00 00 00 00 | |

The same must be done for axis 2.

To start up the axes, the drives must be put into the operational status (operation enable) and the network management functions must be started.

The network management functions enable the application of the Process Data Objects (PDOs) and are initialized by the following telegram for both axes:

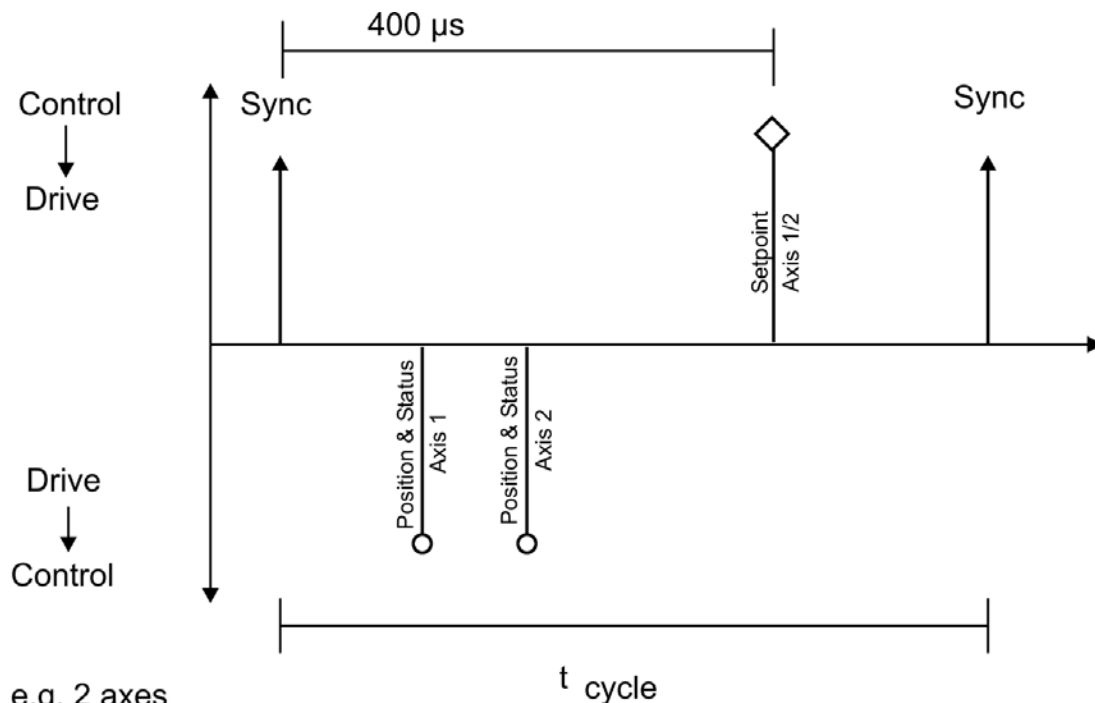
Switch the NMT (Network Management) status machine to operation enable:

| COB-ID | Command specifier (CS) | Node-ID | Comment |
|--------|------------------------|---------|-------------------------|
| 0 | 1 | 1 | NMT enable for all axes |

Next, power is applied to each drive, and they are put into the operation enable condition. This should be done in steps with waiting for the appropriate reaction of the drive (e.g. axis 1):

| COB-ID | Data | Comment |
|--------|-------|--------------------------|
| 201 | 06 00 | Shutdown command |
| 181 | 31 02 | State Ready_to_switch_on |
| 201 | 07 00 | Switch_on command |
| 181 | 33 02 | State Switched_on |
| 201 | 0F 00 | Enable_operation command |
| 181 | 37 02 | State Operation_enabled |
| 201 | 1F 00 | Enable IP-mode |
| 181 | 37 12 | IP-mode enable |

The configuration above now enables a cyclical sequence, as shown in the diagram:



e.g. 2 axes

t_{cycle} 1 ms per axis at 1 MBaud

RPDO 2 can now be used to supply trajectory data for both axes, e.g.:

| COB-ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| 301 | F4 | 01 | 00 | 00 | E8 | 03 | 00 | 00 |

In this example, the first axis receives a trajectory value of 500 increments (Bytes 0 to 3) and the second axis receives a trajectory value of 1000 increments.

The axes accept these values, and the positioning is made when the next SYNC telegram is received.

SYNC telegram

| COB-ID |
|--------|
| 080 |

Afterwards, both axes send back their incremental positions and the contents of their status registers when the SYNC Object with the COB-ID for the 2ndTPDO is received.

| COB-ID | Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 | Byte 7 | Comment |
|--------|--------|--------|--------|--------|--------|--------|--------|--------|--|
| 181 | 23 | 01 | 00 | 00 | 00 | 00 | 03 | 44 | position + manufacturer status register for axis 1 |
| 182 | A5 | 02 | 00 | 00 | 00 | 00 | 03 | 44 | position + manufacturer status register for axis2 |

If an error occurs during operation, the axis concerned transmits an Emergency message, which could appear like this:

Emergency Object

| COB-ID | Emergency error | | Error register | Category | |
|--------|-----------------|------|----------------|----------|--|
| | Low | High | | | |
| 081 | 10 | 43 | 08 | 01 | 00 00 00 00 motor temperature, temperature, manufacturer-specific |
| 081 | 00 | 00 | 08 | 00 | 00 00 00 00 |

22.8 Index

1

| | |
|------------|-----|
| 1000h | 163 |
| 1001h | 164 |
| 1002h | 165 |
| 1003h | 166 |
| 1005h | 167 |
| 1006h | 167 |
| 1008h | 168 |
| 1009h | 168 |
| 100Ah | 168 |
| 100Ch | 169 |
| 100Dh | 169 |
| 1010h | 170 |
| 1014h | 171 |
| 1016h | 171 |
| 1017h | 172 |
| 1018h | 172 |
| 1026h | 174 |
| 1400-1403h | 185 |
| 1600-1603h | 186 |
| 1800-1803h | 188 |
| 1A00-1A03h | 190 |

2

| | |
|------------|-----|
| 2014-2017h | 175 |
| 2018h | 176 |
| 2026h | 177 |
| 204Ch | 200 |
| 20A0h | 178 |
| 20A1h | 178 |
| 20A2h | 178 |
| 20A3h | 179 |
| 20A4h | 179 |
| 20A5h | 180 |
| 20A6h | 180 |
| 20B8h | 181 |

6

| | |
|-------|-----|
| 6040h | 195 |
| 6041h | 196 |
| 6060h | 198 |
| 6061h | 199 |
| 6063h | 207 |
| 6064h | 208 |
| 6065h | 208 |
| 606Ch | 203 |
| 6071h | 205 |
| 6073h | 205 |
| 6077h | 206 |
| 607Ah | 219 |
| 607Ch | 214 |

| | |
|-------|-----|
| 607Dh | 219 |
| 6081h | 220 |
| 6083h | 220 |
| 6084h | 220 |
| 608Fh | 201 |
| 6092h | 202 |
| 6098h | 214 |
| 6099h | 216 |
| 609Ah | 216 |
| 60C0h | 209 |
| 60C1h | 210 |
| 60C2h | 211 |
| 60C4h | 212 |
| 60F4h | 208 |
| 60FDh | 182 |
| 60FEh | 182 |
| 60FFh | 204 |
| 6502h | 183 |

A

| | |
|----------------------------|---------|
| Abbreviations | 60, 137 |
| Acceleration ramp | 359 |
| AL Event | 80 |
| Analog inputs | 275 |
| Analog output | 103 |
| Analog setpoints | 275 |
| Anti-resonance filter | 662 |
| Aux. supply 24V, interface | 267 |

B

| | |
|-----------------------------|-----|
| Basic data types | 147 |
| Basic drive setup | 34 |
| Basic testing | 236 |
| Blink | 362 |
| Block diagrams | |
| current loop | 282 |
| position and velocity loops | 282 |
| Brake view | 80 |

C

| | |
|-----------------------------------|--------|
| CANopen over EtherCAT | 84 |
| Capture | 81 |
| Check communications | 40, 43 |
| Command buffer | 96 |
| Command source | 366 |
| Communication Objects | 149 |
| Connect to another drive | 40 |
| Connected and disconnected states | 38 |
| Connection diagram | |
| AKD-x00306 to x00606 | 264 |
| AKD-x01206 | 265 |
| AKD-x01206 & AKD-xzzz07 | 266 |
| Connection Diagrams | 263 |
| Control word | 195 |

| | | | |
|--|-----------------|---|-----------------|
| Control Word Commands | 85 | Foldback | 77 |
| Controlled Stop | 115 | G | |
| current loop | 121 | General definitions | 163 |
| Current loop gain scheduling | 122 | General objects | 163 |
| Cycle Time | | H | |
| Adjust | 89 | Heartbeat | 158 |
| Max.Values | 89 | Homing | 134 |
| Cyclical Values | 88 | Homing Mode | 214 |
| D | | I | |
| Data types | 147 | I/O-Connection | 104, 273 |
| DC bus link, interface | 270 | Inputs | |
| deceleration | 369 | Analog | 275 |
| Device control | 192 | Digital | 99 |
| Device Not Shown | 38, 46 | Enable | 102 |
| Dig. Encoder emulation, interface | 277 | Programmable | 102 |
| Digital inputs | 99 | Interpolated position mode | 209 |
| Digital Inputs and Outputs | 88 | Interrupt Event | 80 |
| Direct DC mains | 56 | Invalid Firmware | 259 |
| Disable | 373 | J | |
| Disconnected state | 38 | Jog Move | 153 |
| Display Codes | 34 | K | |
| Downloading Firmware | 259 | KAS IDE | 77 |
| Drive Motion Status | 153 | L | |
| Dynamic Braking | 117 | Latch Words | 91 |
| E | | M | |
| Electronic gearing | 106 | Macros | 219 |
| Emergency Messages | 253, 160 | Mailbox | 92 |
| Emergency Object | 151 | Mains supply, interface | 271 |
| emergency timeout | 375 | Manufacturer specific objects | 175 |
| Emulated Encoder Output | 277 | master/slave | 74 |
| ENABLE | 102 | Mixed data types | 148 |
| Enable/Disable | 111 | Modbus | 6 |
| encoder emulation | 74 | Modbus Dynamic Mapping | 9 |
| Error Messages | | Modulo position | 127 |
| parameter and command | 250 | Motion Tasks | 144 |
| EtherCAT onboard | 65 | Motor | 64 |
| EtherCAT Profile | 78 | Motor interface | 268 |
| Examples | 236 | N | |
| Examples, setup | 236 | Network Management Object | 150 |
| Examples, special applications | 249 | Nodeguard | 157 |
| Extended data types | 149 | Non-Plug and Play Feedback Devices | 76 |
| F | | | |
| Factor Groups | 199 | | |
| Fault messages | 240 | | |
| Faults | 364 | | |
| clearing | 249 | | |
| Feedback 1 | 66 | | |
| Feedback 2 | 73 | | |
| Feedback Connection | 279 | | |
| Fieldbus | 96 | | |
| Fieldbus Parameters | 96 | | |

O

| | |
|-----------------------------------|------------|
| Object Dictionary | 223 |
| Operating mode | 198 |
| parameters and commands active in | 225, |
| | 265 |
| Operation Modes | 88 |
| Outputs | |
| Analog | 274 |
| Overview | 50 |

P

| | |
|-----------------------------------|----------------|
| Parameter Comparer | 221 |
| Parameter Load/Save | 221 |
| PDO configuration | 184 |
| PDO Mapping | 87 |
| Performance Servo Tuner | 160 |
| advanced | 164 |
| Phase run-up | 82 |
| Position Control Function | 207 |
| Position Loop | 126 |
| Power | 56 |
| Process Data Object | 155 |
| Profile Position Mode | 218 |
| Profile specific objects | 182 |
| Profile torque mode | 205 |
| Profile Velocity Mode | 203 |
| Programmable Limit Switch | 109 |
| Prohibited Use | 62, 139 |
| Pulse Direction, interface | 278 |

R

| | |
|----------------------------------|------------|
| Receive PDOs | 185 |
| Regen resistor, interface | 269 |
| Regeneration | 58 |
| Registration moves | 149 |
| Response monitoring | 169 |

S

| | |
|----------------------------------|------------|
| Safe Torque Off (STO) | 118 |
| Safety Instructions | |
| Electrical Installation | 64, 141 |
| General | 62, 139 |
| Save On Disconnect | 157 |
| Save On Exit | 156 |
| Save On Firmware Download | 158 |
| Save Options | 156 |
| scope | 205 |
| channels | 205 |
| settings and presets | 212 |
| time-base | 205 |
| SDO abort codes | 154 |
| Service Data Object | 152 |
| Service Motion | 152 |

| | |
|-------------------------------|------------|
| Settings | 52 |
| Setup | 66 |
| Signed Integer | 148 |
| Slave Register | 79 |
| Slider Tuning | 160 |
| Status machine | 192 |
| Status Machine | 84 |
| Status word | 196 |
| Status Word | 86 |
| Synchronization | 90 |
| Synchronization Object | 150 |

T

| | |
|---------------------------|----------------|
| Target group | 59, 136 |
| TCP/IP | 48 |
| Terminal | 218 |
| Time Stamp Object | 150 |
| Transmission modes | 156 |
| Transmit PDOs | 188 |
| Trigger modes | 156 |
| Trouble Shooting | 257 |
| Tuning | 189 |
| TwinCAT | 66 |

U

| | |
|-------------------------|----------------|
| Units | 84 |
| Unsigned Integer | 147 |
| Up/Down Input | 278 |
| Use as directed | 62, 139 |

V

| | |
|----------------------|------------|
| Velocity Loop | 124 |
|----------------------|------------|

W

| | |
|-------------------------------|------------|
| Wake and shake (WS) | 68 |
| Warnings | 240 |
| Watch | 52 |
| Wizard | 35 |
| Workbench over TwinCAT | 71 |

This page intentionally left blank.

This page intentionally left blank.

Sales and Service

We are committed to quality customer service. In order to serve in the most effective way, please contact your local sales representative for assistance.

If you are unaware of your local sales representative, please contact us.

Europe

Kollmorgen Customer Support Europe

Internet www.kollmorgen.com
E-Mail technik@kollmorgen.com
Tel.: +49(0)2102 - 9394 - 0
Fax: +49(0)2102 - 9394 - 3155

North America

Kollmorgen Customer Support North America

Internet www.kollmorgen.com
E-Mail support@kollmorgen.com
Tel.: +1 - 540 - 633 - 3545
Fax: +1 - 540 - 639 - 4162



Appendix B

Parameter and Command Reference Guide

About the Parameter and Command Reference Guide

This reference guide provides descriptive information about each parameter and command used in the drive firmware. Parameters and commands are used to configure the drive or to return status information from the drive using the WorkBench terminal screen. The use of these parameters and commands to perform various drive functions is detailed in related sections of the AKD User Guide.

Drive parameters and commands include the following:

| | | |
|-----------------------------|----------------------------|-----------------------------|
| AIN Parameters | FB2 Parameters | PLS Parameters and Commands |
| AIO Parameters | FBUS Parameters | REC Parameters and Commands |
| AOUT Parameters | GEAR Parameters | REGEN Parameters |
| BODE Parameters | GUI Parameters | SM Parameters |
| CAP Parameters | HOME Parameters | STO Parameters |
| CS Parameters | HWLS Parameters | SWLS Parameters |
| DIN Parameters | IL Parameters | UNIT Parameters |
| DOUT Parameters | MOTOR Parameters | VBUS Parameters |
| DRV Parameters and Commands | MT Parameters and Commands | VL Parameters |
| FB1 Parameters | PL Parameters | WS Parameters |

A summary table of basic information for all parameters and commands is also available:

[Summary of Parameters and Commands](#)

For each parameter or command, this reference guide presents the following tables of information, followed by a description of the command, examples, and links to related information in the User Guide, as appropriate.

| General Information | |
|----------------------|--|
| Type | One of four types: <ul style="list-style-type: none"> • Command: Action or W/O command. • NV Parameter: R/W and stored in nonvolatile (NV) memory • R/W Parameter: Can be either read from or written to the drive. • R/O Parameter. Can only be read from the drive |
| Description | Brief description of the parameter or command and notes if the parameter or command is not active in all opmodes. |
| Units | Appropriate units (see Table of Units for unit descriptions) |
| Range | Permissible range; multiple ranges are sometimes present. |
| Default Value | Determined at setup process time or motor ID; otherwise set to 0.010. |
| Data Type | Integer, Boolean, Float, or String |
| See Also | Links to related information such as other parameters, block diagrams, schematics, or other sections of the product manual. |
| Start Version | The minimum firmware version number required to use the parameter or command |

| Fieldbus | Index/Subindex | Object Start Version |
|--|---|---|
| Fieldbus type, such as EtherCAT COE and CAN-open or Modbus. | Index/subindex values for the parameter or command. The index value may be linked to the Object Dictionary for each fieldbus, if the object dictionary contains more detailed information about the object. | The minimum firmware version number required to use the fieldbus. |

Additional data types may include the following:

| Type | Description |
|-------|-----------------------|
| Error | Illegal type=0 |
| b | Boolean |
| U8 | 8 x unsigned numbers |
| S8 | 8 x signed numbers |
| U16 | 16 x unsigned numbers |
| S16 | 16 x signed numbers |
| U32 | 32 x unsigned numbers |
| S32 | 32 x signed numbers |
| U64 | 64 x unsigned numbers |
| S64 | 64 x signed numbers |

Parameter and Command Naming Conventions

| Abbreviation | Term |
|--------------|----------------------------------|
| ACC | Acceleration |
| APP | Apply |
| CLR | Clear |
| CS | Controlled Stop |
| I | Current |
| D | Current d-component |
| DEC | Deceleration |
| DIR | Direction |
| DIS | Disable |
| DIST | Distance |
| EMUE | Emulated encoder |
| EN | Enable |
| ERR | Error |
| F | Fault |
| FB | Feedback |
| FF | Feedforward |
| K | Gain |
| INT | Integrator |
| LIM | Limit |
| L | Loop |
| MAX | Maximum |
| MIN | Minimum |
| N | Negative |
| NV | Nonvolatile |
| P | Position, Proportional, Positive |
| RLS | Release |
| R | Resistance |
| STATE | Status, State, Stat |
| THRESH | Threshold |
| T | Time |
| TMAX | Timeout |
| U | User |
| V | Velocity, Volt |
| W | Warning |

Summary of Parameters and Commands

This table contains an alphabetical list of parameters and commands, with a brief description for each. The parameter name and description are linked to the parameter tables. The description field also notes if a parameter is not active in all opmodes. Generally speaking, all parameters and commands are active in all opmodes, with the following exceptions:

| Parameter or Command | Active in Opmodes |
|------------------------------------|-------------------|
| GEAR (all parameters and commands) | 2 (position) only |
| HOME (all parameters and commands) | 2 (position) only |
| MT (all parameters and commands) | 2 (position) only |

| Parameter or Command | Active in Opmodes |
|----------------------------------|------------------------------------|
| SM.I1, SM.I | 0 (torque) only |
| SM.V1, SM.V2 | 1 (velocity) only |
| SM.VPM1, SM.VPM2 | 2 (position) only |
| VL (all parameters and commands) | 1 (velocity) and 2 (position) only |

| Parameter or Command | Type | Description |
|----------------------------------|---------|--|
| Analog Input (AIN) | | |
| AIN.CUTOFF | NV | Sets the analog input low-pass filter cutoff frequency. |
| AIN.DEADBAND | NV | Sets the analog input signal deadband. |
| AIN.DEADBANDMODE | NV | Sets the analog input deadband mode. |
| AIN.ISCALE | NV | Sets the analog current scale factor. |
| AIN.OFFSET | NV | Sets the analog input offset. |
| AIN.PSCALE | NV | Sets the analog position scale factor. |
| AIN.VALUE | R/O | Reads the value of the analog input signal. |
| AIN.VSCALE | NV | Sets analog velocity scale factor. |
| AIN.ZERO | Command | Zeroes the analog input signal. |
| Analog Input/Output (AIO) | | |
| AIO.ISCALE | NV | Sets the analog current scale factor. |
| AIO.VSCALE | NV | Sets velocity scale factor. |
| AIO.PSCALE | NV | Sets position scale factor. |
| Analog Output (AOUT) | | |
| AOUT.DEBUGADDR | NV | Sets the memory address to debug. |
| AOUT.DEBUGDATATYPE | NV | Sets the data type of the value to be debugged. |
| AOUT.DEBUGSCALE | NV | Sets the scale to be used for debug. |
| AOUT.ISCALE | NV | Sets the analog current scale factor. |
| AOUT.MODE | NV | Sets the analog output mode. |
| AOUT.OFFSET | NV | Sets the analog input offset. |
| AOUT.PSCALE | NV | Sets the analog position scale factor. |
| AOUT.VALUE | NV | Reads the analog output value. |
| AOUT.VALUEU | R/W | Sets the analog output value. |
| AOUT.VSCALE | NV | Sets velocity scale factor for analog output. |
| Bode plot (BODE) | | |
| BODE.EXCITEGAP | R/W | Controls how often the excitation is updated. |
| BODE.FREQ | R/W | Sets the frequency of the sine excitation source. |
| BODE.IAMP | R/W | Sets current command value used during the Bode procedure. |
| BODE.IFLIMIT | | Sets the current fault duration limit in seconds for the BODE.MODE 5 stability test. |
| BODE.IFTHRESH | | Sets the current fault threshold for the BODE.MODE 5 stability test. |
| BODE.INJECTPOINT | R/W | Sets whether the excitation uses current or velocity excitation type. |
| BODE.MODE | R/W | Sets the mode of the excitation. |
| BODE.MODETIMER | R/W | Sets the watchdog timer of the excitation. |
| BODE.PRBDDEPTH | R/W | Sets the length of the PRB signal before it repeats. |
| BODE.VAMP | R/W | Sets the amplitude of the excitation when in velocity mode. |

| Parameter or Command | Type | Description |
|--------------------------------|------|---|
| BODE.VFLIMIT | | Sets the velocity fault duration limit (seconds) for the BODE.MODE 5 stability test |
| BODE.VFTHRESH | | Sets the current fault threshold for the BODE.MODE 5 stability test. |
| Capture (CAP) | | |
| CAP0.EDGE, CAP1.EDGE | R/W | Selects the capture edge. |
| CAP0.EN, CAP1.EN | R/W | Enables or disables the related capture engine. |
| CAP0.EVENT, CAP1.EVENT | | Controls the precondition logic. |
| CAP0.FILTER, CAP1.FILTER | R/W | Controls the precondition logic. |
| CAP0.MODE, CAP1.MODE | NV | Selects the captured value. |
| CAP0.PLFB, CAP1.PLFB | R/O | Reads captured position value. |
| CAP0.PREEDGE, CAP1.PREEDGE | | Selects the capture precondition edge. |
| CAP0.PREFILTER, CAP1.PREFILTER | R/W | Sets the filter for the precondition input source. |
| CAP0.PRESELECT, CAP1.PRESELECT | R/W | Sets the precondition trigger. |
| CAP0.STATE, CAP1.STATE | R/O | Indicates whether or not trigger source was captured. |
| CAP0.T, CAP1.T | R/O | Reads time capture (if time capture was configured). |
| CAP0.TRIGGER, CAP1.TRIGGER | R/W | Specifies the trigger source for the position capture. |
| Controlled Stop (CS) | | |
| CS.DEC | NV | Sets the deceleration value for the controlled stop process. |
| CS.STATE | NV | Returns the internal status of the controlled stop process. |
| CS.TO | NV | Sets the time value for the drive velocity to be within CS.VTHRESH. |
| CS.VTHRESH | NV | Sets the velocity threshold for the controlled stop. |
| Digital Input (DIN) | | |
| DIN.ROTARY | R/O | Reads the rotary knob value. |
| DIN.STATES | R/O | Reads the digital input states. |
| DIN1.FILTER TO DIN7.FILTER | R/W | Filter mode for digital inputs 1 to 7. |
| DIN1.INV TO DIN7.INV | R/W | Sets the indicated the polarity of a digital input mode. |
| DIN1.MODE TO DIN7.MODE | NV | Sets the digital input modes. |
| DIN1.PARAM TO DIN7.PARAM | R/W | Sets a value used as an extra parameter for digital inputs nodes. |
| DIN1.STATE TO DIN7.STATE | R/O | Reads a specific digital input state. |
| Digital Output (DOUT) | | |
| DOUT.CTRL | NV | Sets the source of digital outputs (firmware or fieldbus). |
| DOUT.RELAYMODE | R/W | Indicates faults relay mode. |
| DOUT.STATES | R/O | Reads the state of the two digital outputs. |
| DOUT1.MODE AND DOUT2.MODE | NV | Sets the digital output mode. |

| Parameter or Command | Type | Description |
|-------------------------------------|---------|--|
| DOUT1.PARAM AND DOUT2.PARAM | NV | Sets extra parameters for the digital outputs. |
| DOUT1.STATE AND DOUT2.STATE | R/O | Reads the digital output state. |
| DOUT1.STATEU AND DOUT2.STATEU | R/W | Sets the state of the digital output node. |
| Drive (DRV) | | |
| DRV.ACC | NV | Describes the acceleration ramp for the velocity central loop. |
| DRV.ACTIVE | R/O | Reads the enable status of an axis. |
| DRV.BLINKDISPLAY | Command | Causes the display to blink for 10 seconds. |
| DRV.BURNINOFFT (Password Protected) | R/W | Sets the burn-in test mode off time. |
| DRV.CLRFAULTHIST | Command | Clears the fault history log in the NV. |
| DRV.CLRFAULTS | Command | Tries to clear all active faults in the drive. |
| DRV.CMDDELAY | R/W | Issues a delay before next command is executed. |
| DRV.CMDSOURCE | NV | Sets the command source (service, fieldbus, analog input, gearing, digital, or Bode). |
| DRV.CRASHDUMP | Command | Retrieves diagnostic information after the drive crashes. |
| DRV.DBILIMIT | NV | Sets the maximum amplitude of the current for dynamic braking. |
| DRV.DEC | NV | Sets the deceleration value for the velocity loop. |
| DRV.DIR | R/W | Changes drive direction. |
| DRV.DIS | Command | Disables the axis (software). |
| DRV.DISMODE | NV | Selects among disable immediately or stop and then disable options. |
| DRV.DISSOURCES | R/O | Returns the possible reason for a drive disable. |
| DRV.DISTO | R/W | Sets the emergency timeout |
| DRV.EMUEDIR | R/W | Sets the direction of the emulated encoder output (EEO) signal. |
| DRV.EMUEMODE | R/W | Sets the mode of the emulated encoder output (EEO) connector. |
| DRV.EMUEMTURN | R/W | Defines the location of the index pulse on the EEO (emulated encoder output) when DRV.EMU-EMODE=2. |
| DRV.EMUERES | R/W | Sets the resolution of the EEO (emulated encoder output). |
| DRV.EMUEZOFFSET | R/W | Sets the location of the EEO (emulated encoder output) index pulse (when DRV.EMU-EMODE=1). |
| DRV.EN | Command | Enables the axis (software). |
| DRV.ENDEFAULT | R/W | Sets the default state of the software enable. |
| DRV.FAULTHIST | R/O | Reads the last 10 faults from NV memory. |
| DRV.FAULTS | R/O | Reads the active faults. |
| DRV.FLASHREAD (Password Protected) | R/O | Reads a value from the serial flash memory. |
| DRV.HANDWHEEL | R/O | Reads the EEO input value. |

| Parameter or Command | Type | Description |
|-------------------------|---------|---|
| DRV.HELP | R/O | Reads the minimum, maximum, and default values for a specific parameter or command. |
| DRV.HELPALL | R/O | Retrieves the minimum, maximum, default, and actual values for all available parameters and commands. |
| DRV.ICONT | R/O | Reads the continuous rated current value. |
| DRV.INFO | R/O | Reads general information about the drive. |
| DRV.IPEAK | R/O | Reads the peak rated current value. |
| DRV.IZERO | R/W | Sets the current that will be used during the DRV.ZERO procedure. |
| DRV.LIST | R/O | Reads the list of available parameters and commands. |
| DRV.LOGICVOLTS | | Reads the logic voltages. |
| DRV.NAME | NV | Sets and reads the name of the drive. |
| DRV.NVLIST | R/O | Lists the NV parameters and values from the RAM. |
| DRV.NVLOAD | W/O | Loads all data from the NV memory of the drive into the RAM parameters. |
| DRV.NVSAVE | Command | Saves the drive parameters from the RAM to the NV memory. |
| DRV.ONTIME | R/O | Returns how long the drive has been running since last power up. |
| DRV.OPMODE | NV | Sets the drive operation mode (current, velocity, or position). |
| DRV.READFORMAT | R/W | Sets the value returned to either decimal or hexadecimal. |
| DRV.RSTVAR | Command | Sets default values in the drive without re-booting the drive and without resetting the NV memory. |
| DRV.RUNTIME | R/O | Returns how long the drive has been running since first activated. |
| DRV.SETUPREQBITS | R/O | Reads the bitwise set status of parameters that must be set before the drive can be enabled |
| DRV.SETUPREQLIST | R/O | Reads the list of parameters that must be set before the drive can be enabled. |
| DRV.STOP | Command | This command stops all drive motion. |
| DRV.TEMPERATURES | R/O | Reads the temperature of drive components. |
| DRV.VER | R/O | Reads the drive version. |
| DRV.VERIMAGE | R/O | Returns the version data from each image. |
| DRV.TEMPERATURES | R/O | Reads the temperature of drive components. |
| DRV.WARNINGS | R/O | Reads the active warnings. |
| DRV.ZERO | R/W | Sets the zero mode. The procedure is activated when the drive is enabled. |
| Feedback 1 (FB1) | | |
| FB1.ENCRES | NV | Sets the resolution of the motor encoder. |
| FB1.HALLSTATE | R/O | Reads the Hall switch values (encoder feedback |
| FB1.HALLSTATEU | R/O | Reads the state of Hall switch U. |
| FB1.HALLSTATEV | R/O | Reads the state of Hall switch V. |
| FB1.HALLSTATEW | R/O | Reads the state of Hall switch W. |

| Parameter or Command | Type | Description |
|--------------------------------|------|--|
| FB1.IDENTIFIED | R/O | Reads the type of feedback device used by the drive/motor. |
| FB1.INITSIGNED | NV | Sets initial feedback value as signed or unsigned. |
| FB1.LDLL | R/O | Reads the motor line-to-line inductance from the FPGA. |
| FB1.LQLL | R/O | Reads the motor line-to-line inductance in the back EMF axis (q axis) from the SFD memory from the FPGA. |
| FB1.MECHPOS | R/O | Reads the mechanical position. |
| FB1.MEMDUMP | W/O | Dumps the memory ID data values of a feedback with memory |
| FB1.MEMVER | R/O | Returns the memory feedback version. |
| FB1.OFFSET | NV | Sets position feedback offset. |
| FB1.POLES | R/O | Reads the number of feedback poles. |
| FB1.OFFSET | NV | Sets position feedback offset. |
| FB1.POLES | R/O | Reads the number of feedback poles. |
| FB1.SELECT | NV | Sets user entered type or identified type (-1). |
| Feedback 2 (FB2) | | |
| FB2.ENCRES | NV | Sets the secondary feedback (FB2) resolution. |
| FB2.MODE | R/W | Sets the mode for the second feedback inputs, EEO connector (X9) and high speed opto inputs (pins 9 and 10 on X7). |
| FB2.SOURCE | R/W | Sets the source for the second feedback input. Choices are the EEO connectors (X9) which are RS485 inputs, or the X7 connector's high speed opto inputs (pins 9 and 10). |
| Fieldbus (FBUS) | | |
| FBUS.PARAM1 TO FBUS.PARAM20 | NV | Set fieldbus specific meanings. |
| FBUS.PLLSTATE | R/O | Returns the status of the PLL |
| FBUS.PLLTHRESH | NV | Sets number of successful synchronized cycles needed to lock the PLL. |
| FBUS.SAMPLEPERIOD | NV | Sets fieldbus sample period. |
| FBUS.SYNCACT | R/O | Reads actual distance from the desired sync distance. |
| FBUS.SYNCDIST | NV | Sets time target for synchronization. |
| FBUS.SYNCWND | NV | Sets symmetrically arranged window around the desired sync distance. |
| FBUS.TYPE | R/O | Shows the active fieldbus type. |
| Gearing (GEAR) | | |
| GEAR.ACCMAX | R/W | Sets the maximum allowed acceleration value; active in opmode 2 (position) only. |
| GEAR.DECMAX | R/W | Sets the maximum allowed deceleration value; active in opmode 2 (position) only. |
| GEAR.IN | R/W | Sets the denominator of the electronic gearing ratio; active in opmode 2 (position) only. |
| GEAR.MODE | R/W | Selects electronic gearing mode; active in opmode 2 (position) only. |

| Parameter or Command | Type | Description |
|-------------------------------------|---------|--|
| GEAR.MOVE | Command | Starts the electronic gearing; active in opmode 2 (position) only. |
| GEAR.OUT | R/W | Sets the numerator of the electronic gearing ratio; active in opmode 2 (position) only. |
| GEAR.VMAX | R/W | Reads the maximum allowed velocity value; active in opmode 2 (position) only. |
| Homing (HOME) | | |
| HOME.ACC | R/W | Sets homing acceleration; active in opmode 2 (position) only. |
| HOME.AUTOMOVE | R/W | Sets homing auto move flag. |
| HOME.DEC | R/W | Sets homing deceleration; active in opmode 2 (position) only. |
| HOME.DIR | NV | Sets homing direction; active in opmode 2 (position) only. |
| HOME.DIST | R/W | Sets homing distance; active in opmode 2 (position) only. |
| HOME.FEEDRATE | R/W | Sets homing velocity factor; active in opmode 2 (position) only. |
| HOME.IPEAKHOME.IPEAK | R/W | Sets the current limit during homing procedure to a mechanical stop; active in opmode 2 (position) only. |
| HOME.MODEHOME.MODE | R/W | Selects the homing mode; active in opmode 2 (position) only. |
| HOME.MOVEHOME.MOVE | Command | Starts a homing procedure; active in opmode 2 (position) only. |
| HOME.PHOME.P | R/W | Sets home position; active in opmode 2 (position) only. |
| HOME.PERRTHRESH HOME.PERRTHRESH | R/W | Sets the position lag threshold; active in opmode 2 (position) only. |
| HOME.REQUIRE | NV | Defines if the axis must be homed before a motion task can be executed. |
| HOME.SETHOME.SET | Command | Immediately sets the home position; active in opmode 2 (position) only. |
| HOME.VHOME.V | R/W | Sets homing velocity; active in opmode 2 (position) only. |
| Hardware Limit Switch (HWLS) | | |
| HWLS.NEGSTATE | R/O | Reads the status of the negative hardware limit switch. |
| HWLS.POSSTATE | R/O | Reads the status of the positive hardware limit switch. |
| Current Loop (IL) | | |
| IL.BUSFF | R/O | Displays the current feedforward value injected by the fieldbus. |
| IL.CMD | R/O | Reads the value of the q-component current controller inside the FPGA. |
| IL.CMDU | R/W | Sets the user current command. |
| IL.DCMD (Password Protected) | R/O | Reads the value of the d-component current controller inside the FPGA. |
| IL.DCMD (Password Protected) | R/O | Reads the value of the d-component current controller inside the FPGA. |
| IL.DCMDU (Password Protected) | R/W | Sets user d-component current command. |

| Parameter or Command | Type | Description |
|----------------------------------|------|---|
| IL.DEADBAND (Password Protected) | R/O | Reads the deadband of two IGBTs in series connection. |
| IL.FB | R/O | Reads the actual value of the d-component current. |
| IL.DFOLDD (Password Protected) | R/O | Reads the motor foldback maximum time at motor peak current. |
| IL.DFOLDR (Password Protected) | R/O | Reads the motor foldback recovery time. |
| IL.DFOLDT | R/O | Reads the motor foldback time constant of the exponential current drop (foldback). |
| IL.DIFOLD | R/O | Reads the drive foldback current limit. |
| IL.DLIMITN | R/W | Sets the negative user (application-specific) d-component current limit. |
| IL.DLIMITP | R/W | Sets the positive user (application-specific) d-component current limit. |
| IL.FB | R/O | Reads the actual value of the d-component current. |
| IL.FOLDFTHRESH | NV | Reads the foldback fault level. |
| IL.FOLDWTHRESH | NV | Sets the foldback warning level. |
| IL.IFOLD | R/O | Reads the overall foldback current limit. |
| IL.INTEN (Password Protected) | NV | Enables/disables the integrator part of the PI loop. |
| IL.IUOFFSET (Password Protected) | R/W | Sets the offset added to the sigma-delta measured current value in the u-winding. |
| IL.IUFB | R/O | Reads the sigma-delta measured current in the u-winding of the motor. |
| IL.IVFB | R/W | Sets the sigma-delta measured current in the u-winding of the motor. |
| IL.IVOFFSET (Password Protected) | R/W | Sets the offset added to the sigma-delta measured current value in the v-winding. |
| IL.KP | NV | Sets the proportional gain of the q-component of the PI regulator. |
| IL.KPDRATIO | NV | Sets the proportional gain of the d-component current PI-regulator as a percentage of IL.KP |
| IL.KPLOOKUPINDEX | R/W | Sets the index into the Current Loop Gain Scheduling Table. |
| IL.KPLOOKUPVALUE | R/W | Sets the value of the current loop gain scheduling index. |
| IL.KPLOOKUPVALUES | R/W | Gets the Current Loop Gain Scheduling Table. |
| IL.KVFF | R/W | Current loop velocity feed-forward gain. |
| IL.LIMITN | NV | Sets the negative user (application-specific) current limit. |
| IL.LIMITP | NV | Sets the positive user (application-specific) current limit. |
| IL.MFOLDD | NV | Sets the motor foldback maximum time at motor peak current. |
| IL.MFOLDR | R/O | Sets the motor foldback recovery time. |
| IL.MFOLDT | NV | Sets the motor foldback time constant of the exponential current drop (foldback). |
| IL.MIFOLD | R/O | Sets the motor foldback current limit. |

| Parameter or Command | Type | Description |
|---------------------------------|------|--|
| IL.PWMFREQ (Password Protected) | NV | Reads and sets the PWM frequency of the IGBTs. |
| IL.VDCMD | R/O | Sets the output of the d-component PI-regulator. |
| IL.VCMD | R/O | Sets the output of the q-component PI regulator. |
| IL.VUFB | R/O | Reads the measured voltage on the u-winding of the motor. |
| IL.VVFB | R/O | Reads the measured voltage on the v-winding of the motor. |
| IL.DLIMITN | R/W | Sets the negative user (application-specific) d-component current limit. |
| IL.DLIMITP | R/W | Sets the positive user (application-specific) d-component current limit. |
| | | |
| IL.FB | R/O | Reads the actual value of the d-component current. |
| LOAD.INERTIA | NV | Sets the load inertia. |

| Motor (MOTOR) Parameters | | |
|--------------------------|---------|---|
| MOTOR.AUTOSET | NV | Determines which drive parameters are calculated automatically. |
| MOTOR.BRAKE | NV | Sets the presence or absence of a motor brake. |
| | | |
| MOTOR.ICONT | NV | Sets the motor continuous current. |
| MOTOR.INERTIA | NV | Sets the motor inertia. |
| MOTOR.IPEAK | NV | Sets the motor peak current. |
| MOTOR.KE | | Sets the motor back EMF constant. |
| MOTOR.KT | NV | Sets the torque constant of the motor. |
| MOTOR.LQLL | NV | Sets the line-to-line motor Lq. |
| MOTOR.VMAX | NV | Sets the maximum motor speed. |
| MOTOR.NAME | NV | Sets the motor name. |
| MOTOR.PHASE | NV | Sets the motor phase. |
| MOTOR.PITCH | NV | Sets the motor pitch. |
| MOTOR.POLES | NV | Sets the number of motor poles. |
| MOTOR.R | NV | Sets the stator winding resistance phase-phase in ohms. |
| MOTOR.TBRAKEAPP | NV | The delay time used for applying the motor brake. |
| MOTOR.TBRAKERLS | NV | The delay time used for releasing the motor brake. |
| MOTOR.TEMP | R/O | Reads the motor temperature represented as the resistance of the motor PTC. |
| MOTOR.TEMPFAULT | NV | Sets the motor temperature fault level. |
| MOTOR.TEMPWARN | NV | Sets the motor temperature warning level. |
| MOTOR.TYPE | NV | Sets the motor type. |
| MOTOR.VOLTMAX | NV | Sets the motor maximum voltage. |
| Motion Task (MT) | | |
| MT.ACC | R/W | Specifies motion task acceleration; active in opmode 2 (position) only. |
| MT.CLEAR | Command | Clears motion tasks from the drive; active in opmode 2 (position) only. |

| | | |
|---------------------------|---------|--|
| MT.CNTL | R/W | Sets motion task control word; active in opmode 2 (position) only. |
| MT.CONTINUE | Command | Continues a stopped motion task; active in opmode 2 (position) only. |
| MT.DEC | R/W | Sets motion task deceleration; active in opmode 2 (position) only. |
| MT.EMERGMT | R/W | Selects a motion task to be triggered after an emergency stop procedure; active in opmode 2 (position) only. |
| MT.HOMEREQUIRE | NV | Defines if the axis is required to be homed before a motion task can be executed. |
| MT.LIST | Command | Lists all initialized motion tasks in the drive; active in opmode 2 (position) only. |
| MT.MTNEXT | R/W | Specifies following motion task number; active in opmode 2 (position) only. |
| MT.TNEXT | R/W | Specifies following motion task time; active in opmode 2 (position) only. |
| MT.LIST | Command | Lists all initialized motion tasks in the drive; active in opmode 2 (position) only. |
| MT.LOAD | Command | Reads/loads a motion task number from the drive; active in opmode 2 (position) only. |
| MT.MOVE | Command | Starts a motion task; active in opmode 2 (position) only. |
| MT.NUM | R/W | Sets the motion task number; active in opmode 2 (position) only. |
| MT.PARAMS | Command | Shows a motion task; active in opmode 2 (position) only. |
| MT.P | R/W | Sets the motion task position; active in opmode 2 (position) only. |
| MT.SET | Command | Sets the motion task in the drive; active in opmode 2 (position) only. |
| MT.TNUM | R/W | Motion task customer table number. |
| MT.V | R/W | Sets the motion task velocity; active in opmode 2 (position) only. |
| Position Loop (PL) | | |
| PL.CMD | NV | Reads the position command directly from the entry to the position loop. |
| PL.ERR | NV | Reads the position error present when the drive is controlling the position loop. |
| PL.ERRFTHRESH | NV | Sets the maximum position error. |
| PL.ERRWTHRESH | NV | Sets the position error warning level. |
| PL.FB | R/O | Reads the position feedback value. |
| PL.FBSOURCE | NV | Sets the feedback source for the position loop. |
| PL.INTINMAX | NV | Limits the input of the position loop integrator by setting the input saturation. |
| PL.INTOUTMAX | NV | Limits the output of the position loop integrator by setting the output saturation. |
| PL.KI | NV | Sets the integral gain of the position loop. |
| PL.KP | NV | Sets the proportional gain of the position regulator PID loop. |

| Programmable Limit Switch (PLS) | | |
|--|---------|---|
| PLS.EN | R/W | Enables programmable limit switch (PLS). |
| PLS.MODE | NV | Selects programmable limit switch mode. |
| PLS.P1 TO PLS.P8 | NV | Sets the trigger point for programmable limit switches. |
| PLS.STATE | R/O | Reads the programmable limit switch state. |
| Recorder (REC) | | |
| REC.ACTIVE | R/O | Indicates if data recording is in progress (active). |
| REC.CH1 to REC.CH6 | R/W | Sets recording channels 1 to 6. |
| REC.DONE | R/O | Checks whether or not the recorder has finished recording. |
| REC.GAP | R/W | Specifies the gap between consecutive samples. |
| REC.NUMPOINTS | R/W | Sets the number of points to record. |
| REC.OFF | R/W | Turns the recorder OFF. |
| REC.RETRIEVE | R/O | Transfers all the recorded data to the communication channel. |
| REC.STOPTYPE | R/W | Sets the recorder stop type. |
| REC.TRIG | Command | Triggers the recorder. |
| REC.TRIGSLOPE | R/W | Sets the trigger slope. |
| REC.TRIGTYPE | R/W | Sets the trigger type. |
| REC.TRIGVAL | R/W | Sets the trigger value. |
| Regen Resistor (REGEN) | | |
| REGEN.POWER | R/O | READS REGEN RESISTOR'S CALCULATED POWER. |
| REGEN.REXT | N/V | SETS THE EXTERNAL, USER-DEFINED REGEN RESISTOR RESISTANCE. |
| REGEN.TEXT | R/W | Sets the external regen resistor thermal protection time constant. |
| REGEN.TYPE | N/V | SETS THE REGEN RESISTOR TYPE. |
| REGEN.WATTEXT | R/W | SETS THE REGEN RESISTOR'S POWER FAULT LEVEL FOR AN EXTERNAL REGEN RESISTOR. |
| Service Motion (SM) | | |
| SM.I1 | R/W | Sets service motion current 1; active in opmode 0 (torque) only. |
| SM.I2 | R/W | Sets service motion current 2; active in opmode 0 (torque) only. |
| SM.MODE | R/W | Sets the service motion mode. |
| SM.MOVE | Command | Starts the service motion. |
| SM.T1 | R/W | Sets the service motion time 1. |
| SM.T2 | R/W | Sets the service motion time 2. |
| SM.V1 | R/W | Sets service motion velocity 1; active in opmode 1 (velocity) and 2 (position). |
| SM.V2 | R/W | Sets service motion velocity 2; active in opmode 1 (velocity) and 2 (position). |
| Units (UNIT) | | |
| UNIT.ACCLINEAR | NV | Sets the linear acceleration/deceleration units. |
| UNIT.ACCROTARY | NV | Sets the rotary acceleration/deceleration units. |

| | | |
|---------------------------|-----|---|
| UNIT.LABEL | NV | Sets user-defined name for user-defined position units. |
| UNIT.PIN | NV | Sets gear IN for the unit conversion. |
| UNIT.PLINEAR | NV | Sets the linear position units. |
| UNIT.POUT | NV | Sets gear out for the unit conversion. |
| UNIT.PROTARY | NV | Sets the position units when the motor type (MOTOR.TYPE) is rotary. |
| UNIT.VLINEAR | NV | Sets the linear velocity units. |
| UNIT.VROTARY | NV | Sets the velocity units when the motor type (MOTOR.TYPE) is rotary. |
| Bus voltage (VBUS) | | |
| VBUS.OVFTHRESH | R/O | Reads the over voltage fault level. |
| VBUS.OVWTHRESH | N/V | Sets voltage level for over voltage warning. |
| VBUS.RMSLIMIT | R/O | Reads the limit for the bus capacitors load. |
| VBUS.UVFTHRESH | R/O | Sets the under voltage fault level. |
| VBUS.UVMODE | | Indicates undervoltage (UV) mode. |
| VBUS.UVWTHRESH | NV | Sets voltage level for undervoltage warning. |
| VBUS.VALUE | R/O | Reads DC bus voltage. |
| Velocity Loop (VL) | | |
| VL.ARPF1 TO VL.ARPF4 | R/W | Sets the natural frequency of the pole (denominator) of anti-resonance (AR) filters 1, 2, 3, and 4; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ARPQ1 TO VL.ARPQ4 | R/W | Sets the Q of the pole (denominator) of anti-resonance (AR) filter 1; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ARTYPE1 TO VL.ARTYPE4 | NV | Indicates the method used to calculate BiQuad coefficients; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ARZF1 TO VL.ARZF4 | R/W | Sets the natural frequency of the zero (numerator) of anti-resonance (AR) filter 1; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ARZQ1 TO VL.ARZQ4 | R/W | Sets the Q of the zero (numerator) of anti-resonance filter #1; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.BUSFF | R/O | Displays the velocity loop feedforward value injected by the field-bus; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.CMD | R/O | Reads the actual velocity command; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.CMDU | R/W | Sets the user velocity command; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.ERR | R/O | Sets the velocity error; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.FB | R/O | Reads the velocity feedback; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.FBFILTER | R/O | Filters VL.FB value; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.FBSOURCE | NV | Sets feedback source for the velocity loop; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.FBUNFILTERED | R/O | Reads the velocity feedback. |

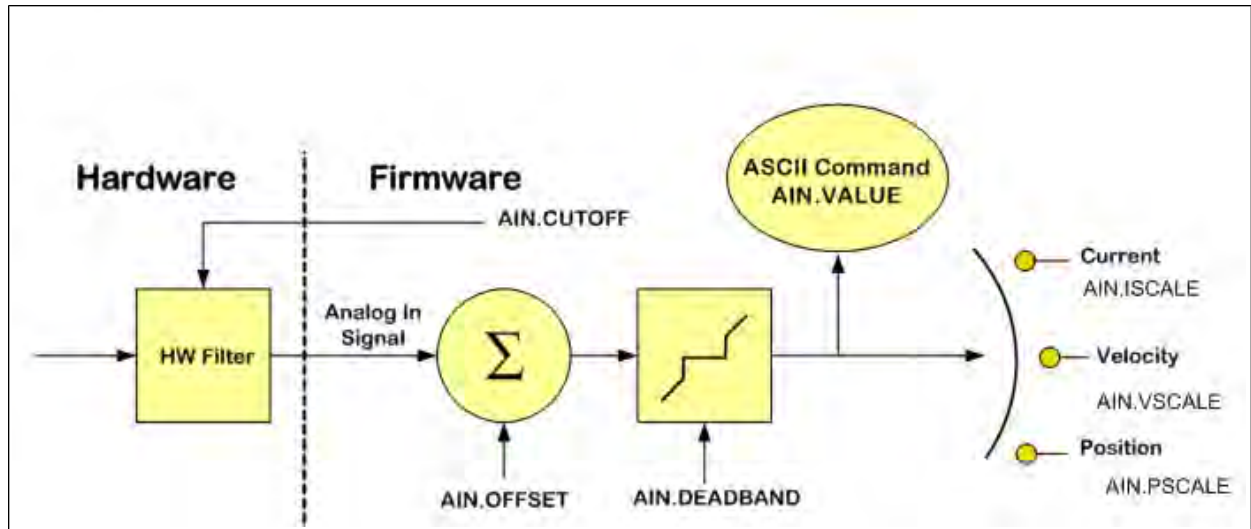
| | | |
|----------------------------|---------|---|
| VL.FF | R/O | Displays the velocity loop overall feedforward value; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.GENMODE | NV | Selects mode of velocity generation (Observer, d/dt); active in opmodes 1 (velocity) and 2 (position) only. |
| VL.KBUSFF | R/W | Sets the velocity loop acceleration feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.KI | NV | Sets the velocity loop integral gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.KP | NV | Sets velocity loop proportional gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.KVFF | R/W | Sets the velocity loop velocity feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.LIMITN | NV | Sets the velocity lower limit; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.LIMITP | NV | Sets the velocity high limit; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.LMJR | R/W | Sets the ratio of the estimated load moment of inertia relative to the motor moment of inertia; active in opmodes 1 (velocity) and 2 (position) only. |
| VL.THRESH | NV | Sets the over speed fault value; active in opmodes 1 (velocity) and 2 (position) only. |
| Wake and Shake (WS) | | |
| WS.ARM | Command | Sets wake and shake to start at the next drive enable. |
| WS.DISTMIN | R/W | Sets the minimum movement required for wake and shake. |
| WS.IMAX | R/W | Sets maximum current used for wake and shake. |
| WS.MODE | R/W | Sets the method used for wake and shake. |
| WS.DISTMAX | R/W | Sets maximum movement allowed for wake and shake. |
| WS.NUMLOOPS | R/W | Sets the number of repetitions for wake and shake. |
| WS.STATE | R/O | Reads wake and shake status |
| WS.T | R/W | Sets wake and shake current-vector appliance time |
| WS.TDELAY1 | NV | Delay for wake and shake timing |
| WS.TDELAY2 | NV | Sets the delay for wake and shake timing. |
| WS.TDELAY3 | NV | Sets the delay for wake and shake between loops in mode 0. |
| WS.VTHRESH | NV | Defines the maximum allowed velocity for Wake & Shake |

This page intentionally left blank.

AIN Parameters

This section describes the analog input (AIN) parameters. AIN parameters function as shown in the block diagram below:

Analog Input Block Diagram



AIN.CUTOFF

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the analog input low-pass filter cutoff frequency. |
| Units | Hz |
| Range | 0 to 10,000 Hz |
| Default Value | 5,000 Hz |
| Data Type | Float |
| See Also | Analog Input Block Diagram |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 0 | M_01-03-00-000 |

Description

AIN.CUTOFF sets the break frequency in Hz for two cascaded single-pole low-pass filters on the hardware command input. Since the two poles are cascaded at the same frequency, the -3 dB frequency is $0.64 \cdot \text{AIN.CUTOFF}$ in hertz and the 10% to 90% step response rise time is $0.53/\text{AIN.CUTOFF}$ in seconds.

Suggested operating values are as follows:

- Analog torque opmode: 5 kHz
- Analog velocity opmode: 2.5 kHz
- General purpose analog input high resolution: 500 Hz

Related Topics

10.4 Analog Input

Velocity Controller Environment Block Diagram (for the drive controller environment).

AIN.DEADBAND

| General Information | |
|---------------------|--|
| Type | NV Parameter |
| Description | Sets the analog input signal deadband. |
| Units | V |
| Range | 0 to 12.5 V |
| Default Value | 0 V |
| Data Type | Float |
| See Also | Analog Input Block Diagram |
| Start Version | M_01-00-00-000 |

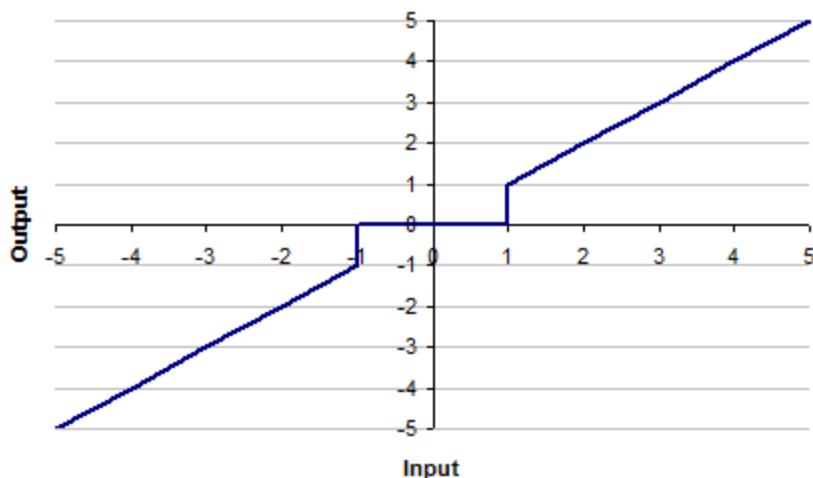
| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 2 | M_01-03-00-000 |

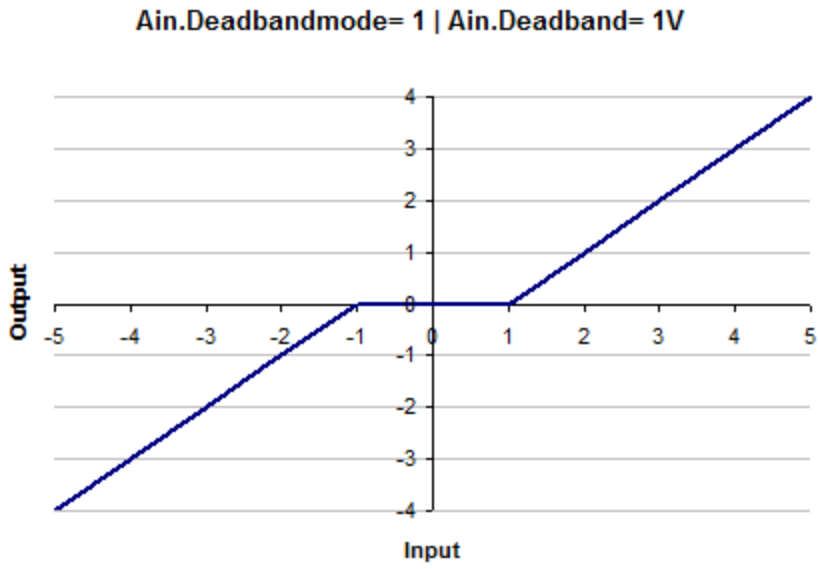
Description

AIN.DEADBAND sets the deadband of the analog input signal. When AIN.DEADBANDMODE is set to 0, and the value of the analog input is less than the value of AIN.DEADBAND, the analog command will be 0. When the analog input is greater or equal to the AIN.DEADBAND, then the analog command will be generated using the scaling specified.

When AIN.DEADBANDMODE is set to 1, the analog command is 0 if the input is less than the deadband value. When the input is greater than the deadband, the output is equal to $(\text{Input} - \text{Deadband}) * \text{Scaling}$. Below are illustrations of this behavior.

Ain.Deadbandmode= 0 | Ain.Deadband= 1V





Related Topics

10.4 Analog Input

Velocity Controller Environment Block Diagram (for the drive controller environment).

AIN.DEADBANDMODE

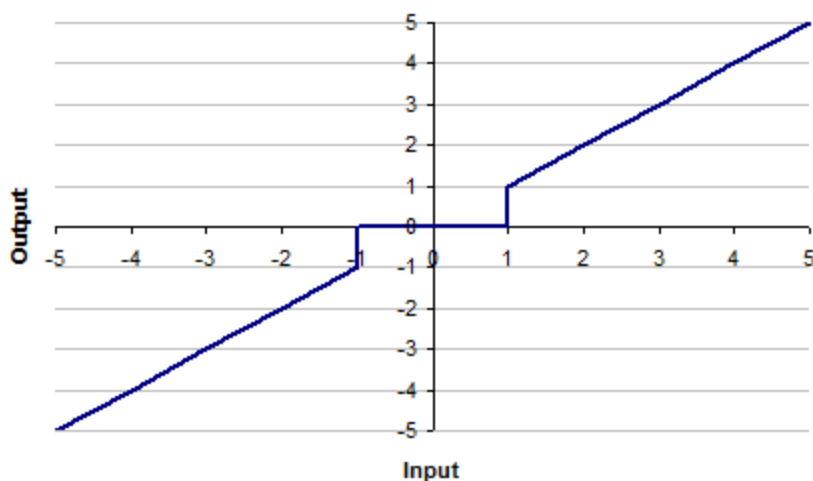
| General Information | |
|---------------------|--------------------------------------|
| Type | NV Parameter |
| Description | Sets the analog input deadband mode. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | AIN.DEADBAND |
| Start Version | M_01-03-06-000 |

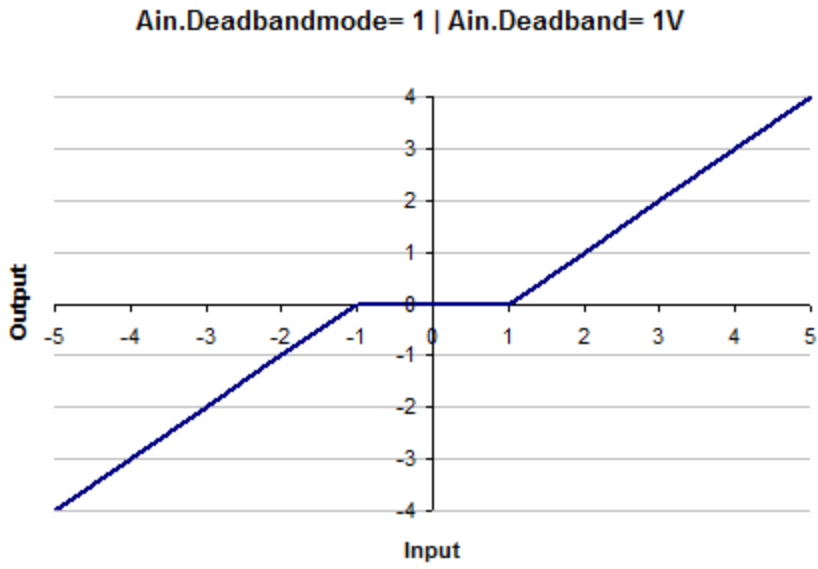
Description

When AIN.DEADBANDMODE is set to 0, and the value of the analog input is less than the value of AIN.DEADBAND, the analog command will be 0. When the analog input is greater or equal to the AIN.DEADBAND, then the analog command will be generated using the scaling specified.

When AIN.DEADBANDMODE is set to 1, the analog command is 0 if the input is less than the deadband value. When the input is greater than the deadband, the output is equal to $(\text{Input} - \text{Deadband}) * \text{Scaling}$. Below are illustrations of this behavior.

Ain.Deadbandmode= 0 | Ain.Deadband= 1V





Related Topics

10.4 Analog Input

Velocity Controller Environment Block Diagram (for the drive controller environment).

AIN.ISCALE

| General Information | |
|----------------------|---------------------------------------|
| Type | NV Parameter |
| Description | Sets the analog current scale factor. |
| Units | A/V |
| Range | 0.001 to 22.4 A/V |
| Default Value | 0.001 A/V |
| Data Type | Float |
| See Also | Analog Input Block Diagram |
| Start Version | M_01-01-01-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 4 | M_01-03-00-000 |

Description

AIN.ISCALE sets the analog current scale factor that scales the analog input (AIN.VALUE) for DRV.OPMODE = 1 (analog torque mode).

The value entered is the motor current per 10 V of analog input. This value may be either higher or lower than 100%, but the actual analog input will be limited by the application current limit (IL.LIMITN and IL.LIMITP).

Related Topics

10.4 Analog Input

Velocity Controller Environment Block Diagram (for the drive controller environment).

AIN.OFFSET

| General Information | |
|----------------------|--------------------------------------|
| Type | NV Parameter |
| Description | Sets the analog input offset. |
| Units | V |
| Range | -10 to +10 V |
| Default Value | 0 V |
| Data Type | Float |
| See Also | Analog Input Block Diagram, AIN.ZERO |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 6 | M_01-03-00-000 |

Description

AIN.OFFSET sets the analog offset, which is added to the analog input command to the drive. This value compensates for the analog input signal (AIN.VALUE) offset or drift.

Related Topics

10.4 Analog Input

Velocity Controller Environment Block Diagram (for the drive controller environment).

AIN.PSCALE

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the analog position scale factor. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts/V, rad/V, deg/V, (custom units)/V, 16-bit counts/V Linear: counts/V, mm/V, $\mu\text{m}/\text{V}$, (custom units)/V, 16-bit counts/V |
| Range | Rotary: 1 to 9,223,372,036,854,775 counts/V 0 to 13,493,026.816 rad/V 0.06 to 179.0 deg/V 0 to 10,737,418.240 (PIN/POUT)/V 0 to 140,737,488,355.327 16-bit counts/V Linear: 1 to 9,223,372,036,854,775 counts/V 0 to 2,147,483.648 mm/V 0 to 2,147,483,648.000 $\mu\text{m}/\text{V}$ 0 to 10,737,418.240 (PIN/POUT)/V 0 to 140,737,488,355.327 16-bit counts/V |
| Default Value | Rotary: 1 counts/V 0 rad/V 0 deg/V 0 (PIN/POUT)/V 0 16-bit counts/V Linear: 1 count/V 0 rad/V 0 deg/V 0 (PIN/POUT)/V 0 16-bit counts/V |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-01-01-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3472h/0 | M_01-00-00-000 |
| Modbus | 8 (64-bit) | M_01-03-00-000 |

Description

AIN.PSCALE is an analog position scale factor that scales the analog input (AIN.VALUE) for DRV.OPMODE = 2, DRV.CMDSOURCE = 3 (analog position mode).

Related Topics

10.4 Analog Input

Velocity Controller Environment Block Diagram (for the drive controller environment).

AIN.VALUE

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the value of the analog input signal. |
| Units | V |
| Range | -12.5 to +12.5 V |
| Default Value | N/A |
| Data Type | Float |
| See Also | AIN.OFFSET, AIN.ZERO, Analog Input Block Diagram |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3470h/4 | M_01-00-00-000 |
| | 3509h | |
| Modbus | 12 | M_01-03-00-000 |

Description

AIN.VALUE reads the analog input value after the value is filtered (as shown in the Analog Input Block Diagram).

Related Topics

10.4 Analog Input

Velocity Controller Environment Block Diagram (for the drive controller environment).

AIN.VSCALE

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets analog velocity scale factor. |
| Units | Depends on UNIT.VROTARY or UNIT.ACCLINEAR Rotary: rpm/V, rps/V, (deg/s)/V, [(custom units)/s]/V, (rad/s)/V Linear: counts/s/V, (mm/s)/V, (μm/s)/V, [(custom units)/s]/V |
| Range | Rotary: 0.060 to 60,000 rpm/V 0.001 to 1,000 rps/V 0.359 to 360,000 (deg/s)/V 0.005 to 5,000 [(custom units)/s]/V 0.006 to 6,283.186 (rad/s)/V Linear: 0.001 to 1.000 counts/s/V 0.001*MOTOR.PITCH to 1,000.000*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH to 1,000,000.000*MOTOR.PITCH (μm/s)/V 0.005 to 5,000 [(custom units)/s]/V |
| Default Value | Rotary: 0.060 rpm/V 0.001 rps/V 0.359 (deg/s)/V 0.005 [(custom units)/s]/V 0.006 (rad/s)/V Linear: 0.001 counts/s/V 0.001*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH (μm/s)/V 0.005 to 5,000 [(custom units)/s]/V |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3629h/0 | M_01-00-00-000 |
| Modbus | 14 | M_01-03-00-000 |

Description

AIN.VSCALE is an analog velocity scale factor that scales the analog input AIN.VALUE) for DRV.OPMODE = 1 (analog velocity mode).

The value entered is the motor velocity per 10 V of analog input. This value may be either higher or lower than the application velocity limit (VL.LIMITP or VL.LIMITN), but the actual analog I/O will be limited by VL.LIMITP or VL.LIMITN.

Related Topics

10.4 Analog Input

Velocity Controller Environment Block Diagram (for the drive controller environment).

AIN.ZERO

| General Information | |
|---------------------|---------------------------------|
| Type | Command |
| Description | Zeroes the analog input signal. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | AIN.VALUE, AIN.OFFSET |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | N/A | M_01-00-00-000 |
| Modbus | 16 | M_01-03-00-000 |

Description

AIN.ZERO causes the drive to zero the analog input signal (AIN.VALUE). You may need to execute this command more than once to achieve zero offset, and AIN.OFFSET is modified in this process.

Related Topics

10.4 Analog Input

Velocity Controller Environment Block Diagram (for the drive controller environment).

AIO Parameters

AIO.ISCALE

| General Information | |
|---------------------|---------------------------------------|
| Type | NV Parameter |
| Description | Sets the analog current scale factor. |
| Units | A/V |
| Range | 0.001 to 22.4 A/V |
| Default Value | 0.001 A/V |
| Data Type | Float |
| See Also | Analog Input Block Diagram |
| Start Version | M_01-00-00-000 |
| End Version | M_01-01-01-000 |

Description

AIO.ISCALE sets the analog current scale factor that scales the following:

- The analog input (AIN.VALUE) for DRV.OPMODE = 0 (analog torque mode), DRV.CMDSOURCE = 3 (analog).
- The analog output (AOUT.VALUE) for AOUT.MODE = 5 or 6. The value entered is the motor current per 1 V of analog input or output. This value may be either higher or lower than 100%, but the actual analog I/O will be limited by the application current limit (IL.LIMITN and IL.LIMITP).

AIO.PSCALE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets position scale factor. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts/V, rad/V, deg/V, (custom units)/V, 16-bit counts/V Linear: counts/V, mm/V, um/V, (custom units)/V, 16-bit counts/V |
| Range | Rotary: 1 to 9,223,372,036,854,775 counts/V 0 to 13,493,026.816 rad/V 0 to 773,094,113.280 deg/V 0 to 10,737,418.240 (custom units)/V 0 to 140,737,488,355.327 16-bit counts/V Linear: 1 to 9,223,372,036,854,775 counts/V 0 to 2147483.648 mm/V 0 to 2147483648.000 um/V 0 to 10737418.240 (custom units)/V 0 to 140737488355.327 16-bit counts/V |
| Default Value | Rotary: 1 counts/V 0 rad/V 0 deg/V 0 (custom units)/V 0 16-bit counts/V Linear: 1 count/V 0 rad/V 0 deg/V 0 (custom units)/V 0 counts16 bit/V |
| Data Type | Float |
| See Also | Analog Input Block Diagram |
| Start Version | M_01-00-00-000 |
| End Version | M_01-01-01-000 |

Description

AIO.PSCALE is an analog position scale factor that scales:

1. The analog input (AIN.VALUE) for DRV.OPMODE = 2 , DRV.CMDSOURCE = 3 (analog position mode)
2. The analog output (AOUT.VALUE) for AOUT.MODE = 6, or 7. (actual position or position error) per 10 V of analog input or output.

AIO.VSCALE

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets velocity scale factor. |
| Units | Depends on UNIT.VROTARY or UNIT.ACCLINEAR Rotary: rpm/V, rps/V, (deg/s)/V, [(custom units)/s]/V, (rad/s)/V Linear: counts/s/V, (mm/s)/V, (um/s)/V, [(custom units)/s]/V |
| Range | Rotary: 0.060 to 60,000 rpm/V 0.001 to 1,000 rps/V 0.359 to 360,000 (deg/s)/V 0.005 to 5,000 [(custom units)/s]/V 0.006 to 6,283.186 (rad/s)/V Linear: 0.001 to 1.000 counts/s/V 0.001*MOTOR.PITCH to 1,000.000*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH to 1,000,000.000*MOTOR.PITCH (um/s)/V 0.005 to 5,000 [(custom units)/s]/V |
| Default Value | Rotary: 0.060 rpm/V 0.001 rps/V 0.359 (deg/s)/V 0.005 [(custom units)/s]/V 0.006 (rad/s)/V Linear: 0.001 counts/s/V 0.001*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH (um/s)/V 0.005 to 5,000 [(custom units)/s]/V |
| Data Type | Float |
| See Also | Analog Input Block Diagram |
| Start Version | M_01-00-00-000 |
| End Version | M_01-01-01-000 |

Description

AIO.VSCALE is an analog velocity scale factor that scales:

1. The analog input (AIN.VALUE) for DRV.OPMODE = 2 (analog velocity mode)
2. The analog output (AOUT.VALUE) for AOUT.MODE = 1, 3, or 7. The value entered is the motor velocity per 10 V of analog input or output. This value may be either higher or lower than the application velocity limit (VL.LIMITP or VL.LIMITN), but the actual analog I/O will be limited by VL.LIMITP or VL.LIMITN.

This page intentionally left blank.

AOUT Parameters

AOUT.DEBUGADDR

| General Information | |
|---------------------|-----------------------------------|
| Type | NV Parameter |
| Description | Sets the memory address to debug. |
| Units | N/A |
| Range | 4 to 4,292,870,142 |
| Default Value | 4 |
| Data Type | Integer |
| See Also | AOUT.MODE |
| Start Version | M_01-01-01-000 |

Description

AOUT.DEBUGADDR sets the memory address to debug when AOUT.MODE = 9 (debug mode).

Related Topics

10.5 Analog Output

AOUT.DEBUGSCALE

| General Information | |
|---------------------|--------------------------------------|
| Type | NV Parameter |
| Description | Sets the scale to be used for debug. |
| Units | N/A |
| Range | 0.001 to 9,223,372,036,854,775.000 |
| Default Value | 1 |
| Data Type | Float |
| See Also | AOUT.MODE |
| Start Version | M_01-01-01-000 |

Description

AOUT.DEBUGSCALE sets the scale to be used for debug when AOUT.MODE = 9 (debug mode).

Related Topics

10.5 Analog Output

AOUT.ISCALE

| General Information | |
|----------------------|---------------------------------------|
| Type | NV Parameter |
| Description | Sets the analog current scale factor. |
| Units | A/V |
| Range | 0.001 to 22.4 A/V |
| Default Value | 0.001 to 22.4 A/V |
| Data Type | Float |
| See Also | AOUT.VALUE |
| Start Version | M_01-01-01-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 18 | M_01-03-00-000 |

Description

AOUT.ISCALE sets the analog current scale factor that scales the analog output (AOUT.VALUE) for AOUT.MODE = 4 or 5. The value entered is the motor current per 10 V of analog input or output. This value may be either higher or lower than 100%, but the actual analog I/O will be limited by the application current limit (IL.LIMITN and IL.LIMITP).

Related Topics

10.5 Analog Output

AOUT.MODE

| General Information | |
|----------------------|------------------------------|
| Type | NV Parameter |
| Description | Sets the analog output mode. |
| Units | N/A |
| Range | 0 to 11 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3470h/1 | M_01-00-00-000 |
| Modbus | 20 | M_01-03-00-000 |

Description

AOUT.MODE sets the analog output functionality.

| AOUTx.MODE | Description |
|------------|---|
| 0 | User variable. The analog output signal is determined by the user (using AOUT.VALUEU). |
| 1 | Actual velocity. The analog signal describes the current velocity value (VL.FB). |
| 2 | Velocity error. The analog signal describes the velocity error value. |
| 3 | Velocity command. The analog signal describes the velocity command value. |
| 4 | Actual current. The analog signal describes the actual current value. |
| 5 | Current command. The analog signal describes the current command value. |
| 6 | Actual position. The analog signal describes the current position value. |
| 7 | Position error. The analog signal describes the position error value. |
| 8 | Triangle wave. The analog signal is a triangle wave (sawtooth pattern). |
| 9 | Debug mode. In this mode the user can define a drive variable to monitor via the analog output (AOUT.VALUEU). |
| 10 | Unfiltered Velocity (VL.FBUNFILTERED) |
| 11 | Filtered Velocity - 10Hz Lowpass (VL.FBFILTER) |

Example

You can use AOUT.MODE and AOUT.VALUEU to configure an output signal as follows:

```
-->AOUT.MODE 0
-->AOUT.VALUEU 5
-->AOUT.VALUEU 4.33
```

Related Topics

10.5 Analog Output

AOUT.OFFSET

| General Information | |
|----------------------|-------------------------------|
| Type | NV Parameter |
| Description | Sets the analog input offset. |
| Units | V |
| Range | -10 to +10 V |
| Default Value | 0 V |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 22 | M_01-03-00-000 |

Description

This parameter sets the analog input offset.

Related Topics

10.5 Analog Output

AOUT.PSCALE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the analog position scale factor. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts/V, rad/V, deg/V, (custom units)/V, 16-bit counts/V Linear: counts/V, mm/V, $\mu\text{m}/\text{V}$, (custom units)/V, 16-bit counts/V |
| Range | Rotary: 1 to 9,223,372,036,854,775 counts/V 0 to 13,493,026.816 rad/V 0 to 773,094,113.280 deg/V 0 to 10,737,418.240 (custom units)/V 0 to 140,737,488,355.327 16-bit counts/V Linear: 1 to 9,223,372,036,854,775 counts/V 0 to 2,147,483.648 mm/V 0 to 2,147,483,648.000 $\mu\text{m}/\text{V}$ 0 to 10,737,418.240 (custom units)/V 0 to 140,737,488,355.327 16-bit counts/V |
| Default Value | Rotary: 1 counts/V 0 rad/V 0 deg/V 0 (custom units)/V 0 16-bit counts/V Linear: 1 counts/V 0 rad/V 0 deg/V 0 (custom units)/V 0 counts16 bit/V |
| Data Type | Float |
| See Also | AOUT.VALUE |
| Start Version | M_01-01-01-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3471h/0 | M_01-00-00-000 |
| Modbus | 24 (64-bit) | M_01-03-00-000 |

Description

AOUT.PSCALE is an analog position scale factor that scales the analog output (AOUT.VALUE) for AOUT.MODE = 6, or 7 (actual position or position error) per 10 V of analog input or output.

Related Topics

10.5 Analog Output

AOUT.VALUE

| General Information | |
|----------------------|--------------------------------|
| Type | N/V Parameter |
| Description | Reads the analog output value. |
| Units | V |
| Range | -10 to +10 V |
| Default Value | 0 |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3470h/2 | M_01-00-00-000 |
| Modbus | 28 (64-bit) | M_01-03-00-000 |

Description

AOUT.VALUE reads the analog output value. This parameter can also be used to set the value of the analog output when AOUT.MODE = 0 (analog output signal is determined by the user).

Related Topics

10.5 Analog Output

AOUT.VALUEU

| General Information | |
|----------------------|-------------------------------|
| Type | R/W Parameter |
| Description | Sets the analog output value. |
| Units | V |
| Range | -10 to +10 V |
| Default Value | 0 |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3470h/3 | M_01-00-00-000 |
| Modbus | 32 (64-bit) | M_01-03-00-000 |

Description

AOUT.VALUEU reads/writes the analog output value when AOUT.MODE = 0 (analog output signal is determined by the user).

Related Topics

10.5 Analog Output

AOUT.VSCALE

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets velocity scale factor for analog output. |
| Units | Depends on UNIT.VROTARY or UNIT.ACCLINEAR Rotary: rpm/V, rps/V, (deg/s)/V, [(custom units)/s]/V, (rad/s)/V Linear: counts/s/V, (mm/s)/V, (µm/s)/V, [(custom units)/s]/V |
| Range | Rotary: 0.060 to 60,000 rpm/V 0.001 to 1,000 rps/V 0.359 to 360,000 (deg/s)/V 0.005 to 5,000 [(custom units)/s]/V 0.006 to 6,283.186 (rad/s)/V Linear: 0.001 to 1.000 counts/s/V 0.001*MOTOR.PITCH to 1,000.000*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH to 1,000,000.000*MOTOR.PITCH (µm/s)/V 0.005 to 5,000 [(custom units)/s]/V |
| Default Value | Rotary: 0.060 rpm/V 0.001 rps/V 0.359 (deg/s)/V 0.005 [(custom units)/s]/V 0.006 (rad/s)/V Linear: 0.001 counts/s/V 0.001*MOTOR.PITCH (mm/s)/V 0.998*MOTOR.PITCH (µm/s)/V 0.005 [(custom units)/s]/V |
| Data Type | Float |
| See Also | AOUT.VALUE |
| Start Version | M_01-01-01-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3470h/5 | M_01-00-00-000 |
| Modbus | 36 | M_01-03-00-000 |

Description

AOUT.VSCALE is an analog velocity scale factor that scales the analog output (AOUT.VALUE) for AOUT.MODE = 1, 2, or 3. The value entered is the motor velocity per 10 V of analog output. This value may be either higher or lower than the application velocity limit (VL.LIMITP or VL.LIMITN), but the actual analog I/O will be limited by VL.LIMITP or VL.LIMITN.

BODE Parameters

BODE.EXCITEGAP

| General Information | |
|---------------------|---|
| Type | R/W Parameter |
| Description | Controls how often the excitation is updated. |
| Units | Drive samples |
| Range | 1 to 255 drive samples |
| Default Value | 2 drive samples |
| Data Type | N/A |
| See Also | BODE.MODE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 38 | M_01-03-00-000 |

Description

BODE.EXCITEGAP controls how often the excitation is updated. The excitation is updated every n drive samples, where n is BODE.EXCITEGAP. For example, if BODE.EXCITEGAP = 2, then the excitation is updated every $2/(16,000 \text{ Hz}) = 1/8,000 \text{ Hz} = 0.000125 \text{ sec}$. When measuring a system, update the excitation only as often as the data is recorded.

Example

Set excitation update rate to 8,000 Hz:

```
-->BODE.EXCITEGAP 2
```

Set excitation update rate to 4,000 Hz:

```
-->BODE.EXCITEGAP 4
```

Get excitation update rate (already set to 8000 Hz):

```
-->BODE.EXCITEGAP 2
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

15 Scope

1.2.1.5 Bode (set command source)

6.5 Settings

F126

Error: Invalid Bode plot mode for this function. and other errors)

BODE.FREQ

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the frequency of the sine excitation source. |
| Units | Hz |
| Range | 0 to 8,000 Hz |
| Default Value | 0 Hz |
| Data Type | Float |
| See Also | BODE.MODE BODE.INJECTPOINT, BODE.IAMP, BODE.VAMP |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 40 | M_01-03-00-000 |

Description

BODE.FREQ sets the frequency of the sine excitation source in Hz. The sine excitation source is used to take frequency response measurements of a system.

Example

Setting up a sine excitation source of 0.2 A at 50 Hz:

```
-->BODE.INJECTPOINT 1
-->BODE.IAMP 0.2
-->BODE.FREQ 50.0
-->BODE.MODE 2
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

15 Scope

1.2.1.5 Bode (set command source)

6.5 Settings

F126

Error: Invalid Bode plot mode for this function. and others)

BODE.IAMP

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets current command value used during the Bode procedure. |
| Units | A |
| Range | +/- Combined drive and motor current limit |
| Default Value | 0.2 A |
| Data Type | Float |
| See Also | BODE.INJECTPOINT, BODE.FREQ |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 42 | M_01-03-00-000 |

Description

BODE.IAMP sets the amplitude of the excitation when in current mode as set in BODE.INJECTPOINT. When using BODE.MODE = 1 and BODE.INJECTPOINT = 1, this parameter will determine the level of noise injected to commanded current value.

Example

Set the excitation current to 0.2 A:

```
-->BODE.IAMP 0.2
```

Get the excitation current (already set to 0.2 A):

```
-->BODE.IAMP 0.200 [A]
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

15 Scope

1.2.1.5 Bode (set command source)

6.5 Settings

F126

Error: Invalid Bode plot mode for this function. and others)

BODE.IFLIMIT

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the current fault duration limit in seconds for the BODE.MODE 5 stability test. |
| Units | s |
| Range | 0.001 to 60.000 |
| Default Value | 0 |
| Data Type | Decimal |
| See Also | BODE.MODE, BODE.MODETIMER, BODE.IFTHRESH, BODE.VFLIMIT, BODE.VFTHRESH |
| Start Version | M_01-02-10-000 |

Description

When BODE.MODE is set to 5, the firmware monitors the drive current (IL.CMD). When IL.CMD goes above BODE.IFTHRESH, an internal counter records the length of time IL.CMD was above BODE.IFTHRESH. If the internal counter reaches BODE.IFLIMIT, Fault 133 – Instability during Autotune will be generated.

The smaller BODE.IFLIMIT, the quicker Fault 133 will be generated when IL.CMD exceeds BODE.IFLIMIT.

Example

Set BODE.IFTHRESH to 6 Amps:

```
-->BODE.IFTHRESH 6
```

Set BODE.IFLIMIT to 0.500 seconds:

```
-->BODE.IFLIMIT 0.5
```

Set BODE.MODE to 5 to enable stability detection:

```
BODE.MODE 5
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

F133

BODE.IFTHRESH

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the current fault threshold for the BODE.MODE 5 stability test. |
| Units | A |
| Range | 0.001 to DRV.IPEAK or MOTOR.IPEAK (whichever is lowest) A |
| Default Value | 0 A |
| Data Type | Decimal |
| See Also | BODE.MODE, BODE.MODETIMER, BODE.VFLIMIT, BODE.VFTHRESH, BODE.IFLIMIT |
| Start Version | M_01-02-10-000 |

Description

When BODE.MODE is set to 5, the firmware monitors the drive current (IL.CMD). When IL.CMD goes above BODE.IFTHRESH, an internal counter records the length of time IL.CMD was above BODE.IFTHRESH. If the internal counter reaches BODE.IFLIMIT, Fault 133 (Instability during Autotune) is generated.

Example

Set BODE.IFTHRESH to 6 Amps:

```
-->BODE.IFTHRESH 6
```

Set BODE.IFLIMIT to 0.500 seconds:

```
-->BODE.IFLIMIT 0.5
```

Set BODE.MODE to 5 to enable stability detection:

```
BODE.MODE 5
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

F133

BODE.INJECTPOINT

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets whether the excitation uses current or velocity excitation type. |
| Units | N/A |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | BODE.IAMP, BODE.MODE, BODE.VAMP |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 44 | M_01-03-00-000 |

Description

BODE.INJECTPOINT sets whether the excitation uses current or velocity excitation type.

| BODE.INJECTPOINT | Description |
|------------------|-------------|
| 0 | None |
| 1 | Current |
| 2 | Velocity |

Example

Set BODE.INJECTPOINT to current:

```
-->BODE.INJECTPOINT 1
```

Get BODE.INJECTPOINT (already set to current):

```
-->BODE.INJECTPOINT 1
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

15 Scope

1.2.1.5 Bode (set command source)

6.5 Settings

F126

Error: Invalid Bode plot mode for this function. and others)

BODE.MODE

| General Information | |
|---------------------|----------------------------------|
| Type | R/W Parameter |
| Description | Sets the mode of the excitation. |
| Units | N/A |
| Range | 0 to 4 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | BODE.INJECTPOINTBODE.VAMP |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 46 | M_01-03-00-000 |

Description

BODE.MODE sets the mode of the excitation. The excitation can be set to the modes shown in the table below. BODE.MODE is always set to **None** when Ethernet communication is disconnected. The peak amplitude of the excitation is set by either BODE.IAMP or BODE.VAMP (depending on BODE.INJECTPOINT).

BODE.MODE is subject to a watchdog timer (BODE.MODETIMER) as follows:

- If BODE.MODETIMER is 0, then BODE.MODE is not affected.
- If BODE.MODETIMER is set to a value greater than 0, then BODE.MODE will be set to 0 (None) after the BODE.MODETIMER time milliseconds.
- If BODE.MODE is a nonzero value, and you reset BODE.MODE to another nonzero value, you will reset the watchdog timer. This mechanism is intended to turn off the excitation signal if you lose communication with the drive.

| BODE.MODE | Description | Comments |
|-----------|-------------|---|
| 0 | None | Turns all excitation off |
| 1 | PRB | Uses Pseudo Random Binary (PRB) excitation. PRB is a signal that is always +/- peak amplitude, varying only in phase. PRB excitation results in a flat excitation frequency spectrum. PRB results in a high peak excitation amplitude, which can help minimize friction in a frequency response test. PRB excitation repeats every $(2^{BODE.PRBDEPTH}) / BODE.EXCITEGAP$ drive samples. This repetition can be used to reveal the effects of friction. |
| 2 | Sine | Uses Sine excitation |
| 3 | Noise | Uses random noise excitation. Noise is a random number generator that varies between +/- peak amplitude. |
| 4 | Offset | Sets a torque offset equal to BODE.IAMP |

Example

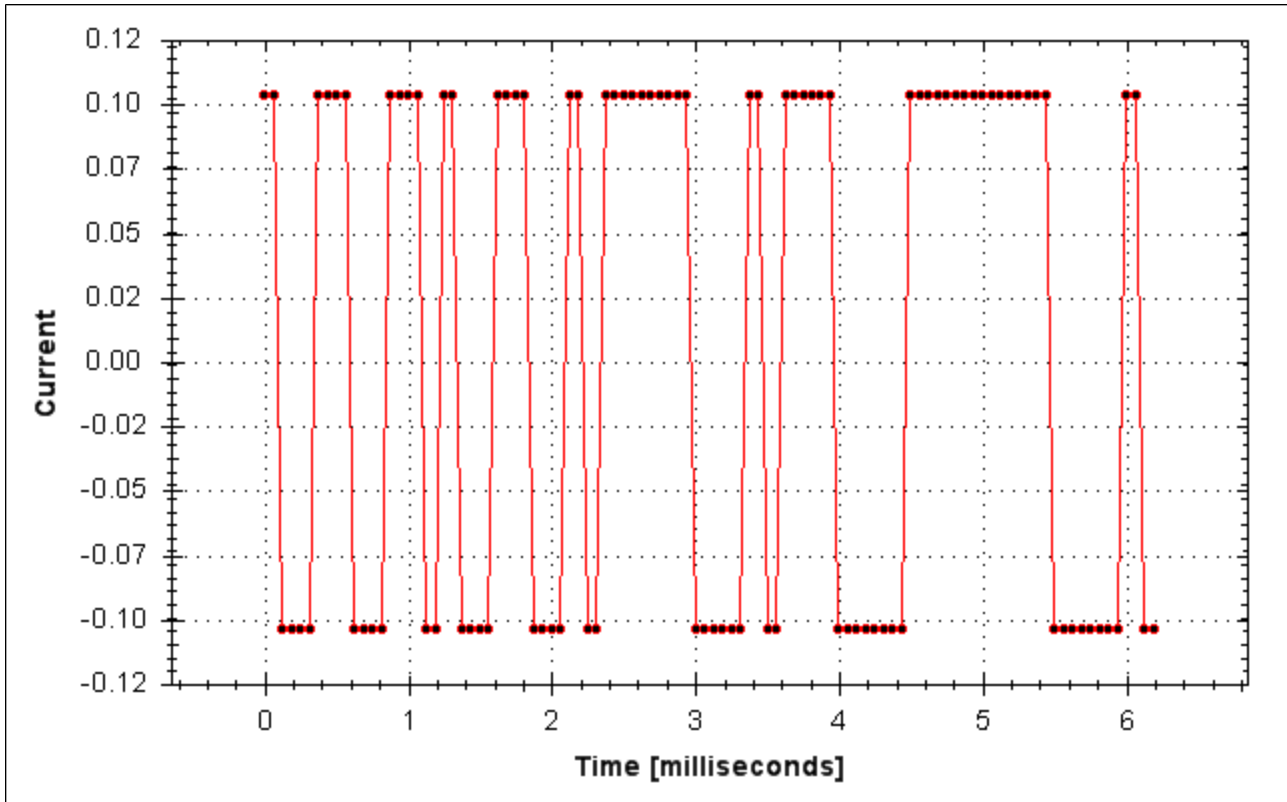
Set BODE.MODE to PRB:

```
-->BODE.MODE 1
```

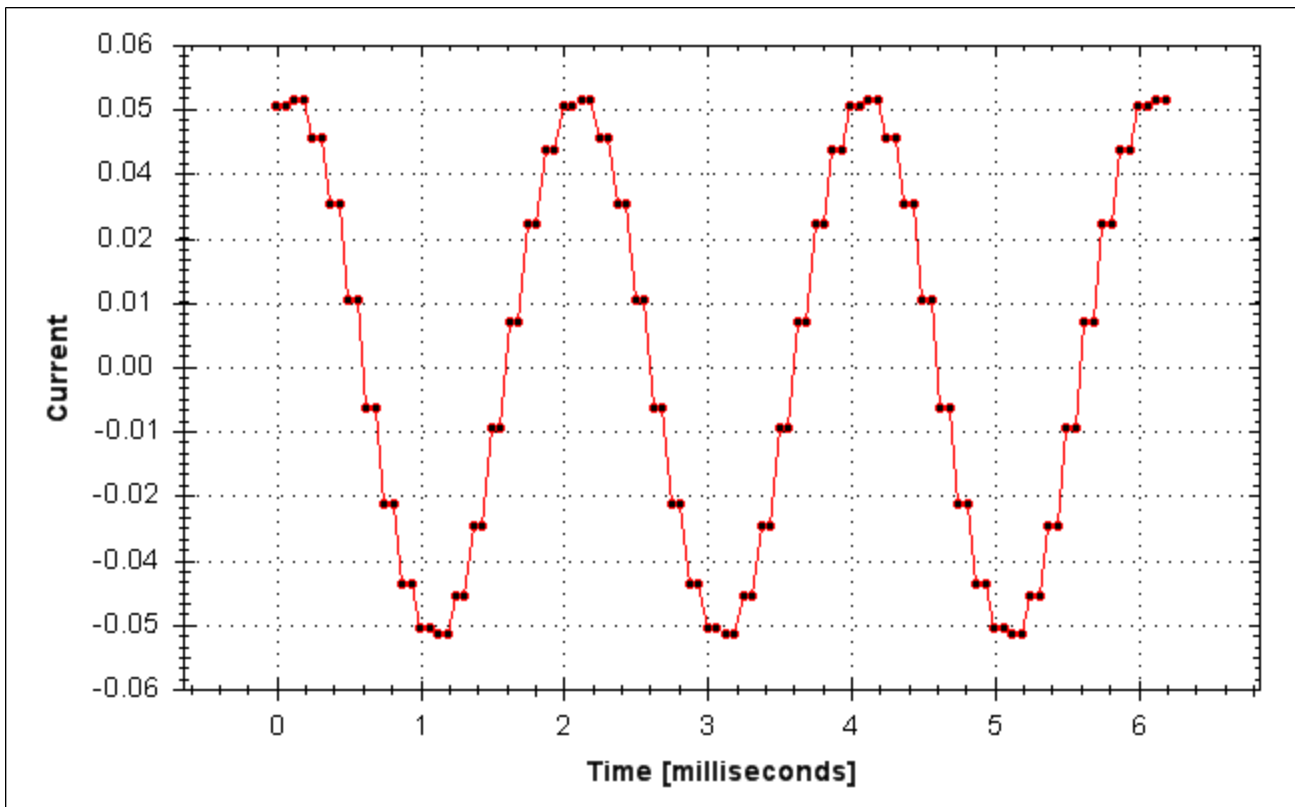
Get BODE.MODE (already set to PRB):

```
-->BODE.MODE 1
```

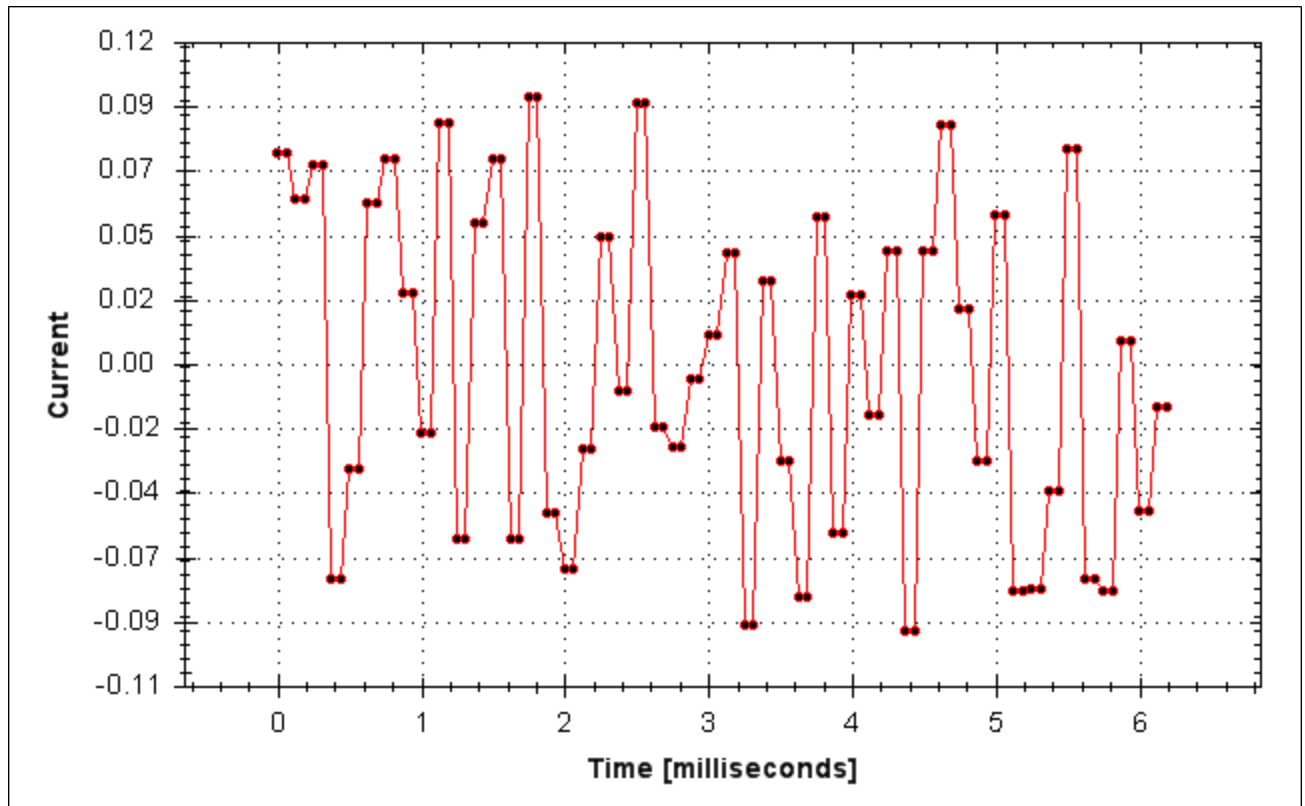
PRB excitation:



Sine excitation:



Noise excitation:



Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

15 Scope

1.2.1.5 Bode (set command source)

6.5 Settings

F126

Error: Invalid Bode plot mode for this function. and others)

BODE.MODETIMER

| General Information | |
|---------------------|--|
| Type | R/W Parameter |
| Description | Sets the watchdog timer of the excitation. |
| Units | ms |
| Range | 0 to 268,435,456 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | BODE.MODE |
| Start Version | M_1-03-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 48 | M_01-03-00-000 |

Description

BODE.MODETIMER sets the watchdog timer for the excitation. This watchdog is used to automatically turn off the excitation of the system if communication is lost. It is highly recommended that you use the watchdog for any excitation measurements. The WorkBench Performance Servo Tuner and Bode tool automatically use these values, requiring no action from you.

If the BODE.MODETIMER is a nonzero value, the Bode watchdog is enabled. BODE.MODE will be set to 0 (None) after the BODE.MODETIMER value elapses. To reset the watchdog timer, reset BODE.MODE to a non-zero value.

| BODE.MODETIMER | Comments |
|----------------|--|
| 0 | BODE.MODE is left at the value you set it to. |
| > 0 | <p>Uses pseudo random binary (PRB) excitation. PRB is a signal that is always +/- peak amplitude, varying only in phase.</p> <p>PRB excitation results in a flat excitation frequency spectrum. PRB also results in a high peak excitation amplitude, which can help minimize friction in a frequency response test.</p> <p>PRB excitation repeats every $(2^{BODE.PRBDEPTH})/BODE.EXCITEGAP$ drive samples. This repetition can be used to reveal the effects of friction.</p> |

Example

Disable BODE.MODETIMER:

```
-->BODE.MODETIMER //
```

Set to 0 to disable the watchdog

```
0
```

```
-->BODE.MODE // Observe starting state of the Bode mode
```

```
0
```

```
-->BODE.MODE 1 // Set Bode mode to PRB
```

```
-->BODE.MODE // Observe Bode mode state is the same after 0.5 seconds
```

```
1
```

```
-->BODE.MODE // Observe Bode mode state is the same after 10 seconds
```

```
1
```

Enable BODE.MODETIMER:

```

->BODE.MODETIMER 1000 // Set watchdog to 1 second
->BODE.MODE 1 // Set Bode mode to PRB
->BODE.MODE // Observe Bode mode state is the same after 0.5 seconds
1
->BODE.MODE // Observe Bode mode state has been set to zero after 1.0 seconds
0

```

Enable and reenale BODE.MODETIMER:

```

->BODE.MODETIMER 2500 // Set watchdog to 2.5 seconds
->BODE.MODE 1 // Set Bode mode to PRB
->BODE.MODE // Observe Bode mode state is the same after 1.5 seconds
1
->BODE.MODE 1 // Set Bode mode to PRB, resetting the watchdog timer to the original 2.5 second value
set above.
->BODE.MODE // Observe Bode mode state is the same after 3.0 seconds after the original enabling of
BODE.MODE 1
->BODE.MODE // Observe Bode mode state has been set to zero after 4.0 seconds after the original enabling
of BODE.MODE
0

```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

15 Scope

1.2.1.5 Bode (set command source)

6.5 Settings

F126

Error: Invalid Bode plot mode for this function. and others)

BODE.PRBDDEPTH

| General Information | |
|---------------------|--|
| Type | R/W Parameter |
| Description | Sets the length of the PRB signal before it repeats. |
| Units | NA |
| Range | 4 to 19 |
| Default Value | 19 |
| Data Type | Integer |
| See Also | BODE.MODE, BODE.INJECTPOINT, BODE.IAMP, BODE.VAMP |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 50 | M_01-03-00-000 |

Description

BODE.PRBDDEPTH sets the length of the PRB signal before it repeats. This applies only when BODE.MODE = PRB. The PRB excitation will repeat after $(2^{\text{BODE.PRBDDEPTH}}) / \text{BODE.EXCITEGAP}$ drive samples.

Example

Set BODE.PRBDDEPTH to 19:

```
-->BODE.PRBDDEPTH 19
```

Get BODE.PRBDDEPTH (already set to 19):

```
-->BODE.PRBDDEPTH 19
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

15 Scope

1.2.1.5 Bode (set command source)

6.5 Settings

F126

Error: Invalid Bode plot mode for this function. and others)

BODE.VAMP

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the amplitude of the excitation when in velocity mode. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEARUNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, µm/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 degree/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 8,000.000 mm/s 0.000 to 8,000,000.000 µm/s 0.000 to 1,250.000 custom units/s |
| Default Value | 0 |
| Data Type | Float |
| See Also | BODE.MODE, BODE.INJECTPOINT |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 52 | M_01-03-00-000 |

Description

BODE.VAMP sets the amplitude of the excitation when in velocity mode as set in BODE.INJECTPOINT.

Example

Set the excitation velocity to 100 RPM

```
-->BODE.VAMP 100
```

Get the excitation velocity(already set to 100 RPM)

```
-->BODE.VAMP
100.000 [rpm]
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

15 Scope

1.2.1.5 Bode (set command source)

6.5 Settings

F126

Error: Invalid Bode plot mode for this function. and others)

BODE.VFLIMIT

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the velocity fault duration limit (seconds) for the BODE.MODE 5 stability test |
| Units | s |
| Range | 0.001 to 60.000 |
| Default Value | 0 |
| Data Type | Decimal |
| See Also | BODE.MODE, BODE.MODETIMER, BODE.IFLIMIT, BODE.IFTHRESH, BODE.VFTHRESH |
| Start Version | M_01-02-10-000 |

Description

When BODE.MODE is set to 5, the firmware monitors the drive feedback velocity VL.FB. When VL.FB goes above BODE.VFTHRESH, an internal counter records the length of time VL.FB was above BODE.VFTHRESH. If the internal counter reaches BODE.VFLIMIT, Fault 133 – Instability during Autotune will be generated. The smaller BODE.VFLIMIT, the quicker Fault 133 will be generated when VL.FB exceeds BODE.VFLIMIT.

Example

Set BODE.VFTHRESH to 10 RPM:

```
-->BODE.VFTHRESH 10
```

Set BODE.VFLIMIT to 0.500 seconds

```
-->BODE.VFLIMIT 0.5
```

Set BODE.MODE to 5 to enable stability detection

```
-->BODE.MODE 5
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

F133

BODE.VFTHRESH

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the current fault threshold for the BODE.MODE 5 stability test. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEARUNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, µm/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 8,000.000 mm/s 0.000 to 8,000,000.000 µm/s 0.000 to 1,250.000 custom units/s |
| Default Value | 0 |
| Data Type | Decimal |
| See Also | BODE.MODE, BODE.MODETIMER, BODE.IFLIMIT, BODE.I-FTHRESH, BODE.VFLIMIT |
| Start Version | M_01-02-10-000 |

Description

When BODE.MODE is set to 5, the firmware monitors the drive feedback velocity VL.FB. When VL.FB goes above BODE.VFTHRESH, an internal counter records the length of time VL.FB was above BODE.VFTHRESH. If the internal counter reaches BODE.VFLIMIT, Fault 133 – Instability during Autotune will be generated.

The smaller BODE.VFLIMIT, the quicker Fault 133 will be generated when VL.FB exceeds BODE.VFLIMIT.

Example

Set BODE.VFTHRESH to 10 RPM:

```
-->BODE.VFTHRESH 10
```

Set BODE.VFLIMIT to 0.500 seconds:

```
-->BODE.VFLIMIT 0.5
```

Set BODE.MODE to 5 to enable stability detection:

```
-->BODE.MODE 5
```

Related Topics

14.3 Using the Performance Servo Tuner

14.3.4 Using the Performance Servo Tuner: Advanced

F133

CAP Parameters

CAP0.EDGE, CAP1.EDGE

| General Information | |
|----------------------|----------------------------|
| Type | R/W Parameter |
| Description | Selects the capture edge. |
| Units | N/A |
| Range | 1 to 3 |
| Default Value | 1 |
| Data Type | U8 |
| See Also | CAP0.PREEDGE, CAP1.PREEDGE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 54 CAP0.EDGE | M_01-03-00-000 |
| | 80 CAP1.EDGE | |

Description

The filtered trigger source is monitored for rising edge, falling edge, or both edges. The event mode logic may ignore the precondition edge detection; however, the trigger always uses edge detection.

The precondition logic has an identical feature controlled by CAP0.PREEDGE, CAP1.PREEDGE.

| Value | Description |
|-------|--------------|
| 0 | Reserved |
| 1 | Rising edge |
| 2 | Falling edge |
| 3 | Both edges |

Related Topics

8.11 Using Position Capture

CAP0.EN, CAP1.EN

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Enables or disables the related capture engine. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------|-----------------|----------------------|
| Modbus | 56 CAP0.EN | M_01-03-00-000 |
| | 82 CAP1.EN | |

Description

This parameter enables or disables the related capture engine. After each successful capture event, this parameter is reset to 0 and must be activated again for the next capture. Also note that CAP0.PLFB, CAP1.PLFB is set to 0 when this parameter is set to 1.

0 = Disable

1 = Enable

Related Topics

8.11 Using Position Capture

CAP0.EVENT, CAP1.EVENT

| General Information | |
|---------------------|----------------------------------|
| Type | R/W Parameter |
| Description | Controls the precondition logic. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3460h/5 | CAP0.EVENT |
| | 3460h/6 | CAP1.EVENT |
| Modbus | 58 | CAP0.EVENT |
| | 84 | CAP1.EVENT |

Description

The event mode controls use of the precondition logic. If this field is not 0, then the precondition input is selected by CAPx.TRIGGER. If this field is 1, then the precondition edge is selected by the CAPx.PREEDGE. The four event modes are listed below.

| Event | Description |
|-------|---|
| 0 | Precondition settings ignored. |
| 1 | Trigger on first trigger event after selected edge on precondition input. |
| 2 | Trigger on first trigger event to occur while precondition input is 1 |
| 3 | Trigger on first trigger event to occur while precondition input is 0. |

Example

Event 0

The following diagram shows an example of Event = 0 (trigger on edge, trigger edge = rising). In this mode, the precondition logic is ignored.

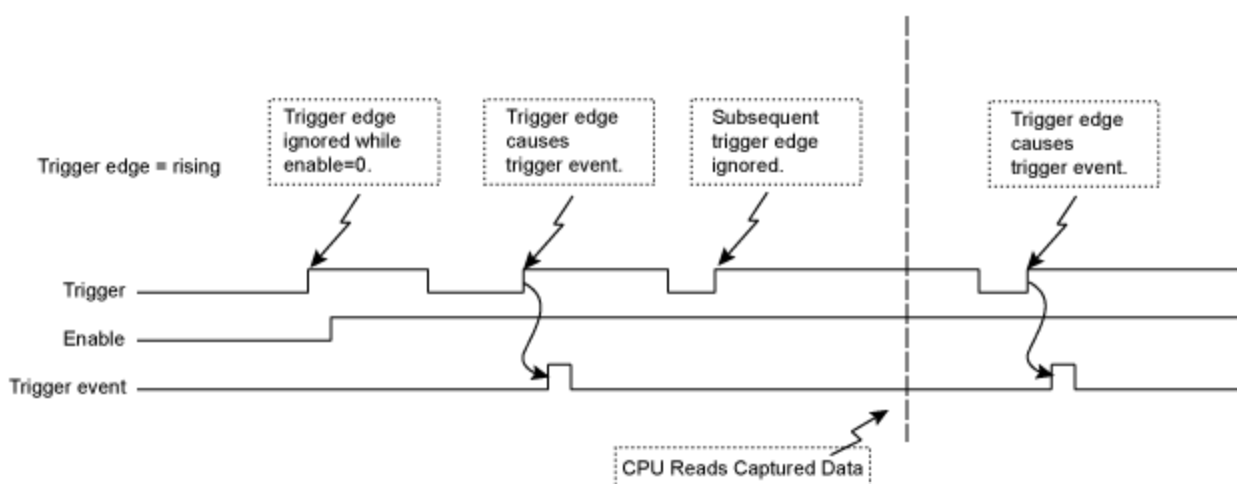


Figure 1: Trigger Edge Mode

Events 2 and 3 (Trigger edge while precondition = 0 or 1)

In these events, the precondition logic samples the current (post-filter) state of the selected precondition source input. The capture engine looks for a trigger edge while the precondition input is at a "1" or "0" state.

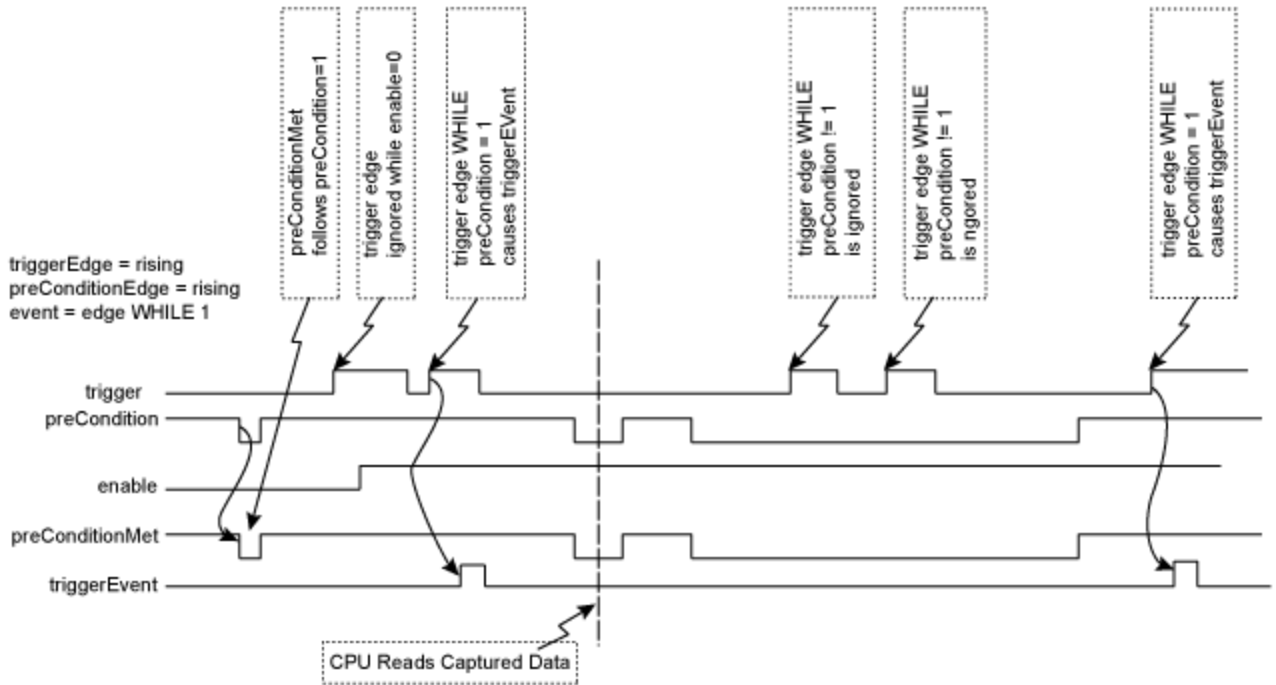


Figure 2: Trigger edge WHILE precondition edge

Event 1 (Trigger edge after precondition)

In this event, each trigger event requires Enable=1, a new precondition edge, followed by a new trigger edge. The sequence requirements are shown in the figure below.

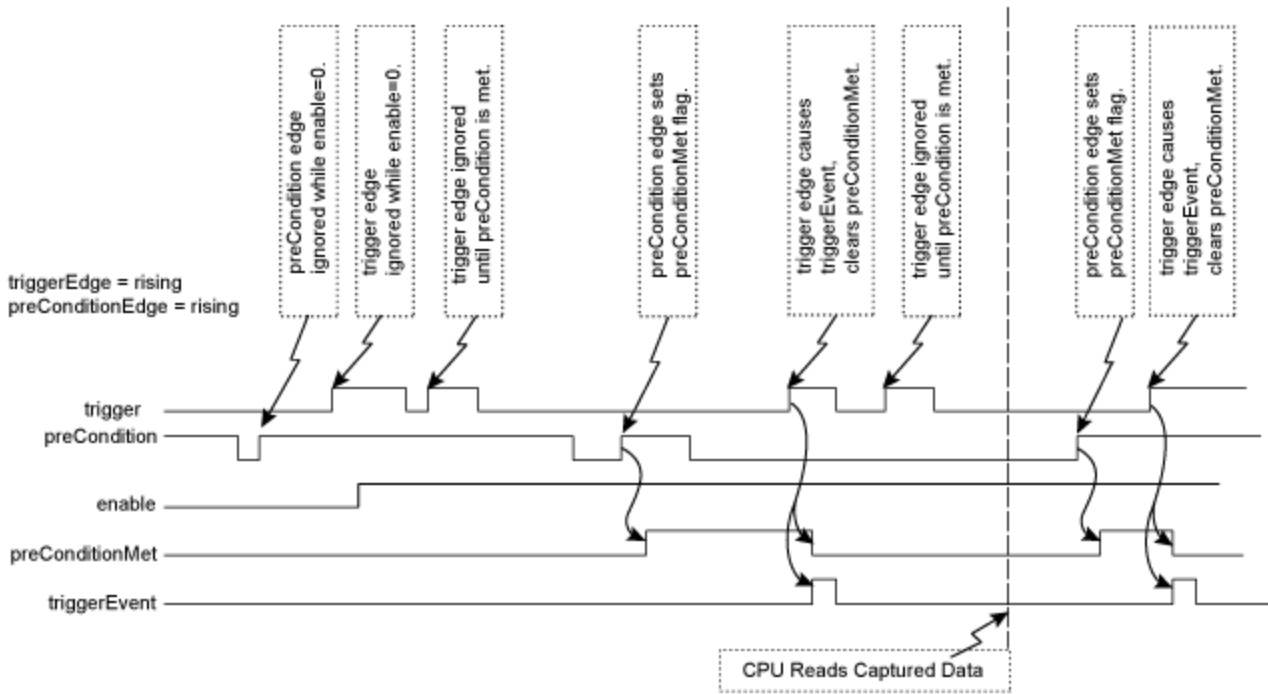


Figure 3: Trigger edge after precondition edge

Note: If the precondition and trigger edges occur at the same time, it is not a valid trigger event. A subsequent trigger edge must occur after the precondition edge. The same time resolves to a single 40 ns clock tick in the trigger event logic (after the optional filter function as well as any sensor, cable, or noise delays).

Related Topics

8.11 Using Position Capture

CAP0.FILTER, CAP1.FILTER

| General Information | |
|---------------------|---|
| Type | R/W Parameter |
| Description | Sets the filter for the capture source input. |
| Units | N/A |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | CAP0.PREFILTER, CAP1.PREFILTER |
| Start Version | M_01-00-00-000 |
| End Version | M_01-03-00-000 |

Description

These parameters are not functional in M_01-03-00-000. In future releases, you can use DINx.FILTER to select a filter on the input channel.

Related Topics

DIN1.FILTER TO DIN7.FILTER

CAP0.MODE, CAP1.MODE

| General Information | |
|----------------------|-----------------------------|
| Type | NV Parameter |
| Description | Selects the captured value. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------|----------------------|
| EtherCAT COE and CANopen | 3460h/3 CAP0.MODE | M_01-00-00-000 |
| | 3460h/4 CAP1.MODE | |
| Modbus | 62 CAP0.MODE | M_01-03-00-000 |
| | 88 CAP1.MODE | |

Description

Mode 0 is the standard position capture, which stores PL.FB. Data can be retrieved with CAP0.PLFB, CAP1.PLFB.

Mode 1 is the drive internal time capture. Data can be retrieved with CAP0.T, CAP1.T.

Mode 2 is the KAS EtherCAT distributed clock time (DCT) capture. Instead of using a position value, the DCT is calculated. There is no user parameter to retrieve the captured DCT. Attempting to set Mode = 2 with anything other than an EtherCAT system will result in an invalid parameter error.

Mode 3 is the capture of the primary encoder signal. This mode is used to home onto a feedback index. This mode sets the other parameters needed for this mode. These parameters can be changed later, but this is not recommended unless the input source of the index signal varies. Parameters set in this mode are:

- CAPx.TRIGGER 10: index mark of primary encoder
- CAPx.EDGE 1: rising edge
- CAPx.EVENT 0: ignore precondition

Also the capture engine is immediately enabled and is continuously triggered again.

Mode 4 is similar to Mode 0 (standard position capture), except that the re-enabling of the capture is done automatically. This mode can be used for the registration move.

Related Topics

8.11 Using Position Capture

CAP0.PLFB, CAP1.PLFB

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads captured position value. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, custom units, 16-bit counts Linear: counts, mm, μ m, custom units, 16-bit counts |
| Range | Full range of a signed 64 bit variable |
| Default Value | 0 |
| Data Type | S64 |
| See Also | UNIT.PROTARY, UNIT.PLINEAR |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-----------------------|----------------------|
| EtherCAT COE and CANopen | 20A0h/0 CAP0.PLFB | M_01-00-00-000 |
| | 20A1h/0 CAP0.PLFB | |
| | 20A2h/0 CAP1.PLFB | |
| | 20A3h/0 CAP1.PLFB | |
| Modbus | 64 (64-bit) CAP0.PLFB | M_01-03-00-000 |
| | 90 (64-bit) CAP1.PLFB | |

Description

This parameter reads the captured position value scaled to actual set units. See UNIT.PROTARY or UNIT.PIN for these units.

Related Topics

8.11 Using Position Capture

CAP0.PREEDGE, CAP1.PREEDGE

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Selects the capture precondition edge. |
| Units | N/A |
| Range | 1 to 3 |
| Default Value | 1 |
| Data Type | U8 |
| See Also | CAP0.EDGE, CAP1.EDGE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3460h/7 | CAP0.PREEDGE |
| | 3460h/8 | CAP1.PREEDGE |
| Modbus | 68 | CAP0.PREEDGE |
| | 94 | CAP1.PREEDGE |

Description

The precondition edge is monitored for rising edge, falling edge, or both. The event mode logic may ignore the precondition edge detection (trigger always uses edge detection).

The filtered trigger source has an identical feature controlled by CAP0.EDGE, CAP1.EDGE.

| Value | Description |
|-------|--------------|
| 0 | Reserved |
| 1 | Rising edge |
| 2 | Falling edge |
| 3 | Both edges |

Related Topics

8.11 Using Position Capture

CAP0.PREFILTER, CAP1.PREFILTER

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the filter for the precondition input source. |
| Units | N/A |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | CAP0.FILTER, CAP1.FILTER |
| Start Version | M_01-00-00-000 |
| End Version | M_01-03-00-000 |

Description

These parameters are not functional in M_01-03-00-000. In future releases, you can use DINx.FILTER to select a filter on the input channel.

Related Topics

DIN1.FILTER TO DIN7.FILTER

CAP0.PRESELECT, CAP1.PRESELECT

| General Information | |
|----------------------|--------------------------------|
| Type | R/W Parameter |
| Description | Sets the precondition trigger. |
| Units | N/A |
| Range | 0 to 11 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | CAP0.TRIGGER, CAP1.TRIGGER |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3460h/9 | CAP0.PRESELECT |
| | 3460h/10 | CAP1.PRESELECT |
| Modbus | 72 | CAP0.PRESELECT |
| | 98 | CAP1.PRESELECT |

Description

This parameter specifies the input signal for the precondition trigger.

| Trigger Source | Input Name |
|----------------|----------------------|
| 0 | General Input 1 (X7) |
| 1 | General Input 2 (X7) |
| 2 | General Input 3 (X7) |
| 3 | General Input 4 (X7) |
| 4 | General Input 5 (X8) |
| 5 | General Input 6 (X7) |
| 6 | General Input 7 (X7) |
| 7 | RS485 Input 1 (X9) |
| 8 | RS485 Input 2 (X9) |
| 9 | RS485 Input 3 (X9) |
| 10 | Primary Index |

Related Topics

8.11 Using Position Capture

CAP0.STATE, CAP1.STATE

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Indicates whether or not trigger source was captured. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|--------------------|----------------------|
| Modbus | 74 CAP.STATE | M_01-03-00-000 |
| | 100 CAP1.STATE | |

Description

When enabling the capture (CAP0.EN, CAP1.EN), this parameter is set to 0 until the next event is captured.

0 = Not captured

1 = Captured

Related Topics

8.11 Using Position Capture

CAP0.T, CAP1.T

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads time capture (if time capture was configured). |
| Units | ns |
| Range | N/A |
| Default Value | N/A |
| Data Type | U32 |
| See Also | CAP0.MODE, CAP1.MODE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 20A0h/0 CAP0.T | M_01-00-00-000 |
| | 20A1h CAP0.T | |
| | 20A2h/0 CAP1.T | |
| | 20A3h/0 CAP1.T | |
| Modbus | 76 CAP0.T | M_01-03-00-000 |
| | 102 CAP1.T | |

Description

If time capture was configured, the captured time is stored in this parameter. The reference time is the occurrence of the last MTS signal (recurring every 62.5 μ s), so this is a purely drive internal time.

Related Topics

8.11 Using Position Capture

CAP0.TRIGGER, CAP1.TRIGGER

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Specifies the trigger source for the position capture. |
| Units | N/A |
| Range | 0 to 11 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | CAP0.PRESELECT, CAP1.PRESELECT |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3460h/1 | CAP0.TRIGGER |
| | 3460h/2 | CAP1.TRIGGER |
| Modbus | 78 | CAP0.TRIGGER |
| | 104 | CAP1.TRIGGER |

Description

This parameter specifies the trigger source (capture input signal).

| Trigger Source | Input Name |
|----------------|-----------------|
| 0 | General Input 1 |
| 1 | General Input 2 |
| 2 | General Input 3 |
| 3 | General Input 4 |
| 4 | General Input 5 |
| 5 | General Input 6 |
| 6 | General Input 7 |
| 7 | RS485 Input 1 |
| 8 | RS485 Input 2 |
| 9 | RS485 Input 3 |
| 10 | Primary Index |
| 11 | Tertiary Index |

Related Topics

8.11 Using Position Capture

CS Parameters

Controlled stop (CS) parameters set the values for the controlled stop process.

CS.DEC

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the deceleration value for the controlled stop process. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s ² , (custom units)/s ² , rad/s ² Linear: counts/s ² , mm/s ² , μm/s ² , (custom units)/s ² |
| Range | Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s ² 0.155 to 4,166,666.752 (custom units)/s ² 0.012 to 5,235,987.968 rad/s ² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s ² 0.031*MOTOR.PITCH to 833333.333*MOTOR.PITCH mm/s ² 30.994*MOTOR.PITCH to 833333333.333*MOTOR.PITCH μm/s ² 0.155 to 4,166,666.667 (custom units)/s ² |
| Default Value | Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s ² 833.333 (custom units)/s ² 1,047.2 rad/s ² Linear: 715,840,000.000 counts/s ² 166.714*MOTOR.PITCHMOTOR.PITCH mm/s ² 166,714.191*MOTOR.PITCHMOTOR.PITCH μm/s ² 833.571 (custom units)/s ² |
| Data Type | Float |
| See Also | CS.VTHRESH, CS.TO, DRV.DIS, DIN1.MODE TO DIN7.MODE, DRV.DISMODE, DRV.DISSOURCES |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3440h/1 | M_01-00-00-000 |
| Modbus | 106 (64-bit) | M_01-03-00-000 |

Description

This parameter sets the deceleration value for the controlled stop process.

Related Topics

10.10 Controlled Stop

10.1 Digital Inputs and Outputs

17.1 Fault and Warning Messages (this table indicates faults for which a controlled stop occurs)

CS.STATE

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Returns the internal status of the controlled stop process. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | CS.DEC, CS.VTHRESH, CS.TODRV.DISMODE, DRV.DISSOURCES |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3441h/0 | M_01-00-00-000 |
| Modbus | 110 | M_01-03-00-000 |

Description

CS.STATE returns the internal state machine value of the controlled stop.

0 = controlled stop is not occurring.

1 = controlled stop is occurring

Related Topics

10.10 Controlled Stop

10.1 Digital Inputs and Outputs

17.1 Fault and Warning Messages (this table indicates faults for which a controlled stop occurs)

CS.TO

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the time value for the drive velocity to be within CS.VTHRESH. |
| Units | ms |
| Range | 1 to 30,000 ms |
| Default Value | 6 ms |
| Data Type | Integer |
| See Also | CS.DEC, CS.VTHRESH, CS.STATE, DRV.DIS, DIN1.MODE TO DIN7.MODE, DRV.DISM, DRV.DISSOURCES |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3440h/3 | M_01-00-00-000 |
| Modbus | 112 | M_01-03-00-000 |

Description

CS.TO is the time value for the drive velocity to be within CS.VTHRESH before the drive disables.

Example

Set time value to 100 ms:

```
-->CS.TO 100
```

Related Topics

10.10 Controlled Stop

10.1 Digital Inputs and Outputs

17.1 Fault and Warning Messages (this table indicates faults for which a controlled stop occurs)

CS.VTHRESH

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the velocity threshold for the controlled stop. |
| Units | rpm, rps, deg/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 8,000.000 mm/s 0.000 to 8,000,000.000 µm/s 0.000 to 1,250.000 custom units/s |
| Default Value | 5 rpm |
| Data Type | Float |

| General Information | |
|----------------------|--|
| See Also | CS.DEC, CS.TO, CS.STATE, DRV.DIS, DIN1.MODE TO DIN7.MODE, DRV.DISM, DRV.DISSOURCES |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|-----------------------------|----------------|----------------------|
| EtherCAT and CANopen | 3440h/2 | M_01-00-00-000 |
| Modbus | 114 | M_01-03-00-000 |

Description

CS.VTHRESH is the velocity threshold for the controlled stop algorithm.

Example

Set velocity threshold for controlled stop at 100 rpm:

```
-->CS.VTHRESH 100
```

Related Topics

10.10 Controlled Stop

10.1 Digital Inputs and Outputs

17.1 Fault and Warning Messages (this table indicates faults for which a controlled stop occurs)

DIN Parameters

DIN.HCMD1 TO DIN.HCMD4

| General Information | |
|----------------------|---|
| Type | NV parameter |
| Description | A buffer of commands to be used in digital input "command buffer" mode. |
| Units | None |
| Range | A string of up to 128 characters |
| Default Value | <Empty> |
| Data Type | String |
| See Also | DINx.MODE, DINx.PARAM, DIN.LCMDx |
| Start Version | M_01-02-08-000 |

Description

DIN.HCMDx sets the string of commands to be used in the digital input mode command buffer. Digital input mode **9-Command buffer** can execute four different sets of command buffers.

Each set of command buffers contains two buffers:

- High buffer: Executes upon a rising edge of a digital input.
- Low buffer: Executes upon a falling edge of a digital input.

DIN.HCMDx sets the string for the four high buffers (depending on x).

Example

Set the command buffer mode to digital input 1:

```
DIN1.MODE 9
```

Set the first sets of buffers to digital input 1:

```
DIN1.PARAM 1
```

Set the command DRV.OPMODE 0 to the high buffer:

```
DIN.HCMD1 DRV.OPMODE 1
```

Now, upon a rising edge in digital input 1, the drive mode is 1.

Related Topics

10.2 Command Buffer

10.1 Digital Inputs and Outputs

DIN.LCMD1 to DIN.LCMD4

| General Information | |
|----------------------|--|
| Type | NV parameter |
| Description | A buffer of commands to be used in digital input "command buffer" mode. |
| Units | N/A |
| Range | A string of up to 128 characters |
| Default Value | Empty |
| Data Type | String |
| See Also | DIN1.MODE TO DIN7.MODE, DIN1.PARAM TO DIN7.PARAM, DIN.HCMD1 TO DIN.HCMD4 |
| Start Version | M_01-02-08-000 |

Description

DIN.LCMDx sets the string of commands to be used in the digital input mode command buffer. Digital input mode **9-Command buffer** can execute four different sets of command buffers.

Each set of command buffers contains two buffers:

- High buffer: Executes upon a rising edge of a digital input.
- Low buffer: Executes upon a falling edge of a digital input.

DIN.LCMDx sets the string for the four "low" buffers, depending on x.

Example

Set the command buffer mode to digital input 1:

```
DIN1.MODE 9
```

Set the first sets of buffers to digital input 1:

```
DIN1.PARAM 1
```

Set the command DRV.OPMODE 0 to the "low buffer":

```
DIN.LCMD1 DRV.OPMODE 0
```

Now, upon a falling edge in digital input 1, the drive mode is 0.

Related Topics

10.2 Command Buffer

10.1 Digital Inputs and Outputs

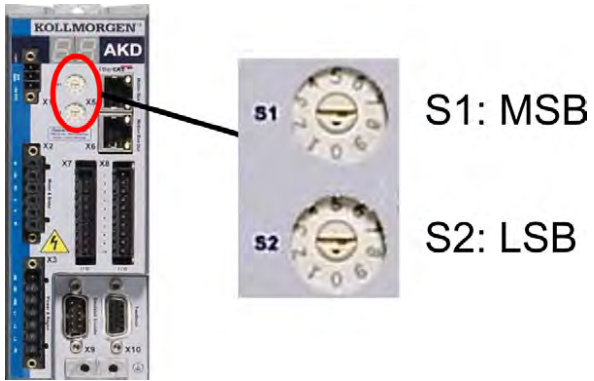
DIN.ROTARY

| General Information | |
|----------------------|------------------------------|
| Type | R/O Parameter |
| Description | Reads the rotary knob value. |
| Units | N/A |
| Range | 0 to 99 |
| Default Value | N/A |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 116 | M_01-03-00-000 |

Description

DIN.ROTARY reads the rotary knob value.



Related Topics

10.1 Digital Inputs and Outputs

DIN.STATES

| General Information | |
|----------------------|---------------------------------|
| Type | R/O Parameter |
| Description | Reads the digital input states. |
| Units | N/A |
| Range | 0000000 to 1111111 |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 118 | M_01-03-00-000 |

Description

DIN.STATES reads the states of the seven digital inputs. The leftmost bit represents digital input 1 (DIN1) and the rightmost bit represents digital input 7 (DIN7).

Related Topics

10.1 Digital Inputs and Outputs

DIN1.FILTER TO DIN7.FILTER

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Filter mode for digital inputs 1 to 7. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 1 for DIN1 and DIN2 2 for DIN3 to DIN7 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-03-07-000 |

Description

This parameter sets the digital input filter configuration for channel x when followed with the values defined below. DINx.FILTER retrieves this information when not followed by data.

| Value | Description |
|---------------|---|
| DINX.FILTER 0 | The drive digital input channel detects all input signals with an input pulse width of ≥ 40 ns (no filtering applied). |
| DINX.FILTER 1 | The drive digital input channel detects all input signals with an input pulse width of ≥ 10.24 μ s, ± 0.64 μ s (fast filter applied). |
| DINX.FILTER 2 | The drive digital input channel detects all input signals with an input pulse width of ≥ 163 μ s, ± 10.24 μ s (standard filter applied). |
| DINX.FILTER 3 | The drive digital input channel detects all input signals with an input pulse width of ≥ 2.62 ms, ± 0.16384 ms (slow filter applied). |

Related Topics

Digital Inputs and Outputs

DIN1.INV TO DIN7.INV

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the indicated the polarity of a digital input mode. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------|-----------------|----------------------|
| Modbus | 120 DIN1.INV | M_01-03-00-000 |
| | 130 DIN2.INV | |
| | 140 DIN3.INV | |
| | 150 DIN4.INV | |
| | 160 DIN5.INV | |
| | 170 DIN6.INV | |
| | 180 DIN7.INV | |

Description

Sets the indicated the polarity of a digital input mode.

Example

DIN1.INV = 0 : Input is active high.

DIN1.INV = 1 : Input is active low.

Related Topics

10.1 Digital Inputs and Outputs

DIN1.MODE TO DIN7.MODE

| General Information | |
|---------------------|-------------------------------|
| Type | R/W Parameter |
| Description | Sets the digital input modes. |
| Units | N/A |
| Range | 0 to 22 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|-------------------------|------------------------|
| EtherCAT COE and CANopen | 3562h/0 | DIN1.MODE |
| | 3565h/0 | DIN2.MODE |
| | 3568h/0 | DIN3.MODE |
| | 356Bh/0 | DIN4.MODE |
| | 36F6h/0 | DIN5.MODE |
| | 36F9h/0 | DIN6.MODE |
| | 36FCh/0 | DIN7.MODE |
| | Profile specific SDOs/0 | DIN1.MODE TO DIN7.MODE |
| | | |
| Modbus | 122 | DIN1.MODE |
| | 132 | DIN2.MODE |
| | 142 | DIN3.MODE |
| | 152 | DIN4.MODE |
| | 162 | DIN5.MODE |
| | 172 | DIN6.MODE |
| | 182 | DIN7.MODE |
| | | M_01-03-00-000 |

Description

DIN1.MODE to DIN7.MODE parameters set the functionality of the digital inputs 1 through 7. Digital inputs and corresponding X7 and X8 pin connectors are described in the *AKD Installation Manual*, section 8.16.4, Digital Inputs. The table below summarizes the digital input modes; for detailed descriptions of each mode, see 10.1 Digital Inputs and Outputs.

| DINx.MODE | Description | Task |
|-----------|--|-----------------|
| 0 | No function; off | 0 - None |
| 1 | Fault reset | 1 - Background |
| 2 | Start motion task (use DINx.PARAM for this task) | 2 - 1 KHz |
| 3 | Motion task select bit (see Motion Tasks) | 3 - Background |
| 4 | Motion task start selected (see Motion Tasks) | 4 - 1 kHz |
| 5 | Start home (see Homing) | 5 - Background |
| 6 | Start jog | 6 - Background |
| 7 | Reserved | 7 - Background |
| 8 | Zero latch | 8 - Background |
| 9 | Command buffer | 9 - Background |
| 10 | Control fault relay | 10 - Background |
| 11 | Home reference | 11 - 1 kHz |

| DINx.MODE | Description | Task |
|-----------|---|-----------------|
| 12 | Reserved | 12 - None |
| 13 | Controlled stop (see Controlled Stop) | 13 - 1 kHz |
| 14 | Reserved | 14 - None |
| 15 | Quick stop | 15 - Background |
| 16 | Activate electronic gearing (see Electronic Gearing) | 16 - Background |
| 17 | Activate electronic gear position shift | 17 - Background |
| 18 | Positive limit switch | 18 - 4 kHz |
| 19 | Negative limit switch | 19 - 4kHz |
| 20 | Brake release | 20 - Background |
| 21 | Current limit | 21 - 4 kHz |
| 22 | Opmode and Command Source switch | 22 - Background |
| 23 | Change algebraic sign of the measured analog input voltage. | 23 - 1 kHz |

Related Topics

10.2 Command Buffer

10.1 Digital Inputs and Outputs

10.1.3 Digital Inputs

10.10 Controlled Stop

12.1 Homing

12.2 Motion Tasks

10.7 Electronic Gearing

17.2 Clearing Faults

F245

CS Parameters

DIN1.PARAM TO DIN7.PARAM

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets a value used as an extra parameter for digital inputs nodes. |
| Units | N/A |
| Range | -9,223,372,036,854,775,000 to +9,223,372,036,854,775,000 |
| Default Value | 0 |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------|-------------------------|----------------------|
| Modbus | 124 (64-bit) DIN1.PARAM | M_01-03-00-000 |
| | 134 (64-bit) DIN2.PARAM | |
| | 144 (64-bit) DIN3.PARAM | |
| | 154 (64-bit) DIN4.PARAM | |
| | 164 (64-bit) DIN5.PARAM | |
| | 174 (64-bit) DIN6.PARAM | |
| | 184 (64-bit) DIN7.PARAM | |

Description

This parameter sets a value that is used as an extra parameter for digital inputs nodes.

Example

The digital input mode "Start motion task" is used to start a motion task. This mode uses an extra parameter as the ID of the motion task to be started.

Related Topics

10.1 Digital Inputs and Outputs

DIN1.STATE TO DIN7.STATE

| General Information | |
|----------------------|---------------------------------------|
| Type | R/O Parameter |
| Description | Reads a specific digital input state. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | N/A |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------|----------------|----------------------|
| Modbus | 128 | DIN1.STATE |
| | 138 | DIN2.STATE |
| | 148 | DIN3.STATE |
| | 158 | DIN4.STATE |
| | 168 | DIN5.STATE |
| | 178 | DIN6.STATE |
| | 188 | DIN7.STATE |
| | | M_01-03-00-000 |

Description

DIN1.STATE to DIN7.STATE reads the state of one digital input according to the number identified in the command.

Related Topics

10.1 Digital Inputs and Outputs

This page intentionally left blank.

DOUT Parameters

DOUT.CTRL

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the source of digital outputs (firmware or fieldbus). |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 190 | M_01-03-00-000 |

Description

DOUT.CTRL sets the source of the digital outputs:

0 = Firmware controlled

1 = Fieldbus controlled

Related Topics

10.1.4 Digital Outputs

DOUT.RELAYMODE

| General Information | |
|----------------------|------------------------------|
| Type | R/W Parameter |
| Description | Indicates faults relay mode. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 192 | M_01-03-00-000 |

Description

DOUT.RELAYMODE indicates the faults relay mode as follows:

If DOUT.RELAYMODE= 0 and faults exist, then the relay is open.

If DOUT.RELAYMODE= 0 and faults do not exist, then the relay is closed.

If DOUT.RELAYMODE = 1 and the drive is disabled, then the relay is open.

If DOUT.RELAYMODE = 1 and the drive is enabled, then the relay is closed.

Related Topics

10.1.4 Digital Outputs

DOUT.STATES

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the state of the two digital outputs. |
| Units | N/A |
| Range | 0 to 11 |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 194 | M_01-03-00-000 |

Description

DOUT.STATES reads the states of the two digital outputs. The rightmost bit represents DOUT2 and the leftmost bit represents DOUT1.

Related Topics

10.1.4 Digital Outputs

DOUT1.MODE AND DOUT2.MODE

| General Information | |
|----------------------|-------------------------------|
| Type | NV Parameter |
| Description | Sets the digital output mode. |
| Units | N/A |
| Range | 0 to 14 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | DOUT1.PARAM AND DOUT2.PARAM |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 196 DOUT1.MODE | M_01-03-00-000 |
| | 206 DOUT2.MODE | |

Description

DOUTx.MODE sets the functionality of the digital outputs. The table below summarizes the digital output modes; for detailed descriptions of each mode, see 10.1 Digital Inputs and Outputs.

| DOUTx.MODE | Description |
|------------|----------------------------------|
| 0 | User (default = 0) |
| 1 | Mains ready |
| 2 | Software limit switch reached |
| 3 | Move complete |
| 4 | In position |
| 5 | Position greater than x |
| 6 | Position less than x |
| 7 | Drive produced warning |
| 8 | Drive enabled |
| 10 | Motor brake |
| 11 | Drive produced fault |
| 12 | Absolute velocity greater than x |
| 13 | Absolute velocity less than x |
| 14 | Homing complete |
| 15 | PLS.STATE bits or connected |

Related Topics

10.1.4 Digital Outputs

DOUT1.PARAM AND DOUT2.PARAM

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets extra parameters for the digital outputs. |
| Units | N/A |
| Range | DOUT1.PARAM: -357,913.941 to 357,913.941 DOUT2.PARAM: -79,164,837,199.872 to 79,164,837,199.856 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|--------------------------|----------------------|
| Modbus | 198 (64-bit) DOUT1.PARAM | M_01-03-00-000 |
| | 208 (64-bit) DOUT2.PARAM | |

Description

DOUT1.PARAM and DOUT2.PARAM set the extra parameter needed for the digital outputs calculations, respectively.

Related Topics

10.1.4 Digital Outputs

DOUT1.STATE AND DOUT2.STATE

| General Information | |
|----------------------|---------------------------------|
| Type | R/O Parameter |
| Description | Reads the digital output state. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | N/A |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------|-----------------|----------------------|
| Modbus | 202 DOUT1.STATE | M_01-03-00-000 |
| | 212 DOUT2.STATE | |

Description

DOUT1.STATE and DOUT2.STATE read the state of one digital output according to the value stated in the command. These parameters can also be used to set a value of one digital output (only if the output mode is idle).

Related Topics

10.1.4 Digital Outputs

DOUT1.STATEU AND DOUT2.STATEU

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the state of the digital output node. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-01-01-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------|------------------|----------------------|
| Modbus | 204 DOUT1.STATEU | M_01-03-00-000 |
| | 214 DOUT2.STATEU | |

Description

DOUT1.STATEU and DOUT2.STATEU set the state of the digital output node as follows:

0 = deactivated

1 = activated

DOUT1.STATEU and DOUT2.STATEU are used when DOUT1.MODE AND DOUT2.MODE = 0 (user mode).

Related Topics

10.1.4 Digital Outputs

This page intentionally left blank.

DRV Parameters and Commands

DRV.ACC

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Describes the acceleration ramp for the velocity loop. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s ² , (custom units)/s ² , rad/s ² Linear: counts/s ² , mm/s ² , μm/s ² , (custom units)/s ² |
| Range | Note: The range and default values of (custom units)/s ² units depend on the values of PIN and POUT. The range and default values listed in this table are derived from the default values of PIN and POUT. Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s ² 0.155 to 4,166,666.752 (custom units)/s ² 0.012 to 5,235,987.968 rad/s ² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s ² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s ² 30.995*MOTOR.PITCH to 2,147,483.647*MOTOR.PITCH μm/s ² 0.155 to 2,147,483.647 (custom units)/s ² |
| Default Value | Note: The range and default values of (custom units)/s ² units depend on the values of PIN and POUT. The range and default values listed in this table are derived from the default values of PIN and POUT. Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s ² 833.333 (custom units)/s ² 1,047.2 rad/s ² Linear: 715,840,000.000 counts/s ² 166.714*MOTOR.PITCH mm/s ² 166,714.191*MOTOR.PITCH μm/s ² 833.571 (custom units)/s ² |
| Data Type | Float |
| See Also | DRV.DEC, UNIT.ACCLINEAR, UNIT.ACCROTARY |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3501h/0 | M_01-00-00-000 |
| Modbus | 216 (64-bit) | M_01-03-00-000 |

Description

Describes the acceleration ramp for the velocity central loop.

Related Topics

10.7.5 Limits

DRV.ACTIVE

| General Information | |
|----------------------|-------------------------------------|
| Type | R/O Parameter |
| Description | Reads the enable status of an axis. |
| Units | N/A |
| Range | 0, 1, 3 |
| Default Value | N/A |
| Data Type | Integer |
| See Also | DRV.EN,DRV.DISSOURCES |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 220 | M_01-03-00-000 |

Description

DRV.ACTIVE reads the enable status of an axis as follows:

- DRV.ACTIVE = 0 drive disabled
- DRV.ACTIVE = 1 drive enabled
- DRV.ACTIVE = 3 drive enabled and in dynamic brake mode

There is no state 2.

When the drive is in state 3, the drive display shows a blinking decimal point. Additionally, if the drive is in state 3 the Parameter Load/Save view does not allow you to download a parameter file.

If an axis is not enabled (DRV.ACTIVE is 0), but DRV.EN is 1 and the hardware enable is high, read the value of DRV.DISSOURCES to query the reason that the drive is not enabled.

Related Topics

3.2 Display Codes

10.9 Enable/Disable

DRV.BLINKDISPLAY

| General Information | |
|---------------------|---|
| Type | Command |
| Description | Causes the display to blink for 10 seconds. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.BLINKDISPLAY causes the drive display located on the front of the drive to blink for 10 seconds. This command allows the user to identify the drive that is currently communicating with WorkBench.

DRV.CLRFAULTHIST

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Clears the fault history log in the NV. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.FAULTHIST |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 222 | M_01-03-00-000 |

Description

DRV.CLRFAULTHIST clears the fault history from the nonvolatile memory of the drive. This command erases all faults returned by DRV.FAULTHIST.

DRV.CLRFAULTS

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Tries to clear all active faults in the drive. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.FAULTS, DRV.EN, DRV.DIS |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 224 | M_01-03-00-000 |

Description

When DRV.CLRFAULTS is sent, the drive will try to clear all active faults. When a fault occurs, the fault is registered in the drive fault handler. DRV.CLRFAULTS clears the fault from the drive fault handler. However, if the fault still exists in the system, DRV.CLRFAULTS fails and the fault is re-registered in the fault handler.

If the DRV.CLRFAULTS succeeds, then the reply to DRV.FAULTS states that no faults exist. If the condition that triggered the fault is still present, the fault condition will remain. See 17.1 [Fault and Warning Messages](#) for details regarding the behavior of individual faults.

Note that executing a drive disable (DRV.DIS) followed by a drive enable (DRV.EN) has the same effect as executing DRV.CLRFAULTS.

Related Topics

17.2 [Clearing Faults](#)

DRV.CMDDELAY

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Issues a delay before next command is executed. |
| Units | ms |
| Range | 0 to 5,000 ms |
| Default Value | 0 ms |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-03-00-000 |

Description

This parameter is used when drive commands are used in a script and a delay is needed between the execution of two consecutive commands. DRV.CMDDELAY creates a delay in the execution of drive commands. In the period of time specified, no commands are executed. This feature is especially useful for command buffers.

Example

If the script is:

```
DRV.EN
IL.CMDU 0.1
```

then DRV.CMDDELAY is used between the two entries to delay execution 5 ms until the drive is enabled:

```
DRV.EN
DRV.CMDDELAY 5
IL.CMDU 0.1
```

Related Topics

10.2 Command Buffer

DRV.CMDSOURCE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the command source (service, fieldbus, analog input, gearing, digital, or Bode). |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | DRV.OPMODE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 226 | M_01-03-00-000 |

Description

DRV.CMDSOURCE specifies the source of the command to the drive. DRV.OPMODE sets the operation mode to the relevant control loop.

DRV.CMDSOURCE values can be set as follows:

| Value | Description |
|-------|-------------------------|
| 0 | Service, TCP/IP command |
| 1 | Fieldbus command |
| 2 | Gearing command |
| 3 | Analog command |

DRV.CMDSOURCE can be changed while the drive is enabled or disabled. If you use the terminal to change the operation mode, then it is recommended that you disable the drive before changing the command source.

⚠ WARNING If you change DRV.CMDSOURCE from the terminal while the drive is enabled, the system may experience a step change in demand.

Example

To set the command source to the TCP/IP channel and the operation mode to velocity:

```
-->DRV.CMDSOURCE 0
```

```
-->DRV.OPMODE 1
```

Related Topics

11 Using Command Source and Operating Modes

DRV.CRASHDUMP

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Retrieves diagnostic information after the drive crashes. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

Drives rarely crash, but if a crash occurs, information that can help diagnose the cause of a crash is saved to the nonvolatile (NV) memory within the drive. After the drive is restarted, you can use the DRV.CRASHDUMP command to retrieve this diagnostic information, which can be emailed to Kollmorgen for further support.

If the drive crashes (display flashes an F and three bars), it saves the diagnostic information to a specific block of the drive NV memory. The DRV.CRASHDUMP command then prints the diagnostic information from this NV memory block. Subsequent crash conditions will overwrite the NV memory block. Since the NV memory block is overwritten, but never erased, the DRV.CRASHDUMP command always shows the diagnostic information for the most recent crash.

DRV.DBILIMIT

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the maximum amplitude of the current for dynamic braking. |
| Units | Arms |
| Range | 0 to minimum of drive peak current (DRV.IPEAK) and motor peak current (MOTOR.IPEAK). |
| Default Value | Minimum of drive continuous current (DRV.ICONT) and motor continuous current (MOTOR.ICONT). |
| Data Type | Float |
| See Also | DRV.DISMODE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3444h/0 | M_01-00-00-000 |
| Modbus | 228 | M_01-03-00-000 |

Description

This parameter sets the maximum amplitude of the current for dynamic braking.

Example

Setting DRV.DBILIMIT to 2 limits the dynamic brake current to 2 Arms.

Related Topics

10.10 Controlled Stop

CS Parameters

10.11 Dynamic Braking

DRV.DEC

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the deceleration value for the velocity loop. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s ² , (custom units)/s ² , rad/s ² Linear: counts/s ² , mm/s ² , μm/s ² , (custom units)/s ² |
| Range | Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s ² 0.155 to 4,166,666.752 (custom units)/s ² 0.012 to 5,235,987.968 rad/s ² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s ² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s ² 30.994*MOTOR.PITCH to 833,333,333.333*MOTOR.PITCH μm/s ² 0.155 to 4,166,666.667 (custom units)/s ² |
| Default Value | Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s ² 833.333 (custom units)/s ² 1,047.2 rad/s ² Linear: 715,840,000.000 counts/s ² 166.71*MOTOR.PITCH4MOTOR.PITCH mm/s ² 166,714.191*MOTOR.PITCHMOTOR.PITCH μm/s ² 833.571 (custom units)/s ² |
| Data Type | Float |
| See Also | DRV.ACC, UNIT.ACCROTARY, UNIT.ACCLINEAR, DRV.OPMODE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3522h/0 | M_01-00-00-000 |
| Modbus | 230 (64-bit) | M_01-03-00-000 |

Description

DRV.DEC sets the deceleration value for the velocity loop command (VL.CMDU) and for the analog velocity command (AIN.VALUE). The operation mode (DRV.OPMODE) must be set to velocity mode for this command to function.

Related Topics

- 10.10 Controlled Stop
- 10.7.5 Limits

DRV.DIR

| General Information | |
|----------------------|--------------------------|
| Type | R/W Parameter |
| Description | Changes drive direction. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

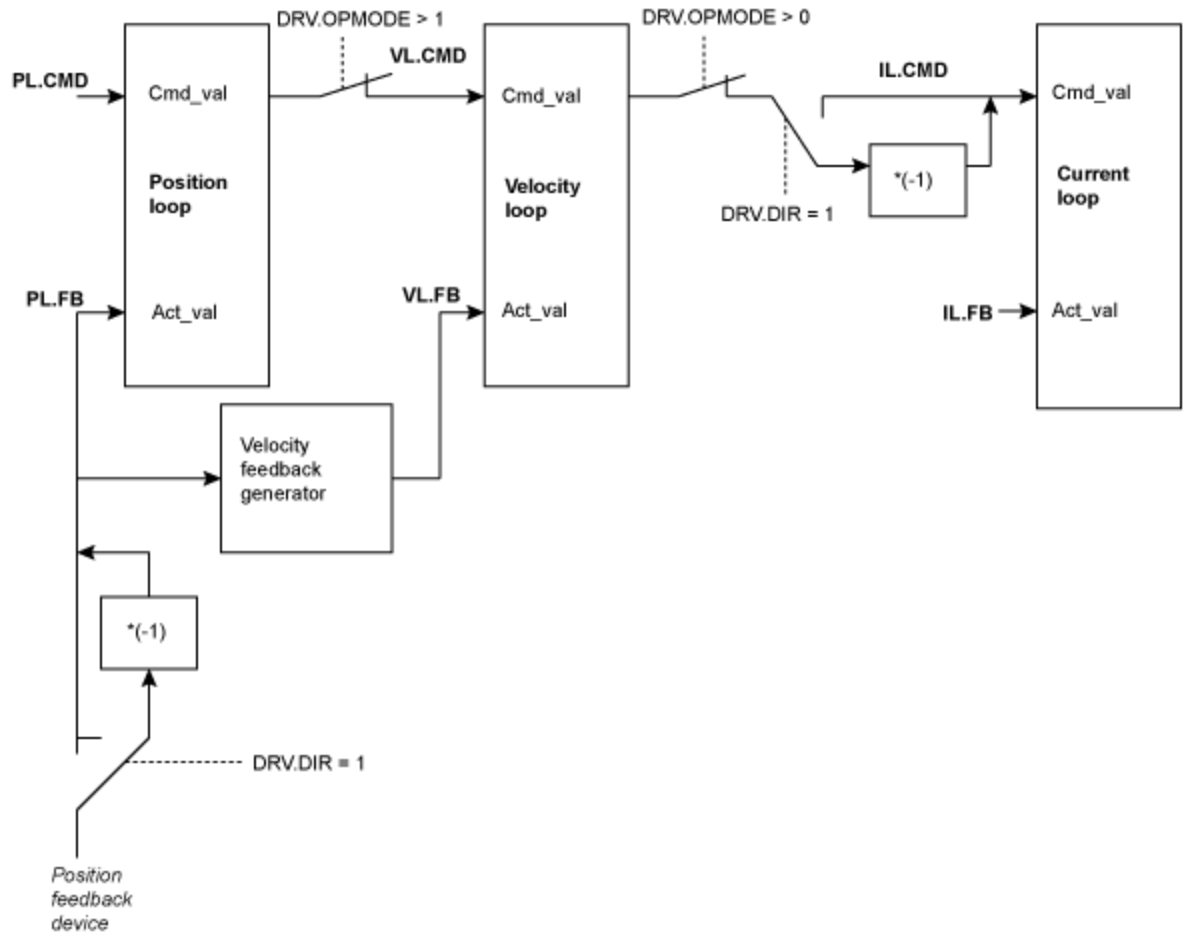
| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 352Ah/0 | M_01-00-00-000 |
| Modbus | 234 | M_01-03-00-000 |

Description

DRV.DIR changes the direction of the motor by changing the algebraic sign of the current command and position feedback value according to the figure below.

Note the following when using DRV.DIR:

- You can only change the DRV.DIR command when the drive is disabled.
- The drive status changes to "Axis not homed" as soon as the DRV.DIR parameter changes value (see DRV.MOTIONSTAT).
- You must verify the settings of the hardware limit switches. If necessary, switch the positive and negative hardware limit switches by swapping the wires at the digital inputs.



DRV.DIS

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Disables the axis (software). |
| Units | N/A |
| Range | N/A |
| Default Value | Analog drive software enabled. All other types of drive software disabled. |
| Data Type | N/A |
| See Also | DRV.EN, DRV.DISSOURCES, DRV.ACTIVE, DRV.DISMODE, DRV.DISTO |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3443h/0 | M_01-00-00-000 |
| Modbus | 236 | M_01-03-00-000 |

Description

DRV.DIS issues a software disable to the drive. The method by which the drive will be disabled (either immediately or with a ramp down first) is controlled by DRV.DISMODE.

By querying the value of DRV.ACTIVE, you can check whether the drive is currently enabled or disabled.

By querying the value of DRV.DISSOURCES, you can check whether the software enable bit is high (software enabled was issued by executing DRV.EN) or the software enable bit is low (software disable was issued by executing DRV.DIS).

If DRV.DIS is commanded the emergency timeout is started. If the drive does not disable or activate dynamic brake within DRV.DISTO, fault "F703" (=> p. 248) is reported.

Related Topics

17.2 Clearing Faults

10.10 Controlled Stop

10.2 Command Buffer

DRV.DISMODE

| General Information | |
|---------------------|---|
| Type | NV Parameter |
| Description | Selects among disable immediately or stop and then disable options. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | DRV.DBILIMIT ,DRV.DISTO, CS.VTHRESH |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35FFh/0 | M_01-00-00-000 |
| Modbus | 238 | M_01-03-00-000 |

Description

DRV.DISMODE sets the drive reaction to a DRV.DIS command.

NOTE

You must disable the drive in order to set DRV.DISMODE.

| Value | Behavior |
|-------|---|
| 0 | Disable axis immediately. |
| 1 | Use dynamic brake to ramp down. The drive remains in the dynamic brake state after the motor has stopped. The drive is disabled in the sense that it does not close the control loop and cannot perform a motion, but PWM stays active. |
| 2 | Use a controlled stop to ramp down and then disable the drive. |
| 3 | Use a controlled stop to ramp down, and then use dynamic brake. The drive remains in the dynamic brake state after the motor has stopped. The drive is disabled in the sense that it does not close the control loop and cannot perform a motion, but PWM stays active. |

In all cases described above, if a brake is configured (MOTOR.BRAKE), the brake closes if VL.FB drops below CS.VTHRESH.

⚠ WARNING Be careful with vertical loads when modifying this parameter. Coordinate this parameter's correct setting properly with the drive brake settings. If these settings are not coordinated, then vertical loads may have no stopping or holding force when the drive is disabled and the load could fall.

Related Topics

10.10 Controlled Stop

CS Parameters

10.11 Dynamic Braking

DRV.DISSOURCES

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Returns the possible reason for a drive disable. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.ACTIVE, DRV.FAULTS, DRV.EN, DRV.DIS |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 240 | M_01-03-00-000 |

Description

DRV.DISSOURCES is a bitwise parameter that returns the status of possible causes of a drive disable. If this parameter is 0, then the drive is enabled.

The return value specific bits are as follows:

| Bit | Status and Response |
|-----|--|
| 0 | Software disable (execute DRV.EN to issue software enable) |
| 1 | Fault exists (read DRV.FAULTS to get the active faults) |
| 2 | Hardware disable (remote enable input is low) |
| 3 | In-rush disable (the in-rush relay is opened) |
| 4 | Initialization disable (the drive did not finish the initialization process) |
| 5 | Controlled stop disable from a digital input. |

Related Topics

10.10 Controlled Stop

DRV.DISTO

| General Information | |
|----------------------|----------------------------|
| Type | R/W Parameter |
| Description | Sets the emergency timeout |
| Units | ms |
| Range | 0 to 120,000 ms |
| Default Value | 1,000 ms |
| Data Type | U32 |
| See Also | DRV.DIS, DRV.DISMODE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3445h/0 | M_01-00-00-000 |
| Modbus | 242 | M_01-03-00-000 |

Description

This timer starts when DRV.DIS is issued (regardless of the DRV.DIS origin). After this timeout elapses, the actual state of the drive is compared to the DRV.DISM MODE setting. If the actual state does not match the DRV.DISM MODE setting, a fault is reported and the hardware immediately executes the DRV.DISM MODE setting (for instance, disable or activate dynamic brake). Setting DRV.DISTO to 0 will disable the timeout.

Related Topics

10.10 Controlled Stop

DRV.EMUEDIR

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the direction of the emulated encoder output (EEO) signal. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | N/A |
| See Also | DRV.EMUEMODE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3493h/0 | M_01-00-00-000 |
| Modbus | 244 | M_01-03-00-000 |

Description

This parameter allows the user to change the direction of the emulated encoder output. DRV.DIR also affects the output direction (through an XOR, "exclusive or", operation). The drive uses DRV.DIR and DRV.EMUEDIR to decide the direction of the emulated encoder output. If DRV.DIR and DRV.EMUEDIR have the same value, then DRV.EMUEDIR is set to 0 (meaning an increase in the motor feedback will result an increase of the encoder emulation output and vice-versa). If these parameters have different values, then DRV.EMUEDIR is set to 1 (meaning an increase in the motor feedback will result in a decrease of the encoder emulation output and vice-versa).

DRV.EMUEMODE

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the mode of the emulated encoder output (EEO) connector. |
| Units | N/A |
| Range | 0 to 5 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | DRV.EMUERES, DRV.EMUEZOFFSET, DRV.EMUEMTURN |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3534h/0 | M_01-00-00-000 |
| Modbus | 246 | M_01-03-00-000 |

Description

This parameter sets the EEO connector to act as either an input or output as follows:

| Setting | Function |
|---------|---|
| 0 | Input (No EEO Output). The EEO connector (X9) is configured as an input. See Feedback 2 (FB2) for definitions for the EEO Connector (X9). Note that this mode is also deprecated and will behave as it did with firmware versions prior to M_01-03-00-000. In earlier firmware versions, this mode indicates that the EEO connector is not operative. |
| 1 | Output, with once per rev index pulse. |
| 2 | Output, with absolute index pulse. |
| 3 | Input, A/B signals. |
| 4 | Input, step and direction signals. |
| 5 | Input, up-down signals. |
| 6 | Output, with once per rev index and Input, step and direction. |
| 7 | Output, with absolute index and Input, step and direction |

Related Topics

8.4.1 Encoder Emulation

DRV.EMUEMTURN

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Defines the location of the index pulse on the EEO (emulated encoder output) when DRV.EMUEMODE=2. |
| Units | counts |
| Range | 0 to 4,294,967,295 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | DRV.EMUEMODE, DRV.EMUERES |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3491h/0 | M_01-00-00-000 |
| Modbus | 248 | M_01-03-00-000 |

Description

If the EEO mode with absolute index is selected (DRV.EMUEMODE=2), the index will be output when the feedback position (in counts) matches this parameter.

Related Topics

8.4.1 Encoder Emulation

DRV.EMUERES

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the resolution of the EEO (emulated encoder output). |
| Units | lines/rev (when DRV.EMUEMODE = 1, 2, or 3) counts/rev (when DRV.EMUEMODE = 4 or 5) |
| Range | 0 to 16,777,215 lines per revolution |
| Default Value | 0 lines per revolution |
| Data Type | Integer |
| See Also | DRV.EMUEMODE |
| Start Version | M_01-00-00-000 (resolution increased from 65,535 to 16,777,215 in M_01-04-00-000) |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3535h/0 | M_01-00-00-000 |
| Modbus | 250 | M_01-03-00-000 |

Description

This parameter sets the emulated encoder (EEO) resolution. DRV.EMUERES also defines how many lines are output for one revolution of the primary feedback (when this port is configured as an output), or how many lines will be considered a full revolution of the handwheel (when this port is configured as an input).

Related Topics

8.4.1 Encoder Emulation

DRV.EMUEZOFFSET

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the location of the EEO (emulated encoder output) index pulse (when DRV.EMUEMODE=1). |
| Units | 1/65536 rev |
| Range | 0 to 65535 rev |
| Default Value | 0 rev |
| Data Type | Integer |
| See Also | DRV.EMUEMODE, DRV.EMUEMTURN |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3537h/0 | M_01-00-00-000 |
| Modbus | 252 | M_01-03-00-000 |

Description

When EEO multitem is selected (DRV.EMUEMODE=1), this parameter sets a position of the EEO index. When the primary feedback position (within a revolution) equals this value, an index pulse will output.

Related Topics

8.4.1 Encoder Emulation

DRV.EN

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Enables the axis (software). |
| Units | N/A |
| Range | N/A |
| Default Value | Analog drive software is enabled. All other types of drive software are disabled. |
| Data Type | N/A |
| See Also | DRV.DIS, DRV.DISSOURCES DRV.ACTIVE |
| Start Version | M_01-00-00-000 |

Description

DRV.EN issues a software enable to the drive. You can query the value of DRV.ACTIVE to check whether the drive is currently enabled or disabled.

You can also query the value of DRV.DISSOURCES to check whether the software enable bit is high (software enabled was issued by executing DRV.EN) or the software enable bit is low (software disable was issued by executing DRV.DIS). If the drive software enable bit is low and DRV.EN is executed, then drive faults are automatically cleared during the software enable process.

Related Topics

- 17.2 Clearing Faults
- 10.2 Command Buffer
- 10.9 Enable/Disable

DRV.ENDEFAULT

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the default state of the software enable. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 256 | M_01-03-00-000 |

Description

DRV. ENDEFAULT sets the default state of the software enable on power up for drives without fieldbuses (DRV.CMDSOURCE other than 1).

Related Topics

10.9 Enable/Disable

DRV.FAULTHIST

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the last 10 faults from NV memory. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.FAULTS, DRV.CLRFAULTHIST |
| Start Version | M_01-00-00-000 |

Description

DRV.FAULTHISTORY returns the last 50 faults that occurred in the drive. The faults are shown with their fault number (which matches the one displayed on the drive display) and a timestamp that indicates when they last occurred.

Issue a DRV.CLRFAULTHIST to clear this fault log.

DRV.FAULTS

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the active faults. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.CLRFAULTS, DRV.FAULTHIST, DRV.CLRFAULTHIST |
| Start Version | M_01-00-00-000 |

Description

DRV.FAULTS returns a list of all currently active faults in the system, preceded by their fault number which matches the number displayed on the drive display.

To clear the faults, either issue a DRV.CLRFAULTS or issue a DRV.DIS followed by DRV.EN.

If no active faults are in the system, then after executing DRV.CLRFAULTS the value read by DRV.FAULTS is "No faults active".

Example

```
-->DRV.FAULTS
502: Bus under voltage.
-->
```

DRV.HANDWHEEL

| General Information | |
|----------------------|----------------------------|
| Type | R/O Parameter |
| Description | Reads the EEO input value. |
| Units | 1/4,294,967,296 rev |
| Range | 0 to 4,294,967,295 rev |
| Default Value | 0 rev |
| Data Type | Integer |
| See Also | DRV.EMUERES, DRV.EMUEMODE |
| Start Version | M_01-00-00-000 |
| End Version | M_01-03-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 2050h/0 | M_01-00-00-000 |
| Modbus | 258 | M_01-03-00-000 |

Description

When the EEO is selected as an input (DRV.EMUEMODE=3,4,5), this parameter reads the EEO value (where 4,294,967,296 is a full revolution, then the value rolls over). DRV.EMUERES defines the how many counts constitute a revolution on the EEO. This parameter represents the feedback 2 positions when feedback 2 is configured to be active.

When secondary feedback is selected (DRV.EMUEMODE is 0 and FB2.SOURCE = 1 (X9), or FB2.SOURCE = 2 (X7)), this parameter represents the secondary feedback position (where 4,294,967,296 is a full revolution, then the value rolls over). FB2.ENCRESES defines how many counts define a revolution for the secondary feedback.

Related Topics

8.4.1 Encoder Emulation

DRV.HELP

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the minimum, maximum, and default values for a specific parameter or command. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

This parameter returns more information about a specific parameter or command.

In most cases, except special parameters, this command tells you the minimum, maximum, default, and actual value of a parameter. Exceptions are commands that do not have these values (such as DRV.EN) or information commands (such as DRV.VER).

Related Topics

16.1 Terminal

DRV.HELPALL

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Retrieves the minimum, maximum, default, and actual values for all available parameters and commands. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

This parameter retrieves all information about all parameters and commands in the firmware. In most cases, DRV.HELPALL returns the minimum, maximum, default, and actual value for each parameter and command. Exceptions include parameters and commands that do not have these values (such as DRV.EN) or pure INFO commands (such as DRV.VER).

Related Topics

16.1 Terminal

DRV.HWENMODE

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Selects the action that the hardware enable digital input will perform. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-03-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3506h/0 | M_01-03-00-000 |
| Modbus | 260 | M_01-03-00-000 |

Description

This parameter selects the action that the hardware enable digital input will perform.

0 = The rising edge of the hardware enable will clear the drive faults.

1 = The rising edge of the hardware enable will not clear the drive faults.

The high/low state of the hardware enable is always used to control the active enable state of the drive.

Related Topics

10.9 Enable/Disable

DRV.ICONT

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the continuous rated current value. |
| Units | Arms |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | DRV.IPEAK |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 262 | M_01-03-00-000 |

Description

DRV.ICONT returns the drive continuous rated current in Arms.

The value of the continuous current is read automatically on drive boot from the power EEPROM of the drive. This value cannot be modified.

DRV.INFO

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads general information about the drive. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.INFO returns general information about the drive.

Example

```
Danaher Motion - Digital Servo Drive
```

```
-----
```

```
Drive mode                : AKD-B00606-NAAN-0000
Continuous current        : 12.000 A
Peak current: 30.000 A
Mac address: 00.23.1B.00.00.16
FPGA size: 1600
Fieldbus type: Analog
Power type                 : MV - Medium Voltage
Product serial number: 16777215
Product manufacturing date code : 65535
Power board serial number   : 0
Power board manufacturing date code: 0
Control board serial number: 0
Control board manufacturing date code: 0
Option board serial number: 0
Option board manufacturing date code: 0
```


DRV.IPEAK

| General Information | |
|----------------------|-------------------------------------|
| Type | R/O Parameter |
| Description | Reads the peak rated current value. |
| Units | Arms |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | DRV.ICONT |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 264 | M_01-03-00-000 |

Description

DRV.IPEAK returns the drive peak rated current in Arms.

The value of the peak current is read automatically on drive boot from the power EEPROM of the drive. This value cannot be modified.

Related Topics

8.8 Foldback

8.3.9 Using Wake and Shake (WS)

DRV.IZERO

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the current that will be used during the DRV.ZERO procedure. |
| Units | Arms |
| Range | Drive peak current to 0 Arms |
| Default Value | 0 Arms |
| Data Type | Float |
| See Also | DRV.ZERO |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 266 | M_01-03-00-000 |

Description

This parameter sets the current that is used during the DRV.ZERO procedure.

DRV.LIST

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the list of available parameters and commands. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.LIST reads the list of available commands and parameters from the drive.

To filter this list, enter DRV.LIST followed by the prefix of the commands and parameters that you wish to display.

Example

Return a list of all available commands in the system:

```
-->DRV.LIST
```

Return all commands with the prefix DRV:

```
-->DRV.LIST DRV
```

DRV.LOGICVOLTS

| General Information | |
|----------------------|---------------------------|
| Type | R/O Parameter |
| Description | Reads the logic voltages. |
| Units | mv , Ω |
| Range | N/A |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.LOGICVOLTS reads the logic voltages data of 1.2V, 2.5V, 3.3V, 5V, 12V, -12V, and 3.3AV.

Example

Below is an example of the output for this command:

```

ch0 = 1.2V      : 1211 mv
ch1 = 2.5V      :2488 mv
ch2 = 3.3V      :3274 mv
ch3 = 5V        :4950 mv
ch4 = 12V       :11892 mv
ch5 = -12V      : -11912 mv
ch6 = 3.3AV     :3300 mv
ch7 = R ohm     :100000 ohm

```

DRV.MEMADDR

| General Information | |
|----------------------|----------------------------------|
| Type | R/W Parameter |
| Description | Sets the read and write address. |
| Units | N/A |
| Range | N/A |
| Default Value | U8 |
| Data Type | N/A |
| See Also | DRV.MEMDATA |
| Start Version | M_01-00-00-000 |

Description

DRV.MEMADDR sets the address that is used by DRV.MEMDATA. The input can be either an internal parameter of the drive or any direct address from the DSP address space (SDRAM, internal RAM, or asynchronous memory). The input value can be either decimal or hexadecimal with 0x prefix.

Type extension can be one of the following:

U8,S8,U16,S16,U32,S32,U64,S64.

Examples

Setting to an internal parameter:

```
-->DRV.MEMADDR CCommandHandler.Debug1
```

Setting to an internal address:

```
-->DRV.MEMADDR 0xffabcde.u16
```

DRV.MEMDATA

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets or reads a value from an internal address. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.MEMADDR |
| Start Version | M_01-00-00-000 |

Description

DRV.MEMDATA reads a value from the address that was set by DRV.MEMADDR or writes a value to this address. The input value can be either decimal or hexadecimal with 0x prefix.

Examples

Read a value from internal address:

```
-->DRV.MEMDATA 01
```

Write a hexadecimal value to an internal address:

```
-->DRV.MEMADDR 0x01
```

DRV.MOTIONSTAT

| General Information | |
|---------------------|---------------------------------------|
| Type | R/O Parameter |
| Description | Reads the motion status of the drive. |
| Units | N/A |
| Range | 0 to 4,294,967,295 |
| Default Value | N/A |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3492h/0 | M_01-00-00-000 |
| Modbus | 268 | M_01-03-00-000 |

Description

This command indicates the current status of the drive internal motion (see table below).

| Bit | Significance | Description |
|-----|--------------|--|
| 0 | 0x00000001 | Motion task is active (high active) |
| 1 | 0x00000002 | Home position found /reference point set (high active) |
| 2 | 0x00000004 | Home routine finished (high active). Bits 1 and 2 both must be set to confirm that the homing process is complete. |
| 3 | 0x00000008 | Homing active (high active) |
| 4 | 0x00000010 | Homing error condition has occurred (high active)* |
| 5 | 0x00000020 | Slave in electronic gearing mode synchronized (high active) |
| 6 | 0x00000040 | Electronic gearing is active (high active) |
| 7 | 0x00000080 | Emergency stop procedure in progress (high active) |
| 8 | 0x00000100 | Emergency stop procedure has an error (high active) |
| 9 | 0x00000200 | Service motion active (high active) |
| 10 | 0x00000400 | A motion task could not be activated /invalid MT (high active)** |
| 11 | 0x00000800 | Motion task target position has been reached. See also MT.TPOSWND (high active). |
| 12 | 0x00001000 | Motion task target velocity has been reached. See also MT.TVELWND (high active). |
| 13 | 130x00002000 | Motion task encountered an exception. A motion task exception can happen during a static motion task activation, or during activation of motion task on the fly (when velocity is not zero). The status bit will be reset automatically on successful activation of any motion, or by a command DRV.CLRFAULT. |
| 14 | 0x00004000 | The target position of a motion task has been crossed. This situation occurs for motion tasks with a change on the fly when triggering the DRV.STOP command just before the reaching the target velocity of the current active motion task. The ramp-down procedure with the motion task deceleration ramp causes the target position to be crossed (high active). |

* A possible error condition for homing to a reference switch could be that no reference switch was found between two hardware limit switches.

** A possible error condition for an invalid motion task could be that a motion task tried to trigger automatically following motion task that has never been initialized (called an "empty motion" task).

Related Topics

Drive Motion Status

DRV.NAME

| General Information | |
|----------------------|---------------------------------------|
| Type | NV Parameter |
| Description | Sets and reads the name of the drive. |
| Units | N/A |
| Range | N/A |
| Default Value | No-Name |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

You can assign a unique name to any drive. This name is one way to identify the drive in a multiple drive network (for instance, in a TCP/IP network on which multiple drives reside).

From the terminal screen, DRV.NAME returns the name of the drive as ASCII characters.

DRV.NVLIST

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Lists the NV parameters and values from the RAM. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.NVLIST lists all the drive parameters that reside in NV memory.

The list includes each parameter name, followed by its current value from the RAM.

DRV.NVLOAD

| General Information | |
|---------------------|---|
| Type | W/O Parameter |
| Description | Loads all data from the NV memory of the drive into the RAM parameters. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.NVLOAD DRV.NVLIST |
| Start Version | M_01-00-00-000 |

Description

DRV.NVLOAD loads all data from the NV memory of the drive into the RAM parameters.

DRV.NVSAVE

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Saves the drive parameters from the RAM to the NV memory. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.RSTVAR |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|------------------------------|----------------------|
| EtherCAT COE and CANopen | Object Dictionary 35EBh/0 | M_01-00-00-000 |
| Modbus | N/A | N/A |

Description

DRV.NVSAVE saves the current drive parameter values from the RAM to the NV memory.

The drive parameters that were saved to the NV are read from the NV on the next drive boot, causing the values to be automatically set to the saved values on every drive boot.

Executing DRV.RSTVAR does not modify the values of the NV, but instead sets the drive values in RAM to their defaults.

DRV.ONTIME

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Returns how long the drive has been running since last power up. |
| Units | Days:Hours:Minutes:Seconds |
| Range | N/A |
| Default Value | N/A |
| Data Type | String |
| See Also | Returns how long the drive has been running since first activated. |
| Start Version | M_01-00-00-000 |

Description

This parameter returns the length of time that the drive has been running for the current session (since the last power up).

DRV.OPMODE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the drive operation mode (current, velocity, or position). |
| Units | N/A |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | DRV.CMDSOURCE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35B4h/0 | M_01-00-00-000 |
| Modbus | 270 | M_01-03-00-000 |

Description

DRV.OPMODE specifies the operation mode of the drive. You must also use DRV.CMDSOURCE to set the source of the command to the drive.

The operation mode values can be set as follows:

| Mode | Description |
|------|---------------------------------|
| 0 | Current (torque) operation mode |
| 1 | Velocity operation mode |
| 2 | Position operation mode |

DRV.OPMODE can be changed while the drive is enabled or disabled. If you are using the terminal to change the operation mode, then it is recommended that you disable the drive before changing the operation mode. If you change the operation mode from the terminal while the drive is enabled, the system may experience a step change in demand.

Example

Set the source of the command to a TCP/IP channel and the desired operation mode to velocity:

```
-->DRV.CMDSOURCE 0
-->DRV.OPMODE 1
```

Related Topics

- 11 Using Command Source and Operating Modes
 - 10.10 Controlled Stop
 - 11.3 Current Loop
 - 10.1 Digital Inputs and Outputs
 - 11.4 Velocity Loop
 - 11.5 Position Loop
- 8.3.9 Using Wake and Shake (WS)

DRV.READFORMAT

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the value returned to either decimal or hexadecimal. |
| Units | N/A |
| Range | 10 or 16 |
| Default Value | 10 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.READFORMAT sets the return values type to either decimal or hexadecimal.

| Format | Description |
|--------|--|
| 10 | Sets the read values to decimal format |
| 16 | Sets the read values to hexadecimal format |

DRV.RSTVAR

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Sets default values in the drive without re-booting the drive and without resetting the NV memory. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 272 | M_01-03-00-000 |

Description

DRV.RSTVAR causes the drive to return to the default values without the need to re-boot the drive first and without resetting the NV memory. Use DRV.RSTVAR to return to the default settings and recover a working drive.

DRV.RUNTIME

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Returns how long the drive has been running since first activated. |
| Units | Days:Hours:Minutes:Seconds |
| Range | N/A |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.RUNTIME returns the length of time that the drive has been running since it was first activated. This time includes the current session and the total amount of time from all previous sessions.

DRV.SETUPREQBITS

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the bitwise set status of parameters that must be set before the drive can be enabled |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.SETUPREQLIST, MOTOR.AUTOSSET |
| Start Version | M_01-00-00-000 |

Description

This parameter returns the bitwise set status of parameters that needs to be set up before the drive can be enabled. Only when this parameter returns 0 can the drive be enabled.

| Parameter | Bits |
|-------------|------------|
| IL.KP | 0x00000001 |
| MOTOR.IPEAK | 0x00000002 |
| MOTOR.ICONT | 0x00000004 |
| MOTOR.VMAX | 0x00000008 |
| MOTOR.POLES | 0x00000010 |
| MOTOR.PHASE | 0x00000020 |

Please note that if MOTOR.AUTOSSET is set to 1 (parameters automatically calculated from motor ID data), then all values in the list will be initialized from the feedback device. Otherwise, the parameters must be set manually.

Example

```
-->DRV.SETUPREQBITS
0
-->
```

DRV.SETUPREQLIST

| General Information | |
|----------------------|--|
| Type | R/O Parameters |
| Description | Reads the list of parameters that must be set before the drive can be enabled. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | DRV.SETUPREQBITS, MOTOR.AUTOSET |
| Start Version | M_01-00-00-000 |

Description

This parameter returns the list of parameters that must be set before the drive can be enabled and also whether or not each one of those parameters is set. Only when all commands have value of 0 can the drive be enabled.

Please note that if MOTOR.AUTOSET is set to 1 (parameters automatically calculated from motor ID data), then all values in the list will be initialized from the feedback device. Otherwise, the parameters must be set manually.

Example

```
-->DRV.SETUPREQLIST

IL.KP 0

MOTOR.ICONT 0

MOTOR.IPEAK 0

MOTOR.POLES 0

-->
```

DRV.STOP

| General Information | |
|---------------------|--------------------------------------|
| Type | Command |
| Description | This command stops all drive motion. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35FEh/0 | M_01-00-00-000 |
| Modbus | 274 | M_01-03-00-000 |

Description

This command stops all drive motion.

DRV.TEMPERATURES

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the temperature of drive components. |
| Units | °C |
| Range | 55 to 125 °C |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3610h/0 | M_01-00-00-000 |
| | 3611h/0 | |
| Modbus | N/A | M_01-03-00-000 |

Description

DRV.TEMPERATURES reads the temperature in different parts of the drive (power and control boards). The temperature is read from temperature sensors located in the drive.

Example

Below is an example of the output for this command :

```
Control Temperature: 39 °C
Power1 Temperature: 31 °C
Power2 Temperature: Sensor does not exist.
Power3 Temperature: Sensor does not exist.
```

DRV.TYPE

| General Information | |
|----------------------|--|
| Type | R/O on analog, EtherCAT and CANopen models R/W on the CC drive model. |
| Description | Selects the operational fieldbus on CC drive models. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 2 |
| Data Type | Integer |
| See Also | FBUS.TYPE, DRV.INFO |
| Start Version | M_01-03-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 276 | M_01-03-00-000 |

Description

DRV.TYPE allows you to select the operational fieldbus for your drive. This parameter is read-write on the CC drive models and read-only on the analog, EtherCAT, and CANopen drive models. To change the operational fieldbus of your drive:

- Set DRV.TYPE to one of the following values:
 - 0 = Analog (no EtherCAT or CANopen) with no position indexer functionality.
 - 1 = Analog (no EtherCAT or CANopen) with position indexer functionality.
 - 2 = EtherCAT
 - 3 = CANopen
- Save the parameters to the NV memory on the drive by issuing the DRV.NVSAVE command
- Power cycle the 24 V supply to the drive. When the drive has finished powering up, it will be working with the new selection.

Changing DRV.TYPE does not immediately change the type of the drive fieldbus selection. You must power cycle the drive to start the drive with the selected functionality.

You cannot use EtherCAT and CANopen at the same time. Use FBUS.TYPE or DRV.INFO to identify the fieldbus currently in use.

DRV.TYPE does not change if you use DRV.RSTVAR.

Related Topics

- 2 AKD Models

DRV.VER

| General Information | |
|----------------------|--------------------------|
| Type | R/O Parameter |
| Description | Reads the drive version. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.VER reads both FPGA and firmware versions.

The version data presented is hard coded in the firmware code.

Example

Below is an example of the output for this command:

```
Danaher Motion - Digital Servo Drive
-----
FPGA version : FP0004_0001_00_07
Firmware Version : M_0-0-15_T_2009-01-19_10-36-28_IR
```

DRV.VERIMAGE

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Returns the version data from each image. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.VERIMAGE reads the versions of the different images in the drive. This parameter returns the version data from each image .i00 file.

Example

Below is an example of the output for this parameter:

```
Danaher Motion - Digital Servo Drive
-----
Resident Firmware: R_0-0-11
Operational Firmware: M_0-0-15
Resident FPGA: FPB004_0001_00_07
Operational FPGA : FP0004_0001_00_07
```

DRV.WARNINGS

| General Information | |
|---------------------|----------------------------|
| Type | R/O Parameter |
| Description | Reads the active warnings. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

DRV.WARNINGS returns a list of all currently active warnings in the system.

DRV.ZERO

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the zero mode. The procedure is activated when the drive is enabled. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | DRV.IZERO |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 278 | M_01-03-00-000 |

Description

The zero procedure is a sequence in which phase commutation is initialized. During this procedure, the motor is held at a certain known electrical position (by applying a current defined by DRV.IZERO). After the motor rests at this position, the commutation angle is calculated and set automatically.

FB1 Parameters

FB1.BISSBITS

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Specifies the number of Biss Sensor (Position) Bits for the BiSS Mode C encoder in use. |
| Units | bits |
| Range | 0 to 64 bits |
| Default Value | 32 bits |
| Data Type | Integer |
| See Also | FB1.SELECT, FB1.IDENTIFIED |
| Start Version | M_01-01-00-100 and M_01-01-03-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 280 | M_01-03-00-000 |

Description

FB1.BISSBITS specifies the number of BiSS sensor (position) bits for the BiSS Mode C encoder in use. Typically the value is either 26 or 32 for a BiSS Mode C Renishaw encoder. The required value for this parameter is provided by the feedback device manufacturer for the particular device being used.

Related Topics

8.2 Feedback 1

FB1.ENCRES

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the resolution of the motor encoder. |
| Units | Encoder counts |
| Range | 0 to $2^{32}-1$ |
| Default Value | 1,024 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3533h/0 | M_01-00-00-000 |
| Modbus | 282 | M_01-03-00-000 |

Description

This parameter sets or gets the resolution of the motor encoder (encoder feedback systems only) in number of counts per revolution for a rotary motor and the number of encoder pitches per motor pole pitch for a linear motor. The number of encoder counts per revolution is obtained by multiplying the motor catalog resolution in units of PPR by four. For example, for a 1024 PPR resolution motor, the number of encoder counts per revolution is $1024 * 4 = 4096$. For this motor FB1.ENCRES must be set to 4096.

For linear motors, the value of FB1.ENCRES is set to the number of encoder pitches per motor pole pitch. For a motor with 32 mm pole pitch, and a $40 \mu\text{m}$ encoder pitch, the value for FB1.ENCRES should be set to $32 \text{ mm} / 40 \mu\text{m} = 800$.

Related Topics

8.2 Feedback 1

FB1.HALLSTATE

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the Hall switch values (encoder feedback only). |
| Units | Binary |
| Range | 0 0 0 to 1 1 1 |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

FB1.HALLSTATE reads the Hall switch values (encoder feedback only).

Related Topics

8.2 Feedback 1

FB1.HALLSTATEU

| General Information | |
|----------------------|-----------------------------------|
| Type | R/O Parameter |
| Description | Reads the state of Hall switch U. |
| Units | N/A |
| Range | 0 and 1 |
| Default Value | 1 |
| Data Type | Integer |
| See Also | FB1.HALLSTATE |
| Start Version | M_01-03-07-000 |

Description

FB1.HALLSTATEU reads the state of Hall switch U.

Related Topics

8.2 Feedback 1

FB1.HALLSTATEV

| General Information | |
|----------------------|-----------------------------------|
| Type | R/O Parameter |
| Description | Reads the state of Hall switch V. |
| Units | N/A |
| Range | 0 and 1 |
| Default Value | 1 |
| Data Type | Integer |
| See Also | FB1.HALLSTATE |
| Start Version | M_01-03-07-000 |

Description

FB1.HALLSTATEV reads the state of Hall switch V.

Related Topics

8.2 Feedback 1

FB1.HALLSTATEW

| General Information | |
|----------------------|-----------------------------------|
| Type | R/O Parameter |
| Description | Reads the state of Hall switch W. |
| Units | N/A |
| Range | 0 and 1 |
| Default Value | 1 |
| Data Type | Integer |
| See Also | FB1.HALLSTATE |
| Start Version | M_01-03-07-000 |

Description

FB1.HALLSTATEW reads the state of Hall switch W.

Related Topics

8.2 Feedback 1

FB1.IDENTIFIED

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the type of feedback device used by the drive/motor. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | Integer |
| See Also | FB1.SELECT |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 284 | M_01-03-00-000 |

Description

This parameter is set according to FB1.SELECT on drive power up if FB1.SELECT is not -1; otherwise the parameter value is read from the drive memory.

| Type | Description |
|------|---|
| 0 | Unknown |
| 10 | Incremental encoder with A/B Quad, marker pulse and Hall |
| 11 | Incremental encoder with A/B Quad, marker pulse and no Hall |
| 20 | Sine Encoder , with marker pulse and Hall |
| 21 | Sine encoder , with marker pulse & No Halls |
| 30 | EnDat 2.1 with Sine Cosine |
| 31 | EnDat 2.2 |
| 32 | BiSS with Sine Cosine |
| 33 | HIPERFACE |
| 34 | BiSS Mode C Renishaw |
| 40 | Resolver |
| 41 | SFD |

Related Topics

8.2 Feedback 1

FB1.INITSIGNED

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets initial feedback value as signed or unsigned. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | FB1.ORIGIN |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 286 | M_01-03-00-000 |

Description

This parameter sets whether the initial value of the feedback read from the feedback device will be set as a signed or as an unsigned value.

0 = Unsigned

1 = Signed

The drive internal process for the feedback initialization is as follows:

1. Reads the position feedback.
2. Adds the origin to the feedback.
3. Determines modulo from Step 2 by the actual feedback bits.
4. Sets the position feedback sign according to FB1.INITSIGNED.

Related Topics

8.2 Feedback 1

FB1.MECHPOS

| General Information | |
|----------------------|--------------------------------|
| Type | R/O Parameter |
| Description | Reads the mechanical position. |
| Units | counts |
| Range | 0 to 4,294,967,295 counts |
| Default Value | N/A |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 288 | M_01-03-00-000 |

Description

FB1.MECHPOS reads the mechanical angle which is equal to the lower 32 bits in the 64-bit position feedback word.

Related Topics

8.2 Feedback 1

FB1.MEMVER

| General Information | |
|----------------------|--------------------------------------|
| Type | R/O Parameter |
| Description | Returns the memory feedback version. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

FB1.MEMVER returns the memory feedback version (only applicable for feedbacks with memory).

Related Topics

8.2 Feedback 1

FB1.OFFSET

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets position feedback offset. |
| Units | Depends on UNIT.ACCROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, custom units, 16-bit counts Linear: counts, mm, μ m, custom units, 16-bit counts |
| Range | Rotary: -5,123,372,000,000,005.000 to 5,123,372,000,000,005.000 counts -7495.067 to 7495.067 rad -429,436.096 to 429,436.096 deg -5,964.390 to 5,964.390 [custom units] -78,176,452.637 to 78,176,452.636 16-bit counts Linear: -5,123,372,000,000,005.000 to 5,123,372,000,000,005.000 counts -1192.878*MOTOR.PITCH to 1192.878*MOTOR.PITCH mm -1192877.952*MOTOR.PITCH to 1192877.952*MOTOR.PITCH μ m -5964.390 to 5964.390 custom units -78176452.637 to 78176452.636 16-bit counts |
| Default Value | 0 |
| Data Type | Float |
| See Also | PL.FB |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3490h/0 | M_01-00-00-000 |
| Modbus | 290 (64-bit) | M_01-03-00-000 |

Description

FB1.OFFSET is a value added to the position feedback (PL.FB).

Example

If PL.FB is 10 deg and FB1.OFFSET is set to -10 deg, then the next read of PL.FB will return ~0 deg.

Related Topics

8.2 Feedback 1

FB1.ORIGIN

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Adds to the initial feedback position. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: counts, rad, deg, custom units, 16-bit counts Linear: counts, mm, µm, custom units, 16-bit counts |
| Range | Rotary: 0.000 to 5,123,372,000,000.000 counts 0.000 to 7,495.067 rad 0.000 to 429,436.096 deg 0.000 to 5,964.390 custom units 0.000 to 78,176,452.636 16-bit counts Linear: 0.000 to 5,123,372,000,000.000 counts 0.000 to 1,192.878 mm 0.000 to 1,192,877.952 µm 0.000 to 5,964.390 custom units 0.000 to 78,176,452.636 counts 16 Bit |
| Default Value | 0 counts |
| Data Type | Float |
| See Also | FB1.INITSIGNED |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3656h/0 | M_01-00-00-000 |
| Modbus | 294 (64-bit) | M_01-03-00-000 |

Description

FB1.ORIGIN is a value that is added to the feedback device position. Initial value and modulo are determined from the number of bits of the feedback:

Initial position value = (<feedback from device> + FB1.ORIGIN) modulo <number of feedback bits>

The number of feedback bits is set according to the feedback type. For memory feedbacks it is the number of feedback bits; for none memory it is always single turn.

The drive internal process for the feedback initialization is as follows:

1. Reads the position feedback.
2. Adds the origin to the feedback.
3. Determines modulo from Step 2 by the actual feedback bits.
4. Sets the position feedback sign according to FB1.INITSIGNED.

Example

This example uses UNIT.PROTARY set to 2 (degrees)

It also assumes that the drive is connected to a single turn feedback device with memory.

FB1.ORIGIN is set to 22 and saved into NV memory.

Drive boots and reads from feedback device position 340 degrees. According to the description section above, calculation will be:

$$(340 + 22) \text{ modulo } 360 = 2 \text{ degrees.}$$

Therefore the initial feedback value will be set to 2 degrees.

Related Topics

8.2 Feedback 1

FB1.PFIND

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | A procedure that allows the user to find the commutation angle for encoder feedback, which has no halls. |
| Units | NA |
| Range | 0, 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | FB1.PFINDCMDU |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 298 | M_01-03-00-000 |

Description

A procedure that allows the user to find the commutation angle for encoder feedback (which has no Halls).

Related Topics

8.2 Feedback 1

FB1.PFINDCMDU

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Current value used during the phase finding procedure (PFB.PFIND=1) |
| Units | A |
| Range | 0 to DRV.IPEAK |
| Default Value | 0 |
| Data Type | Float |
| See Also | PFB.PFIND |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 300 | M_01-03-00-000 |

Description

FB1.PFINDCMDU sets the current value used during the phase finding procedure.

Related Topics

8.2 Feedback 1

FB1.POLES

| General Information | |
|----------------------|-------------------------------------|
| Type | R/O Parameter |
| Description | Reads the number of feedback poles. |
| Units | N/A |
| Range | 2 to 128 |
| Default Value | 2 |
| Data Type | Integer |
| See Also | MOTOR.POLES |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 302 | M_01-03-00-000 |

Description

FB1.POLES sets the number of individual poles in the feedback device. This variable is used for the commutation function, as well as for velocity feedback scaling, and represents the number of individual poles (not pole pairs). The division value of motor poles (MOTOR.POLES) and feedback poles (FB1.POLES) must be an integer when moving drive to enable, otherwise a fault is issued.

Related Topics

8.2 Feedback 1

FB1.PSCALE

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets position scaling value for fieldbus transferred position objects. |
| Units | N/A |
| Range | 0 to 32 |
| Default Value | 20 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 304 | M_01-03-00-000 |

Description

Position values transferred over fieldbus are converted from native 64-bit values to a maximum 32-bit position value. This parameter sets the resolution/revolution of position values back to the controller.

FB1.PSCALE determines the counts per revolution of position values delivered by fieldbus. The default value is 20, which yields 2^{20} counts/revolution. This scaling is used for CAN PDOs 6064 (Position Actual Value) and 60F4 (Following Error Actual Value).

Example

The drive always works internally with 64-bit position values. The drive internal 64-bit actual position should contain the following value:

0x0000.0023.1234.ABCD

The lower 32 bits represent the mechanical angle of the feedback. The upper 32 bits represent the number of turns.

FB1.PSCALE = 20

The 32-bit position is: 0x0231234A

FB1.PSCALE = 16

The 32-bit position is: 0x00231234

Related Topics

8.2 Feedback 1

FB1.RESKTR

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the resolver nominal transformation ratio. |
| Units | N/A |
| Range | 0.001 to 50.000 |
| Default Value | 0.5 |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 306 | M_01-03-00-000 |

Description

This parameter sets the resolver nominal transformation ratio. It affects the resolver excitation output amplitude. The value can be obtained from the resolver data sheet.

Related Topics

8.2 Feedback 1

FB1.RESREFPHASE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the electrical degrees of phase lag in the resolver. |
| Units | electrical degrees |
| Range | -180 to 180° |
| Default Value | -2° |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 308 | M_01-03-00-000 |

Description

This parameter sets the electrical degrees of phase lag in the resolver.

See the motor resolver datasheet for the value for this parameter .

Related Topics

8.2 Feedback 1

FB1.SELECT

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets user entered type or identified type (-1). |
| Units | N/A |
| Range | -1, 10, 20, 30, 31, 32, 40, 41 |
| Default Value | -1 |
| Data Type | Integer |
| See Also | FB1.IDENTIFIED |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 353Bh/0 | M_01-00-00-000 |
| Modbus | 310 | M_01-03-00-000 |

Description

FB1.SELECT sets the feedback type manually (see FB1.IDENTIFIED) or allows the drive to automatically identify the feedback type on power up.

FB1.SELECT Input Values

| Input Value | Description |
|-------------|--|
| -1 | The drive automatically identifies the type of feedback as part of the power up process. Setting this value does not modify FB1.IDENTIFIED, unless it is saved in the NV memory for the next power up. If a feedback with memory is connected to the drive, the value of FB1.IDENTIFIED is set automatically to the feedback identified and all parameters read from the feedback are set according to the values read from the feedback. If no feedback is connected or a feedback with no memory is connected, the value of FB1.IDENTIFIED is set to 0 (no feedback identified) and all values normally read from the feedback are read from NV memory (if stored in NV) otherwise they are set to the default values. |
| 10 | Manually sets the type to incremental encoder. This input sets the value of FB1.IDENTIFIED to 10. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified). |
| 20 | Manually sets the type to sine encoder. This input sets the value of FB1.IDENTIFIED to 20. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified). |
| 30 | Manually sets the type to Endat 2.1. This input sets the value of FB1.IDENTIFIED to 30. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified). |
| 31 | Manually sets the type to Endat 2.2. This input sets the value of FB1.IDENTIFIED to 31. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified). |
| 32 | Manually sets the type to BiSS. This input sets the value of FB1.IDENTIFIED to 32. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified). |
| 33 | Manually sets the type to Hyperface. This input sets the value of FB1.IDENTIFIED to 33. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified). |

| Input Value | Description |
|-------------|--|
| 40 | Manually sets the type to resolver. This input sets the value of FB1.IDENTIFIED to 40. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified). |
| 41 | Manually sets the type to SFD. This input sets the value of FB1.IDENTIFIED to 41. If the feedback setting fails, FB1.IDENTIFIED is automatically set to 0 (no feedback identified). |

FB1.SELECT Feedback Types

| Type | Description |
|------|---|
| 0 | Unknown |
| 10 | Incremental encoder with A/B Quad, marker pulse and Hall |
| 11 | Incremental encoder with A/B Quad, marker pulse and no Hall |
| 20 | Sine Encoder , with marker pulse and Hall |
| 21 | Sine encoder , with marker pulse & No Halls |
| 30 | EnDat 2.1 with Sine Cosine |
| 31 | EnDat 2.2 |
| 32 | BiSS with Sine Cosine |
| 33 | HIPERFACE |
| 34 | BiSS Mode C Renishaw |
| 40 | Resolver |
| 41 | SFD |

Related Topics

8.2 Feedback 1

FB1.TRACKINGCAL

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Controls tracking calibration algorithm. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 312 | M_01-03-00-000 |

Description

This parameter turns the tracking calibration algorithm on or off for sine-cosine or resolver.

0 = Tracking calibration is off.

1 = Tracking calibration is on.

Related Topics

8.2 Feedback 1

FB2 Parameters

FB2.ENCRES

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the secondary feedback (FB2) resolution. |
| Units | counts/rev |
| Range | 0 to 262,140 counts/rev |
| Default Value | 0 |
| Data Type | Integer |
| See Also | FB2.MODE, FB2.SOURCE |
| Start Version | M_01-03-00-000 |

Description

This parameter sets the feedback 2 (FB2) resolution and defines how many counts input into the secondary feedback will be considered a full revolution.

Related Topics

8.4 Feedback 2

FB2.MODE

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the mode for the second feedback inputs, EEO connector (X9) and high speed opto inputs (pins 9 and 10 on X7). |
| Units | N/A |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | FB2.ENCRES, FB2.SOURCE, PL.FBSOURCE |
| Start Version | M_01-03-00-000 |

Description

This parameter sets the feedback 2 input type as follows:

0 = Input A/B Signals

1 = Input Step and Direction Signals

2 = Input, up-down signals

Related Topics

8.4 Feedback 2

FB2.SOURCE

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the source for the second feedback input. Choices are the EEO connectors (X9) which are RS485 inputs, or the X7 connector's high speed opto inputs (pins 9 and 10). |
| Units | N/A |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | FB2.ENCRESES, FB2.MODE, PL.FBSOURCE |
| Start Version | M_01-03-00-000 |

Description

This parameter sets the secondary feedback source to be either the EEO connector (X9) or the high speed opto inputs on the I/O Connector (X7) as follows:

0 = None

1 = Feedback Source X9 (EEO connector)

2 = Feedback Source X7 (High Speed Opto Inputs on the I/O Connector)

Related Topics

8.4 Feedback 2

FBUS Parameters

FBUS.PARAM1 TO FBUS.PARAM20

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Set fieldbus specific meanings. |
| Units | N/A |
| Range | See table below. |
| Default Value | See table below. |
| Data Type | Unsigned 32 |
| See Also | CANopen Communication Manual, EtherCAT Communication Manual |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---|---|----------------------|
| EtherCAT COE and CANopen | 36E5h/0 FBUS.PARAM01 36EAh/0 FBUS.PARAM06 | M_01-00-00-000 |
| | 36E6h/0 FBUS.PARAM02 36EBh/0 FBUS.PARAM07 | |
| | 36E7h/0 FBUS.PARAM03 36ECh/0 FBUS.PARAM08 | |
| | 36E8h/0 FBUS.PARAM04 36EDh/0 FBUS.PARAM09 | |
| | 36E9h/0 FBUS.PARAM05 36EEh/0 FBUS.PARAM10 | |
| EtherCAT COE and CANopen Range | EtherCAT COE | CANopen |
| | N/A | 125; 250; 500; 1000 |
| | FBUS.PARAM01 0 to 1 | 0 to 1 |
| | FBUS.PARAM02 1,001 to 65,535 | N/A |
| | FBUS.PARAM03 0 to 1 | 0 to 1 |
| | FBUS.PARAM04 0 to 1 | 0 to 1 |
| EtherCAT COE and CANopen Default Value | EtherCAT COE | CANopen |
| | N/A | 125 |
| | FBUS.PARAM01 1 | 0 |
| | FBUS.PARAM02 0 | N/A |
| | FBUS.PARAM03 1 | 0 |
| | FBUS.PARAM04 0 | 0 |
| Modbus | 314 FBUS.PARAM01 334 FBUS.PARAM11 | M_01-03-00-000 |
| | 316 FBUS.PARAM02 336 FBUS.PARAM12 | |
| | 318 FBUS.PARMA 338 FBUS.PARAM13 | |
| | 320 FBUS.PARAM04 340 FBUS.PARAM14 | |
| | 322 FBUS.PARAM05 342 FBUS.PARAM15 | |
| | 324 FBUS.PARAM06 344 FBUS.PARAM16 | |
| | 326 FBUS.PARAM07 346 FBUS.PARAM17 | |
| | 328 FBUS.PARAM08 348 FBUS.PARAM18 | |
| | 330 FBUS.PARAM09 350 FBUS.PARAM19 | |
| | 332 FBUS.PARAM10 352 FBUS.PARAM20 | |

Description

FBUS.PARAM01 sets the baud rate for the CANopen bus. Supported baud rates are 125, 250, 500 and 1000 kBaud.

FBUS.PARAM02 switches the phase locked loop (PLL) for synchronized use: 0 = OFF, 1 = ON

FBUS.PARAM03 sets the configured station alias for EtherCAT.

FBUS.PARAM04 switches the surveillance of SYNC-signals: 0 = OFF, 1 = ON

FBUS.PARAM05 is used to configure some special behaviors of the DS402.

FBUS.PARAM06 - FBUS.PARAM10 are reserved.

FBUS.PARAM04 Additional Notes

FBUS.PARAM04 enables (1) or disables (0) the synchronization supervision of the CANOpen or EtherCAT fieldbus.

Default values:

- CANOpen: disabled (0)
- EtherCAT: enabled (1)

The synchronization supervision is active when:

- FBUS.PARAM04 = 1
- The first CANOpen Sync message or first EtherCAT frame was received.

When more than three CANOpen sync messages or seven EtherCAT frames have not been received, and the drive is enabled, fault F125, "Synchronization lost", occurs.

FBUS.PARAM05 Additional Notes

Bit 0 configures the behavior of DS402 state machine in case of fault resets.

- Bit 0 = 1: Faults can only be reset using DS402 control word bit 7.
- Bit 0 = 0: The reset can also be done via telnet or digital input and the DS402 state machine reflects this condition.
- Bit 1 = 1: The state of the hardware enable does not change the state machine state Operation Enable.
- Bit 0 = 0: If the state Operation Enable or Switched on is active it falls back to the state switched On Disabled, if the Hardware enable goes to 0.
- Bit 2 = 1: Workbench/Telnet can not software enable the drive, when CAN/EtherCAT are Operational.
- Bit 0 = 0: Workbench/Telnet can software enable the drive.
- Bit 3 = 1: DS402-state machine is not influenced, if the software-enable is taken away via Telnet.
- Bit 0 = 0: DS402-state machine is influenced, if the software-enable is taken away via Telnet.

FBUS.PLLSTATE

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Returns the status of the PLL |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | FBUS.PARAM1 TO FBUS.PARAM20, Fieldbus documentation |
| Start Version | M_01-02-00-000 |

Description

FBUS.PLLSTATE returns the status of the phase locked loop (PLL). The PLL states are as follows:

| PLL State | Description |
|-----------------------------|--|
| PLL not activated | This state is set using FBUS.PARAM02. Not each fieldbus or operation mode needs synchronization. |
| PLL activated, but unlocked | The PLL is activated but has not yet been locked successfully. This state is related to the fieldbus master, as well as to the fieldbus mode of operation. |
| PLL activated and locked | PLL is fully operational and locked |

For more information, see the *AKD CANopen Manual*, Phase Locked Loop.

FBUS.PLLTHRESH

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets number of successful synchronized cycles needed to lock the PLL. |
| Units | N/A |
| Range | 0 to 10,000 |
| Default Value | 0 |
| Data Type | Integer, U32 |
| See Also | Appendix B: Fieldbus Manuals |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 354 | M_01-03-00-000 |

Description

This parameter sets number of successful synchronized cycles needed to lock the PLL.

FBUS.SAMPLEPERIOD

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets fieldbus sample period. |
| Units | Whole multiples of MTS 250 μ s |
| Range | 4 to 128 and value must be a power of 4 |
| Default Value | 32 = 2 ms |
| Data Type | U8 |
| See Also | Appendix B: Fieldbus Manuals |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | Profile specific SDOs/0 | M_01-00-00-000 |
| Modbus | 356 | M_01-03-00-000 |

Description

This parameter sets the fieldbus cycle time. It is normally written in the startup phase of the field busses via the object 60C2 subindex 1 (interpolation time units) and 2 (interpolation time index), where the index stands for a power of 10 seconds (for instance, -3 stands for milliseconds) and the units are the counts of these units. Kollmorgen recommends the following standard cycle rates, 250 μ s (4), 500 μ s (8), 1 ms (16), 2 ms (32), 4 ms (64).

FBUS.SYNCACT

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads actual distance from the desired sync distance. |
| Units | ns |
| Range | 0 to 250,000 ns |
| Default Value | 0 ns |
| Data Type | Integer, U 32 |
| See Also | Appendix B: Fieldbus Manuals |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 358 | M_01-03-00-000 |

Description

This parameter reads actual distance from the desired sync distance.

FBUS.SYNCDIST

| General Information | |
|----------------------|---------------------------------------|
| Type | NV Parameter |
| Description | Sets time target for synchronization. |
| Units | ns |
| Range | 0 to 250,000 ns |
| Default Value | 100,000 ns |
| Data Type | Integer, U32 |
| See Also | Appendix B: Fieldbus Manuals |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 360 | M_01-03-00-000 |

Description

This parameter sets time target for synchronization.

FBUS.SYNCWND

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets symmetrically arranged window around the desired sync distance. |
| Units | ns |
| Range | 0 to 1,000,000 ns |
| Default Value | 50,000 ns |
| Data Type | Integer, U2 |
| See Also | Appendix B: Fieldbus Manuals |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 362 | M_01-03-00-000 |

Description

This parameter sets symmetrically arranged window around the desired sync distance.

FBUS.TYPE

| General Information | |
|----------------------|---------------------------------|
| Type | R/O Parameter |
| Description | Shows the active fieldbus type. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | Fieldbus Manuals |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 364 | M_01-03-00-000 |

Description

FBUS.TYPE shows the active fieldbus type.

0 = No fieldbus available

1 = SynqNet

2 = EtherCAT

3 = CANopen

GEAR Parameters

GEAR.ACCMAX

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the maximum allowed acceleration value; active in opmode 2 (position) only. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s ² , (custom units)/s ² , rad/s ² Linear: counts/s ² , mm/s ² , μm/s ² , (custom units)/s ² |
| Range | Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s ² 0.155 to 4,166,666.752 (custom units)/s ² 0.012 to 5,235,987.968 rad/s ² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s ² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s ² 30.994*MOTOR.PITCH to 83,333,333.333*MOTOR.PITCH μm/s ² 0.155 to 4,166,666.667 (custom units)/s ² |
| Default Value | Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s ² 833.333 (custom units)/s ² 1,047.2 rad/s ² Linear: 715,840,000.000 counts/s ² 166.714*MOTOR.PITCHMOTOR.PITCH mm/s ² 166,714.191*MOTOR.PITCHMOTOR.PITCH μm/s ² 833.571 (custom units)/s ² |
| Data Type | Float |
| See Also | UNIT.ACCROTARY, UNIT.ACCLINEAR, GEAR.DECMAX |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 366 (64-bit) | M_01-03-00-000 |

Description

This parameter limits the acceleration of the slave to a numerical higher value.

Related Topics

10.7 Electronic Gearing

GEAR.DECMAX

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the maximum allowed deceleration value; active in opmode 2 (position) only. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s ² , (custom units)/s ² , rad/s ² Linear: counts/s ² , mm/s ² , μm/s ² , (custom units)/s ² |
| Range | Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s ² 0.155 to 4,166,666.752 (custom units)/s ² 0.012 to 5,235,987.968 rad/s ² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s ² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s ² 30.994*MOTOR.PITCH to 833,333,333.333*MOTOR.PITCH μm/s ² 0.155 to 4,166,666.667 (custom units)/s ² |
| Default Value | Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s ² 833.333 (custom units)/s ² 1,047.2 rad/s ² Linear: 715,840,000.000 counts/s ² 166.714*MOTOR.PITCHMOTOR.PITCH mm/s ² 166,714.191*MOTOR.PITCHMOTOR.PITCH μm/s ² 833.571 (custom units)/s ² |
| Data Type | Float |
| See Also | UNIT.ACCROTARY, UNIT.ACCLINEAR, GEAR.ACCMAX |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 370 (64-bit) | M_01-03-00-000 |

Description

This parameter limits the deceleration of the slave to a numerical higher value.

Related Topics

10.7 Electronic Gearing

GEAR.IN

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the denominator of the electronic gearing ratio; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 1 to 65,535 |
| Default Value | 1 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 374 | M_01-03-00-000 |

Description

This parameter sets the denominator of the gear ratio for the electronic gearing mode. The gear ratio is used in order to increase and decrease the slave velocity. The slave velocity can be calculated by the following formula:

Slave velocity = Master velocity * GEAR.OUT/GEAR.IN

Be sure that you set the external master source number of signals per revolution correctly. Also, select the gear ratio so that the maximum electronic gearing velocity (GEAR.VELMAX) is not exceeded.

Master velocity_{max} * GEAR.OUT/GEAR.IN < GEAR.VELMAX

Related Topics

10.7 Electronic Gearing

GEAR.MODE

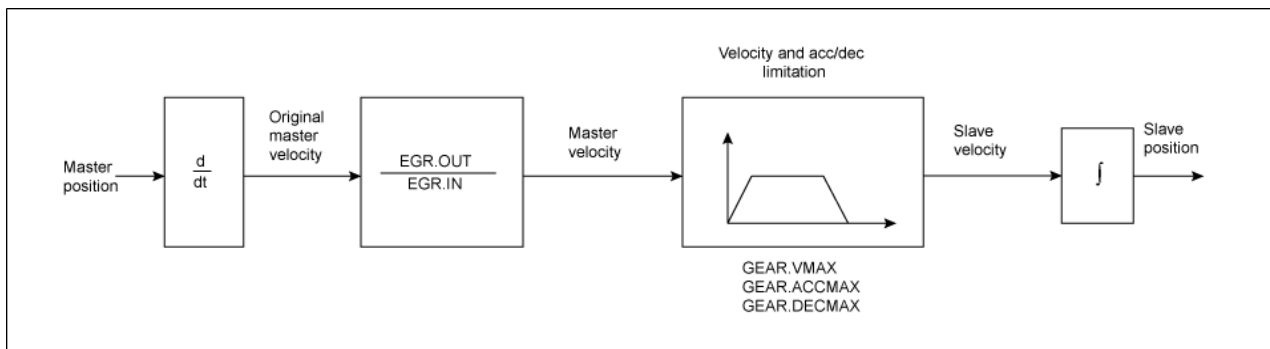
| General Information | |
|---------------------|--|
| Type | R/W Parameter |
| Description | Selects electronic gearing mode; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 376 | M_01-03-00-000 |

Description

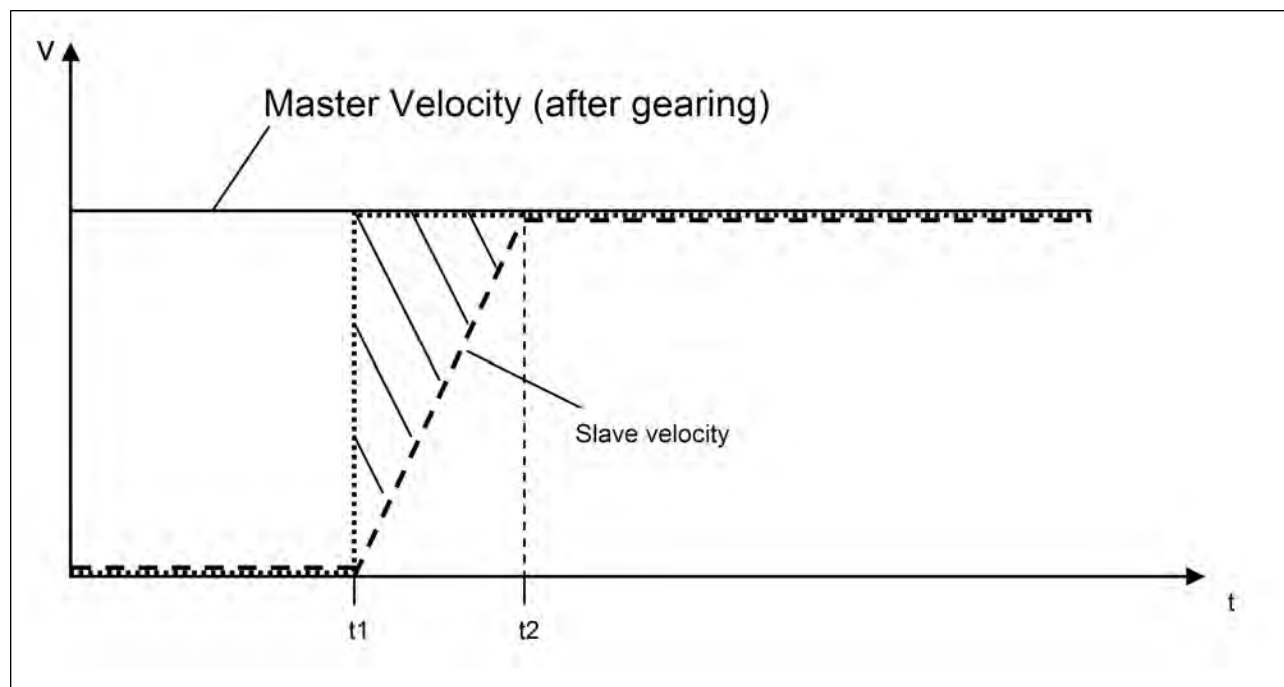
This parameter selects the electronic gearing mode at the beginning of the electronic gearing procedure. The gearing mode determines drive synchronization. In the context of electronic gearing, synchronization means that the slave follows the master pulses without losing counts due to acceleration or velocity limitations.

- Mode 0: The slave is not immediately synchronized after a GEAR.MOVE command. The slave accelerates until the master velocity (velocity after the gearing) has been reached. The drive is synchronized as soon as the velocity of the master has been reached.
- Mode 1: The slave is immediately synchronized after a GEAR.MOVE command.



Block diagram of the electronic gearing feature

The slave position is forwarded to the position loop. Ensure that DRV.OPMODE has been set to 2 and DRV.CMDSOURCE has been set to 2. The slave is able to reach the master velocity according to the GEAR.VMAX setting. GEAR.VMAX does not limit the slave velocity.



Acceleration process for GEAR.MODE 0

| Time | Description |
|-----------------|--|
| $t < t_1$ | The master sends already signals to the slave drive, but the GEAR.MOVE command has not been triggered yet. |
| $t = t_1$ | A GEAR.MOVE command has been triggered. |
| $t_1 < t < t_2$ | The slave accelerates according to the GEAR.ACCMAX setting. The position, which is marked with solid lines, will be ignored. |
| $t = t_2$ | The slave has reached the master velocity and is from now on considered as synchronized. Synchronization means that the slave will not lose any more position counts coming from the master. |
| $t > t_2$ | The slave follows the master input signals. |

Related Topics

10.7 Electronic Gearing

GEAR.MOVE

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Starts the electronic gearing; active in opmode 2 (position) only. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 378 | M_01-03-00-000 |

Description

The command GEAR.MOVE starts the electronic gearing procedure according to the selected electronic gearing mode. The electronic gearing process can be stopped using the DRV.STOP command.

Related Topics

10.7 Electronic Gearing

GEAR.OUT

General Information

| | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the numerator of the electronic gearing ratio; active in opmode 2 (position) only. |
| Units | N/A |
| Range | -32,768 to +32,767 |
| Default Value | 1 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 380 | M_01-03-00-000 |

Description

This parameter is the numerator of the gear ratio for the electronic gearing mode. The gear ratio is used in order to increase/decrease the slave velocity. The slave velocity can be calculated by the following formula:

$$\text{Slave velocity} = \text{Master velocity} * \text{GEAR.OUT/GEAR.IN}$$

Make sure that the external master source has been set properly. Also, be certain to select a gear ratio such that the maximum electronic gearing velocity (GEAR.VELMAX) will not be exceeded.

$$\text{Master velocity}_{\text{max}} * \text{GEAR.OUT/GEAR.IN} < \text{GEAR.VELMAX}$$

Related Topics

10.7 Electronic Gearing

GEAR.VMAX

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Reads the maximum allowed velocity value; active in opmode 2 (position) only. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, μ m/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 250.000*MOTOR.PITCH mm/s 0.000 to 250,000.000*MOTOR.PITCH μ m/sec 0.000 to 1,250.000 custom units/s |
| Default Value | Rotary: 3,000 rpm 50 rps 18,000.002 deg/s 250.000 custom units/s 314.159 rad/s Linear: 0.050 counts/s 50 mm/s 50,000.004MOTOR.PITCH μ m/s 250.000 custom units/s |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 382 | M_01-03-00-000 |

Description

This parameter limits the maximum velocity of the slave drive.

Related Topics

10.7 Electronic Gearing

GUI Parameters

GUI parameters are used within WorkBench for data reporting and data storage.

GUI.DISPLAY

| General Information | |
|----------------------|---------------------------|
| Type | R/O Parameter |
| Description | Reads drive display data. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | Display |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

This parameter reports to the GUI what the drive currently is displaying. For all GUI commands, the data is compressed and formatted for the GUI, not for the user.

GUI.PARAM01

| General Information | |
|----------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

GUI.PARAM02

| General Information | |
|----------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

GUI.PARAM03

| General Information | |
|----------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

GUI.PARAM04

| General Information | |
|----------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

GUI.PARAM05

| General Information | |
|----------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

GUI.PARAM06

| General Information | |
|----------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

GUI.PARAM07

| General Information | |
|----------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

GUI.PARAM08

| General Information | |
|----------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

GUI.PARAM09

| General Information | |
|----------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

GUI.PARAM10

| General Information | |
|---------------------|--------------------------------|
| Type | NV Parameter |
| Description | Used by the GUI to store data. |
| Units | N/A |
| Range | 2,147,483,648 to 2,147,483,647 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The GUI uses this parameter to store data. Only the GUI can modify this data (not the user).

HOME Parameters

HOME.ACC

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets homing acceleration; active in opmode 2 (position) only. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s ² , (custom units)/s ² , rad/s ² Linear: counts/s ² , mm/s ² , μm/s ² , (custom units)/s ² |
| Range | Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s ² 0.155 to 4,166,666.752 (custom units)/s ² 0.012 to 5,235,987.968 rad/s ² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s ² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s ² 30.994*MOTOR.PITCH to 833,333,333.333*MOTOR.PITCH μm/s ² 0.155 to 4,166,666.667 (custom units)/s ² |
| Default Value | Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s ² 833.333 (custom units)/s ² 1,047.2 rad/s ² Linear: 715,840,000.000 counts/s ² 166.714*MOTOR.PITCHMOTOR.PITCH mm/s ² 166,714.191*MOTOR.PITCHMOTOR.PITCH μm/s ² 833.571 (custom units)/s ² |
| Data Type | Float |
| See Also | UNIT.ACCROTARY, UNIT.ACCLINEAR |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | 3502h/0 | |
| | Profile specific SDOs/0 | M_01-00-00-000 |
| Modbus | 384 (64-bit) | M_01-03-00-000 |

Description

This parameter determines the acceleration of the motor during the homing procedure.

Related Topics

12.1 Homing

HOME.AUTOMOVE

| General Information | |
|----------------------|-----------------------------|
| Type | R/W Parameter |
| Description | Sets homing auto move flag. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | HOME.MODE |
| Start Version | M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 388 | M_01-03-00-000 |

Description

HOME.AUTOMOVE allows the drive to start a homing procedure after enabling the drive.

HOME.AUTOMOVE=0: The drive is not allowed to start automatically a homing procedure after the enable command.

HOME.AUTOMOVE = 1: The drive automatically starts a homing procedure after the enable command.

When HOME.AUTOMOVE is set to 1, the drive continuously checks the following conditions:

1. Is the drive enabled (DRV.ACTIVE = 1)?
2. Is the drive in DRV.OPMODE = 2?
3. Has the command source been adjusted to 0 (DRV.CMDSOURCE = 0)?
4. Is no other motion currently active (see DRV.MOTIONSTAT)?

As soon as all of the conditions above are true, the drive activates the homing procedure, which has been selected via the HOME.MODE setting. The automatic homing procedure is finished as soon as the homing procedure has been successfully triggered by the drive. From now on the drive will not attempt to trigger any additional homing procedure.

HOME.AUTOMOVE is not functional in release M_01-03-00-000 for homing procedures which require an external index signal (HOME.MODE 3, 6, 10, and 11).

Related Topics:

12.1 Homing

HOME.DEC

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets homing deceleration; active in opmode 2 (position) only. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s ² , (custom units)/s ² , rad/s ² Linear: counts/s ² , mm/s ² , μm/s ² , (custom units)/s ² |
| Range | Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s ² 0.155 to 4,166,666.752 (custom units)/s ² 0.012 to 5,235,987.968 rad/s ² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s ² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s ² 30.994*MOTOR.PITCH to 833,333.333*MOTOR.PITCH μm/s ² 0.155 to 4,166,666.667 (custom units)/s ² |
| Default Value | Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s ² 833.333 (custom units)/s ² 1,047.2 rad/s ² Linear: 715,840,000.000 counts/s ² 166.714*MOTOR.PITCHMOTOR.PITCHmm/s ² 166,714.191MOTOR.PITCH μm/s ² 833.571 (custom units)/s ² |
| Data Type | Float |
| See Also | UNIT.ACCROTARY, UNIT.ACCLINEAR |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | 3524h/0 | M_01-00-00-000 |
| | Profile specific SDOs/0 | |
| Modbus | 390 (64-bit) | M_01-03-00-000 |

Description

This parameter sets the deceleration of the motor during the homing procedure.

Related Topics:

12.1 Homing

HOME.DIR

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets homing direction; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 1 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 6098h6098h | M_01-00-00-000 |
| Modbus | 394 | M_01-03-00-000 |

Description

This parameter determines the direction in which the motor should start to move during a homing procedure.

0 = Movement in negative direction.

1 = Movement in positive direction.

Related Topics

12.1 Homing

HOME.DIST

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets homing distance; active in opmode 2 (position) only. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEARUNIT.ACCLINEAR Rotary: counts, rad, deg, custom units, 16-bit counts Linear: counts, mm, μ m, custom units, 16-bit counts |
| Range | N/A |
| Default Value | 0 |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 396 | M_01-03-00-000 |

Description

This parameter takes effect only after the homing procedure is complete (see the HOME.MODE description). HOME.DIST specifies an additional movement after the homing procedure is complete. The drive uses the homing acceleration, deceleration, and velocity parameters for this movement. This parameter can be used to let the motor move away from the home position by the value of HOME.DIST.

A value not equal to 0 triggers an additional movement of the selected homing distance after the general homing procedure. A value of 0 for HOME.DIST causes no additional movement.

Related Topics

12.1 Homing

HOME.FEEDRATE

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets homing velocity factor; active in opmode 2 (position) only. |
| Units | % |
| Range | 0 to 100% |
| Default Value | 50% |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | Profile specific SDOs/2 | M_01-00-00-000 |
| Modbus | 400 | M_01-03-00-000 |

Description

This parameter is used in order to reduce the velocity during the index search (index = zero-pulse of a feedback device). This parameter determines the percentage of the homing velocity (HOME.V) that should be used during the index-search.

Related Topics

12.1 Homing

HOME.IPEAK

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the current limit during homing procedure to a mechanical stop; active in opmode 2 (position) only. |
| Units | A |
| Range | ± Drive peak current A |
| Default Value | $[(1/120) * DRV.IPEAK]$ A |
| Data Type | Float |
| See Also | HOME.MODE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35E2h/0 | M_01-00-00-000 |
| Modbus | 402 (64-bit) | M_01-03-00-000 |

Description

This parameter sets the intermediate current limit during a homing procedure to a mechanical stop (HOME.MODE 8 and 9). The current-controller limit (IL.LIMITP and IL.LIMITN) is set to ±HOME.IPEAK while the homing procedures are active.

HOME.IPEAK is active as soon as the homing procedure starts and remains active until the home position is found. Previous current limit settings are re-activated before the motor covers the homing distance (HOME.DIST ≠ 0).

Related Topics

12.1 HomingHoming mode 8: Move Until Position Error Exceeded

HOME.MODE

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Selects the homing mode; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 0 to 10 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 6098h6098h | M_01-00-00-000 |
| Modbus | 406 | M_01-03-00-000 |

Description

HOME.MODE specifies the homing procedure of the drive. The homing modes available in the drive are summarized in the following table; see 12.1 Homing for a detailed discussion and examples for each mode:

| Mode | Description |
|------|---|
| 0 | Home using current position |
| 1 | Find limit input |
| 2 | Find input limit then find zero angle |
| 3 | Find input limit then find index |
| 4 | Find home input, including hardware limit switches |
| 5 | Find home input then find zero angle, including hardware limit switches |
| 6 | Find home input then find index, including hardware limit switches. |
| 7 | Find zero angle |
| 8 | Move until position error exceeded |
| 9 | Move until position error exceeded, then find zero angle |
| 10 | Move until position error exceeded, then find index |
| 11 | Find index signal, without any pre-condition |
| 12 | Homing to a home-switch, including mechanical stop detection |

Related Topics

12.1 Homing

HOME.MOVE

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Starts a homing procedure; active in opmode 2 (position) only. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

The HOME.MOVE command starts a homing procedure. The DRV.OPMODE must be set to 2 (closed position loop) and DRV.CMDSOURCE must be set to 0 (TCP/IP command).

Related Topics

12.1 Homing

HOME.P

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets home position; active in opmode 2 (position) only. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, custom units, 16-bit counts Linear: counts, mm, μ m, custom units, 16-bit counts |
| Range | N/A |
| Default Value | 0 |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | Profile specific SDOs/0 | M_01-00-00-000 |
| Modbus | 410 (64-bit) | M_01-03-00-000 |

Description

This parameter sets the home position. The command and actual position of the drive will be set to this value as soon as a homing event occurs. The homing events differ in each homing mode.

Related Topics

12.1 Homing

HOME.PERRTHRESH

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the position lag threshold; active in opmode 2 (position) only. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, custom units, 16-bit counts Linear: counts, mm, μ m, custom units, 16-bit counts |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3482h/0 | M_01-00-00-000 |
| Modbus | 414 (64-bit) | M_01-03-00-000 |

Description

This parameter is used for the homing modes against a mechanical stop (HOME.MODE = 8 and 9). The absolute value of the following error (PL.ERR) is compared with HOME.PERRTHRESH in order to detect a mechanical stop.

Related Topics

Homing mode 8: Move Until Position Error Exceeded

HOME.REQUIRE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Defines if the axis must be homed before a motion task can be executed. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 1 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-03-00-005 |

Description

This parameter defines whether or not the axis must be homed before a motion task can be executed.

- HOME.REQUIRE = 1: Homing must be complete ("Home Done" is true) before a motion task can execute.
- HOME.REQUIRE = 0: The axis does not need to be homed before a motion task can execute. When HOME.REQUIRE is set to 0, "Home Done" can be either true or false before a motion task can execute.

Related Topics

12.1 Homing

12.2 Motion Tasks

HOME.SET

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Immediately sets the home position; active in opmode 2 (position) only. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35F0h/0 | M_01-00-00-000 |
| Modbus | 418 | M_01-03-00-000 |

Description

The HOME.SET command immediately homes the drive. The drive can be homed in an enabled or disabled state. Motion in the current mode of operation (DRV.OPMODE=0) or velocity mode of operation (DRV.OPMODE=1) is not affected by the HOME.SET command. Motion in the position mode of operation (DRV.OPMODE=2) is immediately aborted when the HOME.SET command is issued.

Related Topics

12.1 Homing

HOME.V

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets homing velocity; active in opmode 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, μ m/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 250.000*MOTOR.PITCH mm/s 0.000 to 250,000.000*MOTOR.PITCH μ m/sec 0.000 to 1,250.000 custom units/s |
| Default Value | Rotary: 60 rpm 1 rps 359.999 deg/s 5 (custom units)/s 6.283 rad/s Linear: 0.001 counts/s 1*MOTOR.PITCH mm/s 999.998*MOTOR.PITCH μ m/sec 5.000 custom units/s |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 6099h/1 | M_01-00-00-000 |
| Modbus | 420 | M_01-03-00-000 |

Description

This parameter sets the velocity of the motor during the homing procedure.

Related Topics

12.1 Homing

HWLS Parameters

HWLS.NEGSTATE

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the status of the negative hardware limit switch. |
| Units | 0 to 1 |
| Range | N/A |
| Default Value | Boolean |
| Data Type | HWLS.POSSTATE |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 422 | M_01-03-00-000 |

Description

HWLS.NEGSTATE reads the status of the negative HW limit switch as follows:

0 = Low

1 = High

Related Topics

10.1 Digital Inputs and Outputs

HWLS.POSSTATE

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the status of the positive hardware limit switch. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | N/A |
| Data Type | Boolean |
| See Also | HWLS.NEGSTATE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 424 | M_01-03-00-000 |

Description

HWLS.POSSTATE reads the status of the positive hardware limit switch as follows:

0 = Low

1 = High

Related Topics

10.1 Digital Inputs and Outputs

This page intentionally left blank.

IL Parameters

IL.BUSFF

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Displays the current feedforward value injected by the fieldbus. |
| Units | Arms |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | IL.KBUSFF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 426 | M_01-03-00-000 |

Description

This parameter displays the current feedforward value injected by the fieldbus.

Related Topics

11.3 Current Loop

IL.CMD

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the value of the q-component current controller inside the FPGA. |
| Units | Arms |
| Range | ± Drive peak current (DRV.IPEAK) |
| Default Value | N/A |
| Data Type | Float |
| See Also | DRV.IPEAK |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 428 | M_01-03-00-000 |

Description

IL.CMD displays the q-component current command value of the current loop after any limitation (such as a parameter setting or I^2t calculation).

IL.CMD is limited also by motor peak current, IL.LIMITN and IL.LIMITP.

Related Topics

10.4 Analog Input

11.3 Current Loop

IL.CMDU

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the user current command. |
| Units | Arms |
| Range | Minimum range value = maximum of IL.LIMITN and -MOTOR.IPEAK Maximum range value = minimum of IL.LIMITP and MOTOR.IPEAK |
| Default Value | 0 Arms |
| Data Type | Float |
| See Also | DRV.IPEAK, DRV.OPMODE, DRV.CMDSOURCE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 430 | M_01-03-00-000 |

Description

This parameter sets the user current command value.

The current command value, which is provided to the current loop (IL.CMD), can be limited further using a parameter setting or I^2t calculation. IL.CMDU is limited also by motor peak current, IL.LIMITN and IL.LIMITP.

Related Topics

11.3 Current Loop

IL.DIFOLD

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the drive foldback current limit. |
| Units | Arms |
| Range | 0 to 2,147,483.647 Arms |
| Default Value | N/A |
| Data Type | Float |
| See Also | Foldback |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3559h/0 | M_01-00-00-000 |

Description

IL.DIFOLD is the output of the drive foldback algorithm. It is an artificial current, which can be higher or lower than the drive peak current (DRV.IPEAK). When IL.DIFOLD is lower than the existing current limit (such as IL.LIMITP), it becomes the active current limit.

IL.DIFOLD decreases when the actual current is higher than drive continuous current and increases (up to a certain level) when the actual current is lower than drive continuous current.

Related Topics

11.3 Current Loop

IL.FB

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the actual value of the d-component current. |
| Units | Arms |
| Range | ± Drive peak current (DRV.IPEAK) |
| Default Value | N/A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3558h/0 | M_01-00-00-000 |
| Modbus | 432 | M_01-03-00-000 |

Description

This parameter reads the measured, de-rotated actual current value of the motor.

Related Topics

11.3 Current Loop

IL.FF

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Displays the current loop overall feedforward value |
| Units | Arms |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | IL.KBUSFF, IL.KVFF, IL.OFFSET, IL.FRICTION, IL.KACCF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 434 | M_01-03-00-000 |

Description

This parameter displays the current loop overall feedforward value.

Related Topics

11.3 Current Loop

IL.FOLDFTHRESH

| General Information | |
|----------------------|---------------------------------|
| Type | R/O Parameter |
| Description | Reads the foldback fault level. |
| Units | Arms |
| Range | 0 to 500 Arms |
| Default Value | Drive peak current (DRV.IPEAK) |
| Data Type | Float |
| See Also | Foldback |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3420h/0 | M_01-00-00-000 |
| Modbus | 436 | M_01-03-00-000 |

Description

IL.FOLDFTHRESH is the fault level of the current foldback algorithm. If IL.IFOLD drops below the value for IL.FOLDFTHRESH, then a fault is generated and the drive is disabled.

To avoid reaching the current foldback fault level, set IL.FOLDFTHRESHU well below the continuous current value for both the drive and the motor or set the IL.FOLDFTHRESHU value to zero.

Related Topics

11.3 Current Loop

IL.FOLDFTHRESHU

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the user value for the foldback fault level. |
| Units | Arms |
| Range | 0 to 500 Arms |
| Default Value | Drive peak current (DRV.IPEAK) |
| Data Type | Float |
| See Also | IL.FOLDFTHRESH, Foldback |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3421h/0 | M_01-00-00-000 |
| Modbus | 438 | M_01-03-00-000 |

Description

IL.FOLDFTHRESHU is the fault level of the current foldback algorithm. The value of IL.FOLDFTHRESH is the minimum of DRV.IPEAK, MOTOR.IPEAK, and IL.FOLDFTHRESHU.

Related Topics

11.3 Current Loop

IL.FOLDWTHRESH

| General Information | |
|----------------------|----------------------------------|
| Type | NV Parameter |
| Description | Sets the foldback warning level. |
| Units | Arms |
| Range | 0 to 500 Arms |
| Default Value | 0 A |
| Data Type | Float |
| See Also | Foldback |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 355Ah/0 | M_01-00-00-000 |
| Modbus | 440 | M_01-03-00-000 |

Description

IL.FOLDWTHRESH is the warning level of the current foldback algorithm. When IL.IFOLD drops **below** IL.FOLDWTHRESH a warning is generated.

To ensure that the current foldback warning level is never reached, IL.FOLDWTHRESH should be set well below the continuous current value for both the drive and the motor. You can also set the IL.FOLDFTHRESH value to zero.

Related Topics

11.3 Current Loop

IL.FRICTION

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets friction compensation value. |
| Units | A |
| Range | 0 to the minimum of user positive current limit (IL.LIMITP) and motor peak current (MOTOR.IPEAK).IL.LIMITP |
| Default Value | 0 |
| Data Type | Float |
| See Also | IL.FF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3422h/0 | M_01-00-00-000 |
| Modbus | 442 | M_01-03-00-000 |

Description

Position command derivative sign is multiplied by this value to be injected to the current command.

Related Topics

11.3 Current Loop

IL.IFOLD

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the overall foldback current limit. |
| Units | A |
| Range | 0 to 2,147,483.647 A |
| Default Value | N/A |
| Data Type | Float |
| See Also | Foldback |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3425h/0 | M_01-00-00-000 |
| Modbus | 444 | M_01-03-00-000 |

Description

Two current foldback algorithms run in parallel in the drive: the drive foldback algorithm and the motor foldback algorithm. Each algorithm uses different sets of parameters.

Each algorithm has its own foldback current limit, IL.DIFOLD and IL.MIFOLD. The overall foldback current limit is the minimum of the two at any given moment.

$$IL.IFOLD = \min (IL.DIFOLD, IL.MIFOLD).$$

IL.DIFOLD is an artificial current, which can be higher or lower than the drive or motor peak current. When IL.IFOLD becomes lower than the existing current limit (such as IL.LIMITP), it becomes the active current limit.

Related Topics

11.3 Current Loop

IL.IUFB

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the sigma-delta measured current in the u-winding of the motor. |
| Units | A |
| Range | ± Drive peak current (DRV.IPEAK) |
| Default Value | N/A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 446 | M_01-03-00-000 |

Description

This parameter displays the measured current in the u-winding of the motor.

Related Topics

11.3 Current Loop

IL.IVFB

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the sigma-delta measured current in the u-winding of the motor. |
| Units | A |
| Range | ± Drive peak current (DRV.IPEAK) |
| Default Value | 0 A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 448 | M_01-03-00-000 |

Description

IL.IVFB is an offset value that is added to the measured current in the u-winding of the motor. This value is used for compensating for an error in the current measurement. The drive measures 256 times the current in the u-winding when powering-up the drive. Afterwards, the drive calculates the average value of the measured current and uses this value for the offset value.

Related Topics

11.3 Current Loop

IL.KACCFF

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets current loop acceleration feedforward gain value |
| Units | mArms/(rad/s ²) |
| Range | 0.0 to 2.0 mArms/(rad/s ²) |
| Default Value | 0 mArms/(rad/s ²) |
| Data Type | Float |
| See Also | IL.FF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3426h/0 | M_01-00-00-000 |
| Modbus | 450 | M_01-03-00-000 |

Description

This value sets the gain for the acceleration feedforward (a scaled second derivative of the position command is added to the current command value).

This parameter is valid only in the position mode (DRV.OPMODE = 2).

Related Topics

11.3 Current Loop

IL.KBUSFF

| General Information | |
|----------------------|---|
| Type | RW |
| Description | Current loops fieldbus injected feed-forward gain |
| Units | NA |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | Float |
| See Also | IL.FF, IL.BUSFF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 452 | M_01-03-00-000 |

Description

This parameter scales the feedforward term added by the fieldbus to the current command. The nominal feed-forward value can be multiplied by this gain value.

This parameter is only used in the position mode (DRV.OPMODE = 2).

Related Topics

11.3 Current Loop

IL.KP

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the proportional gain of the q-component of the PI regulator. |
| Units | V/A |
| Range | 0 to 2,000 V/A |
| Default Value | Read from the motor or, if no memory, 50.009 V/A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3598h/0 | M_01-00-00-000 |
| Modbus | 454 | M_01-03-00-000 |

Description

IL.KP is used to modify the proportional gain of the PI-loop that controls the q-component of the current.

Related Topics

11.3 Current Loop

IL.KPDRATIO

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the proportional gain of the d-component current PI-regulator as a percentage of IL.KP |
| Units | N/A |
| Range | 0 to 100 |
| Default Value | 1 |
| Data Type | Float |
| See Also | IL.KP |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3596h/0 | M_01-00-00-000 |
| Modbus | 456 | M_01-03-00-000 |

Description

This parameter modifies the proportional gain of the PI-loop, which controls the d-component of the current.

Related Topics

11.3 Current Loop

IL.KPLOOKUPINDEX

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the index into the Current Loop Gain Scheduling Table. |
| Units | N/A |
| Range | 0 to 255 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | IL.KPLOOKUPVALUEIL.KPLOOKUPVALUESIL.KP |
| Start Version | M_01-04-00-000 |

Description

This parameter sets the index into the Current Loop Gain Scheduling Table. The table is 256 records long, spanning 0 A to $1.62 * \text{DRV.IPEAK}$.

To determine the level of current that corresponds to a table index, use the following equation:

$$\text{IL.CMD} = (\text{Table Index} / 157) * \text{DRV.IPEAK}$$

Related Topics

Current Loop Gain Scheduling

Current Loop

IL.KPLOOKUPVALUE

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the value of the current loop gain scheduling index. |
| Units | % |
| Range | 0 to 100.000% |
| Default Value | 0 |
| Data Type | Float |
| See Also | IL.KPLOOKUPINDEXIL.KPLOOKUPVALUESIL.KP |
| Start Version | M_01-04-00-000 |

Description

This parameter sets the value at the current index into the Current Loop Gain Scheduling Table. The table is 256 records long, spanning 0 A to $1.62 * DRV.IPEAK$. The value can range from 0% to 100% and determines what percentage of IL.KP will be applied to the current loop.

To determine what level of current corresponds to a table index, use the following equation:

$$IL.CMD = (Table\ Index / 157) * DRV.IPEAK$$

Example

Assume:

$DRV.IPEAK = 9\ A$

$IL.KPLOOKUPINDEX = 100$

$IL.KPLOOKUPVALUE = 50$

$IL.KP = 240$

When $IL.CMD = 100 / 157 * 9 = 5.73\ A$, IL.KP will not be 240, but will be $50\% * 240 = 120$.

Related Topics

Current Loop Gain Scheduling

Current Loop

IL.KPLOOKUPVALUES

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Gets the Current Loop Gain Scheduling Table. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | Table |
| See Also | IL.KPLOOKUPINDEX, IL.KPLOOKUPVALUEIL.KP |
| Start Version | M_01-04-00-000 |

Description

Retrieves the Current Loop Gain Scheduling Table in a comma delimited table.

This table is 256 records long, and the table will return values in the following format:

```
-->IL.KPLOOKUPVALUES
```

```
Index Value
```

```
0, 100.000
```

```
1, 100.000
```

```
2, 100.000
```

```
3, 100.000
```

```
4, 100.000
```

```
5, 100.000
```

```
6, 100.000
```

```
7, 100.000
```

```
8, 100.000
```

```
9, 100.000
```

```
10, 100.000
```

Related Topics

Current Loop Gain Scheduling

Current Loop

IL.KVFF

| General Information | |
|----------------------|--|
| Type | R/W |
| Description | Current loop velocity feed-forward gain. |
| Units | N/A |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | Float |
| See Also | IL.FF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 458 | M_01-03-00-000 |

Description

This parameter sets the gain for the velocity loop feedforward. The nominal feedforward value can be multiplied by this gain value.

This parameter is only used in position mode (DRV.OPMODE = 2).

Related Topics

11.3 Current Loop

IL.LIMITN

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the negative user (application-specific) current limit. |
| Units | A |
| Range | Negative drive peak current (DRV.IPEAK) to 0 A |
| Default Value | Negative drive peak current (DRV.IPEAK) |
| Data Type | Float |
| See Also | IL.LIMITP |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 356Fh/0 | M_01-00-00-000 |
| Modbus | 460 | M_01-03-00-000 |

Description

This parameter sets the negative user limit clamp value of the torqueproducing q-component current command (IL.CMD). The current command is additionally limited by the motor peak current setting (MOTOR.IPEAK) and by the present value of the foldback I²t peak motor current protection.

Related Topics

11.3 Current Loop

IL.LIMITP

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the positive user (application-specific) current limit. |
| Units | A |
| Range | 0 A to drive peak current (DRV.IPEAK) |
| Default Value | Drive peak current (DRV.IPEAK) |
| Data Type | Float |
| See Also | IL.LIMITN |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 356Eh/0 | M_01-00-00-000 |
| Modbus | 462 | M_01-03-00-000 |

Description

This parameter sets the positive user limit clamp value of the torque-producing q-component current command (IL.CMD). The current command is additionally limited by the motor peak current setting (MOTOR.IPEAK) and by the present value of the foldback I²t peak motor current protection.

Related Topics

11.3 Current Loop

IL.MFOLDD

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Sets the motor foldback maximum time at motor peak current. |
| Units | s |
| Range | 0.1 to 2400 s |
| Default Value | 10 s |
| Data Type | Float |
| See Also | Foldback |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 464 | M_01-03-00-000 |

Description

IL.MFOLDD sets the maximum time allowed for the motor to remain at peak current before starting to fold towards the motor continuous current. When at motor peak current, IL.MFOLDD is the amount of time before the foldback algorithm starts to reduce the current.

Related Topics

11.3 Current Loop

IL.MFOLDR

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Sets the motor foldback recovery time. |
| Units | s |
| Range | 0.1 to 65,535 s |
| Default Value | Calculated from other foldback parameters. |
| Data Type | Float |
| See Also | Foldback |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 466 | M_01-03-00-000 |

Description

IL.MFOLDR sets the recovery time for the motor foldback algorithm. If 0 current is applied for at least the recovery time duration, it is possible to apply motor peak current for the duration of IL.MFOLDD time.

The IL.MFOLDR value is automatically calculated from other foldback parameters.

Related Topics

11.3 Current Loop

IL.MFOLDT

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Sets the motor foldback time constant of the exponential current drop (foldback). |
| Units | s |
| Range | 0.1 to 2,400 s |
| Default Value | 10 s |
| Data Type | Float |
| See Also | Foldback |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 468 | M_01-03-00-000 |

Description

IL.MFOLDT sets the time constant of the exponential drop (foldback) of the current towards motor continuous current.

Related Topics

11.3 Current Loop

IL.MIFOLD

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Sets the motor foldback current limit. |
| Units | A |
| Range | 0 to 2147483.647 A |
| Default Value | N/A |
| Data Type | Float |
| See Also | Foldback |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35A4h/0 | M_01-00-00-000 |
| Modbus | 470 | M_01-03-00-000 |

Description

IL.MIFOLD sets the output of the motor foldback algorithm. It is an artificial current, which can be higher or lower than the motor peak current. When IL.MIFOLD becomes lower than the existing current limit (IL.LIMITP) it becomes the active current limit.

IL.MIFOLD decreases when the actual current is higher than motor continuous current and increases (up to a certain level) when the actual current is lower than the motor continuous current.

Related Topics

11.3 Current Loop

IL.OFFSET

| General Information | |
|----------------------|---|
| Type | RW Parameter |
| Description | A constant current command added to compensate for gravity. |
| Units | A |
| Range | [IL.LIMITN to IL.LIMITP |
| Default Value | 0 A |
| Data Type | Float |
| See Also | IL.FF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3423h/0 | M_01-00-00-000 |
| Modbus | 472 | M_01-03-00-000 |

Description

This value is added to the overall current loop feedforward value.

Related Topics

11.3 Current Loop

IL.VCMD

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Sets the output of the q-component PI regulator. |
| Units | Vrms |
| Range | 0 Vrms to bus voltage |
| Default Value | N/A |
| Data Type | Integer |
| See Also | IL.VDCMD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 474 | M_01-03-00-000 |

Description

Sets the output of the current loop that controls the q-component of the current.

Related Topics

11.3 Current Loop

IL.VUFB

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the measured voltage on the u-winding of the motor. |
| Units | V |
| Range | -1200*VBusScale to +1200*VBusScale |
| Default Value | N/A |
| Data Type | Integer |
| See Also | IL.VVFB |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 476 | M_01-03-00-000 |

Description

Reads the measured voltage on the u-winding of the motor.

Related Topics

11.3 Current Loop

IL.VVFB

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the measured voltage on the v-winding of the motor. |
| Units | V |
| Range | -1200*VBusScale to +1200*VBusScale |
| Default Value | N/A |
| Data Type | Integer |
| See Also | IL.VUFB |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 478 | M_01-03-00-000 |

Description

The range for this parameter depends on whether the drive model is an MV/240 Vac or an HV/480 Vac.

The VBusScale parameter sets the drive model:

MV/240 Vac: VBusScale = 1

HV/480 Vac: VBusScale = 2

VBusScale is used for multiple parameter ranges that are model dependent, such as IL.KP.

Related Topics

11.3 Current Loop

23 LOAD.INERTIA

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the load inertia. |
| Units | kgcm ² for rotary motors kg for linear motors |
| Range | 1 to 1,000,000 kgcm ² or kg |
| Default Value | 0 kgcm ² or kg |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-03-06-000 |

Description

LOAD.INERTIA sets the load inertia.

Related Topics

Motor

24 MODBUS Parameters

MODBUS.PIN AND MODBUS.POUT

See UNIT.PIN and UNIT.POUT.

MODBUS.PSCALE

| General Information | |
|----------------------|---|
| Type | Parameter |
| Description | Sets the resolution of value PL.FB in bits. |
| Units | bits |
| Range | 0 to 32 bits |
| Default Value | 20 bits |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-04-00-000 |

Description

This parameter sets the resolution of PL.FB value in bits.

Example

```
MODBUS.PSCALE = 10
```

PL.FB has a resolution of 10 bits per rev (1024 counts per rev).

MODBUS.UNITLABEL

| General Information | |
|----------------------|---|
| Type | Parameter |
| Description | Configures the drive for the induction motor's minimum voltage at standstill. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-04-00-000 |

Description

This parameter configures the drive for the induction motor's minimum voltage at standstill. It is given as a percentage (%) of the motor's rated voltage. MOTOR.VOLTMIN is used to calculate the constant Volts per Hertz characteristics of the drive/motor and should be set to a value that generates a current of about 40% of the rated current at standstill.

Related Topics

Motor

MOTOR Parameters

MOTOR.AUTOSSET

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Determines which drive parameters are calculated automatically. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3451h/0 | M_01-00-00-000 |
| Modbus | 480 | M_01-03-00-000 |

Description

This parameter determines whether or not certain drive parameters (for example, IL.KP or MOTOR.POLES) are calculated automatically. A value of 1 causes the parameters to be automatically calculated from the motor ID data (read from memory-supporting feedback devices, such as SFD, Endat, and BISS). Automatically calculated parameters are read-only. A value of 0 disables the automatic calculation and you must set the parameters manually. Manually set parameters are read-write.

Related Topics

8.1 Motor

MOTOR.BRAKE

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the presence or absence of a motor brake. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3587h/0 | M_01-00-00-000 |
| Modbus | 482 | M_01-03-00-000 |

Description

The MOTOR.BRAKE parameter notifies the firmware whether a brake exists or not. It does not apply or release the brake. If a brake is found to be present, the firmware considers hardware indications regarding the brake circuits (such as open circuit or short circuit). If a brake does not exist, then the firmware ignores the hardware indications since they are irrelevant.

| Value | Status |
|-------|---|
| 0 | Motor brake does not exist. |
| 1 | Motor brake exists and brake hardware circuitry checks are enabled. |

Enabling the MOTOR.BRAKE (value set to 1) when no motor brake exists creates a fault.

The motor brake is polled every 16 ms.

Related Topics

8.1 Motor

MOTOR.BRAKERLS

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Allows a user to release the motor brake. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3450h/0 | M_01-00-00-000 |
| Modbus | 484 | M_01-03-00-000 |

Description

This command allows a user to release the motor brake.

0 = Drive controls the brake.

1 = Brake is released.

Note: A digital input mode is also used for the same purpose. The two mechanisms are independent.

Related Topics

8.1 Motor

MOTOR.BRAKESTATE

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the actual status of the motor brake. |
| Units | N/A |
| Range | Brake released or not present. Brake applied. |
| Default Value | Brake applied or not present. |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

This parameter reads the actual status of the motor brake and can only show two states:

1 = Brake released or not present

2 = Brake applied

Related Topics

8.1 Motor

MOTOR.CTF0

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the thermal constant of the motor coil. |
| Units | mHz |
| Range | 0.265 to 16,000 mHz |
| Default Value | 10 mHz |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3454h/0 | M_01-00-00-000 |
| Modbus | 486 | M_01-03-00-000 |

Description

This parameter is used to configure the thermal constant of the motor coil, which is the break frequency of a single-pole low-pass filter model of the thermal dynamics of the motor coil.

This parameter, together with MOTOR.IPEAK and MOTOR.ICONT, determine the motor foldback parameters IL.MFOLDD, IL.MFOLDT, and IL.MFOLDR.

Calculating MOTOR.CTF0

Given a motor coil/winding thermal time constant T in seconds, then:

$$\text{MOTOR.CTF0} = 1/(2\pi T)$$

Related Topics

8.1 Motor

MOTOR.ICONT

| General Information | |
|----------------------|------------------------------------|
| Type | NV Parameter |
| Description | Sets the motor continuous current. |
| Units | A |
| Range | 0.1 to 500 A |
| Default Value | 1.0 A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 358Eh/0 | M_01-00-00-000 |
| Modbus | 488 | M_01-03-00-000 |

Description

This parameter is used to configure the motor continuous current.

Related Topics

8.1 Motor

MOTOR.IDDATAVALID

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reports the status of the motor memory. |
| Units | N/A |
| Range | N/A |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 490 | M_01-03-00-000 |

Description

MOTOR.IDDATAVALID reports the status of the motor memory status.

The valid values for this keyword are the following:

| Value | Description |
|-------|---|
| 0 | Error in identification |
| 1 | Success in identification |
| 2 | Identification in process |
| 3 | Identification not started yet |
| 4 | Success recognizing feedback, but failed to verify OEM data integrity |

Related Topics

8.1 Motor

MOTOR.INERTIA

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the motor inertia. |
| Units | kgcm ² for rotary motors kg for linear motors |
| Range | 1 to 200,000 kgcm ² or kg |
| Default Value | 100 kgcm ² or kg |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35ABh/0 | M_01-00-00-000 |
| Modbus | 492 | M_01-03-00-000 |

Description

This parameter sets the motor inertia.

Related Topics

8.1 Motor

MOTOR.IPEAK

| General Information | |
|----------------------|------------------------------|
| Type | NV Parameter |
| Description | Sets the motor peak current. |
| Units | mA |
| Range | 0.200 to 1,000 A |
| Default Value | 2.000 A |
| Data Type | Float |
| See Also | IL.LIMITP, IL.LIMITN |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 358Fh/0 | M_01-00-00-000 |
| Modbus | 494 | M_01-03-00-000 |

Description

This parameter configures the drive for the motor's peak, instantaneous-rated current. MOTOR.IPEAK is used to limit clamp the magnitude of the torque producing q-component current command (IL.CMD).

Related Topics

8.1 Motor

MOTOR.KE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the motor back EMF constant. |
| Units | V _{peak} /krpm for Rotary Motors V _{peak} /m/s for Linear Motors |
| Range | 0.0 to 100,000 |
| Default Value | 0 |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-03-06-000 |

Description

MOTOR.KE defines the back EMF constant for the motor. The back EMF constant defines how much voltage is generated at the motors coils. The relationship between MOTOR.KE and speed is described by the following equation:

$$\text{Coil Voltage} = \text{MOTOR.KE} * \text{VL.FB}$$

Where:

VL.FB is in units of krpm for rotary motors and in units of m/s for linear motors

Related Topics

Motor

MOTOR.KT

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the torque constant of the motor. |
| Units | Nm/A |
| Range | 0.001 Nm/A to 1,000,000.000 Nm/A for rotary motors. 0.001 Nm/A to 1,000,000.000 N/A for linear motors. |
| Default Value | 0.1 Nm/A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3593h/0 | M_01-00-00-000 |
| Modbus | 496 | M_01-03-00-000 |

Description

This parameter is the torque constant of the motor in Nm/A. The value can be online checked according to the following equation:

$$K_t = 60 \cdot \sqrt{3} \cdot U_i / (2 \cdot \pi \cdot n)$$

Where:

U_i = induced voltage of the motor

n = actual rotor velocity

Related Topics

8.1 Motor

MOTOR.LQLL

| General Information | |
|----------------------|---------------------------------|
| Type | NV Parameter |
| Description | Sets the line-to-line motor Lq. |
| Units | mH |
| Range | 1 to 2 ³² H |
| Default Value | 17.000 H |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3455h/0 | M_01-00-00-000 |
| Modbus | 498 | M_01-03-00-000 |

Description

This parameter is used to configure the motor line-to-line inductance.

Related Topics

8.1 Motor

MOTOR.NAME

| General Information | |
|---------------------|----------------------|
| Type | NV Parameter |
| Description | Sets the motor name. |
| Units | N/A |
| Range | 11 chars |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

This parameter is used to set the motor name.

Related Topics

8.1 Motor

MOTOR.PHASE

| General Information | |
|----------------------|-----------------------|
| Type | NV Parameter |
| Description | Sets the motor phase. |
| Units | Electrical degrees |
| Range | 0 to 360° |
| Default Value | 0° |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 359Ch/0 | M_01-00-00-000 |
| Modbus | 500 | M_01-03-00-000 |

Description

This parameter sets the motor phase.

Related Topics

8.1 Motor

MOTOR.PITCH

| General Information | |
|----------------------|-----------------------|
| Type | NV Parameter |
| Description | Sets the motor pitch. |
| Units | µm |
| Range | 1,000 to 1,000,000 µm |
| Default Value | 1.000 µm |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 502 | M_01-03-00-000 |

Description

This parameter defines the pole-to-pair pitch for the linear motor in micrometers.

Related Topics

8.1 Motor

MOTOR.POLES

| General Information | |
|----------------------|---------------------------------|
| Type | NV Parameter |
| Description | Sets the number of motor poles. |
| Units | N/A |
| Range | 0 to 128 |
| Default Value | 6 |
| Data Type | Integer |
| See Also | FB1.POLES |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 359Dh/0 | M_01-00-00-000 |
| Modbus | 504 | M_01-03-00-000 |

Description

MOTOR.POLES sets the number of motor poles. This command is used for commutation control and represents the number of individual magnetic poles of the motor (not pole pairs). The division value of motor poles (MOTOR.POLES) and feedback poles (FB1.POLES) must be an integer when setting drive to enable, otherwise a fault is issued.

Related Topics

8.1 Motor

MOTOR.R

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the stator winding resistance phase-phase in ohms. |
| Units | Ω |
| Range | 0.001 to 650 Ω |
| Default Value | 10 Ω |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3456h/0 | M_01-00-00-000 |
| Modbus | 506 | M_01-03-00-000 |

Description

MOTOR.R sets the stator winding resistance phase-to-phase in ohms.

Related Topics

8.1 Motor

MOTOR.RTYPE

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Defines the type of thermal resistor inside the motor. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 508 | M_01-03-00-000 |

Description

This parameter defines the type of thermal resistor used inside of the motor to measures motor temperature.

0 = PTC

1 = NTC

Related Topics

8.1 Motor

MOTOR.TBRAKEAPP

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | The delay time used for applying the motor brake. |
| Units | ms |
| Range | 0 to 1,000 ms |
| Default Value | 75 ms |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 366Eh/0 | M_01-00-00-000 |
| Modbus | 510 | M_01-03-00-000 |

Description

This parameter is used to configure the mechanical delay when applying the motor brake. MOTOR.TBRAKEAPP is a time delay that is applied when a brake exists and the drive is disabled at the end of a controlled stop. This delay lasts from the time that the brake is commanded to apply until the time that the drive is disabled. This feature allows you to disable the drive and apply the brake on a vertical application without the load falling. Without this time delay, if you immediately disable the drive, then the load falls during the time needed for the brake to mechanically apply.

Related Topics

8.1 Motor

MOTOR.TBRAKERLS

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | The delay time used for releasing the motor brake. |
| Units | ms |
| Range | 0 to 1,000 ms |
| Default Value | 75 ms |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 366Fh/0 | M_01-00-00-000 |
| Modbus | 512 | M_01-03-00-000 |

Description

This parameter is used to configure the mechanical delay when releasing the motor brake. MOTOR.TBRAKERLS is a time delay that is applied when a brake exists and the drive is enabled. When the drive is enabled, the brake is commanded to release and, during the MOTOR.TBRAKERLS period of time, the drive does not accept a motion command. This delay allows the brake to fully release before the drive begins a new motion.

Related Topics

8.1 Motor

MOTOR.TEMP

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the motor temperature represented as the resistance of the motor PTC. |
| Units | Ω |
| Range | 0 to $2^{32} \Omega$ |
| Default Value | N/A |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3612h/0 | M_01-00-00-000 |
| Modbus | 514 | M_01-03-00-000 |

Description

This parameter is used to get the motor temperature which is represented as the resistance of the motor PTC.

Related Topics

8.1 Motor

MOTOR.TEMPFAULT

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the motor temperature fault level. |
| Units | Ω |
| Range | 0 to 2,000,000,000 Ω |
| Default Value | 0 Ω = switched off |
| Data Type | Integer |
| See Also | MOTOR.TEMP |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3586h/0 | M_01-00-00-000 |
| Modbus | 516 | M_01-03-00-000 |

Description

This parameter is used to configure the motor temperature fault level as a resistance threshold of the motor PTC. A zero value prevents any warning from being issued.

Related Topics

8.1 Motor

MOTOR.TEMPWARN

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the motor temperature warning level. |
| Units | Ω |
| Range | 0 to 2,000,000,000 Ω |
| Default Value | 0 Ω = switched off |
| Data Type | Integer |
| See Also | MOTOR.TEMP |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3453h/0 | M_01-00-00-000 |
| Modbus | 518 | M_01-03-00-000 |

Description

This parameter is used to configure the motor temperature warning level as a resistance threshold of the motor PTC.

A zero value prevents any warning from being created.

Related Topics

8.1 Motor

MOTOR.TYPE

| General Information | |
|----------------------|----------------------|
| Type | NV Parameter |
| Description | Sets the motor type. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------|----------------|----------------------|
| Modbus | 520 | M_01-03-00-000 |

Description

MOTOR.TYPE sets the drive control algorithms to different motor types as follows:

0 = Rotary motor

1 = Linear motor

Related Topics

8.1 Motor

MOTOR.VMAX

| General Information | |
|----------------------|-------------------------------|
| Type | NV Parameter |
| Description | Sets the maximum motor speed. |
| Units | rpm |
| Range | 100 to 40,000 rpm |
| Default Value | 3,000 rpm |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35A3h/0 | M_01-00-00-000 |
| Modbus | 522 | M_01-03-00-000 |

Description

This parameter is used to configure the maximum speed of the motor.

Related Topics

8.1 Motor

MOTOR.VOLTMAX

| General Information | |
|----------------------|---------------------------------|
| Type | NV Parameter |
| Description | Sets the motor maximum voltage. |
| Units | Vrms |
| Range | 110 to 900 Vrms |
| Default Value | 230 Vrms |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3452h/0 | M_01-00-00-000 |
| Modbus | 524 | M_01-03-00-000 |

Description

This parameter sets the maximum permissible motor voltage. For instance, if a motor that is rated for a 400 V supply is connected to the drive, then the MOTOR.VOLTMAX setting is 400. This value also sets regen resistor and over voltage thresholds in the drive to acceptable values for the motor so that the motor windings are not damaged.

Related Topics

8.1 Motor

MOTOR.VOLTMIN

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the minimum voltage for V/f control. |
| Units | % |
| Range | 0 to 100% |
| Default Value | 2% |
| Data Type | U16 |
| See Also | MOTOR.VRATED, MOTOR.VOLTRATED |
| Start Version | |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3457h/3 | M_01-00-00-000 |
| Modbus | | M_01-03-00-000 |

Description

This parameter configures the drive for the induction motor's minimum voltage at standstill. It is given as a percentage (%) of the motor's rated voltage. MOTOR.VOLTMIN is used to calculate the constant volts per Hertz characteristics of the drive and motor and should be set to a value that generates a current of about 40% of the rated current at standstill.

Related Topics

8.1 Motor

MOTOR.VOLTRATED

| General Information | |
|----------------------|-------------------------------|
| Type | NV Parameter |
| Description | Sets the motor rated voltage. |
| Units | V |
| Range | 50 to 1,000 V |
| Default Value | 230 V |
| Data Type | U16 |
| See Also | MOTOR.VRATED, MOTOR.VOLTMIN |
| Start Version | M_01-03-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|------------------------|----------------------|
| EtherCAT COE and CANopen | 3457h/2 | M_01-03-00-000 |
| Modbus | Modbus Parameter Table | M_01-03-00-000 |

Description

This parameter configures the drive for the induction motor's rated voltage as indicated on the nameplate. MOTOR.VOLTRATED is used to calculate the constant Volts per Hertz characteristics of the drive and motor.

Related Topics

Motor

MOTOR.VRATED

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the motor rated velocity (not maximum velocity) |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEARUNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, μ m/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 8,000.000 mm/s 0.000 to 8,000,000.000 μ m/s 0.000 to 1,250.000 custom units/s |
| Default Value | 0 rpm |
| Data Type | U16 |
| See Also | MOTOR.VOLTRATED, MOTOR.VOLTMIN |
| Start Version | M_01-03-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|------------------------|----------------------|
| EtherCAT COE and CANopen | 3457h/1 | M_01-03-00-000 |
| Modbus | Modbus Parameter Table | M_01-03-00-000 |

Description

This parameter configures the drive for the rated velocity of the induction motor as indicated on the nameplate. MOTOR.VRATED is used to calculate the constant volts per Hertz characteristics of the drive and motor.

Related Topics

Motor

This page intentionally left blank.

MT Parameters and Commands

MT.ACC

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Specifies motion task acceleration; active in opmode 2 (position) only. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s ² , (custom units)/s ² , rad/s ² Linear: counts/s ² , mm/s ² , μm/s ² , (custom units)/s ² |
| Range | Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s ² 0.155 to 4,166,666.752 (custom units)/s ² 0.012 to 5,235,987.968 rad/s ² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s ² 0.031*MOTOR.PITCH to 833,333.333*MOTOR.PITCH mm/s ² 30.994*MOTOR.PITCH to 833,333,333.333*MOTOR.PITCH μm/s ² 0.155 to 4,166,666.667 (custom units)/s ² |
| Default Value | Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s ² 833.333 (custom units)/s ² 1,047.2 rad/s ² Linear: 715,840,000.000 counts/s ² 166.714*MOTOR.PITCHMOTOR.PITCH mm/s ² 166,714.191*MOTOR.PITCHMOTOR.PITCH μm/s ² 833.571 (custom units)/s ² |
| Data Type | Float |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | Profile specific SDOs/0 | M_01-00-00-000 |
| Modbus | 526 (64-bit) | M_01-03-00-000 |

Description

MT.ACC specifies the motion task acceleration and is used by the MT.SET and MT.LOAD command. This parameter is a temporary value, since a motion task is only set after a MT.SET command. The motion task acceleration is further limited by the maximum allowed acceleration DRV.ACC

A value of 0 for MT.ACC should not be used when setting a motion task via MT.SET because this value causes a validity check of the MT.SET command to fail.

A value of 0 for MT.ACC after an MT.LOAD command displays an empty (not initialized) motion task.

Related Topics

12.2 Motion Tasks

MT.CLEAR

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Clears motion tasks from the drive; active in opmode 2 (position) only. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC,MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 530 | M_01-03-00-000 |

Description

MT.CLEAR clears a motion task from the drive. This command needs one argument in order to clear a motion task. A motion task consists of the following parameters: MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC,MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT

A value of -1 clears all motion tasks in the drive (MT.CLEAR -1).

Example

MT.CLEAR 5: Clear motion task number 5.

After performing a command such as MT.PARAMS 5, the drive displays the following:

```
5 0.000 Counts 0.000 rpm 0 0.000 rpm/s 0.000 rpm/s 0 0 0
ms
```

A value of 0 for velocity, acceleration, or deceleration displays motion task as uninitialized.

Related Topics

12.2 Motion Tasks

MT.CNTL

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets motion task control word; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 0 to 4,294,967,295 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | MT.NUM , MT.P, MT.V, MT.ACCMT.V, MT.DEC, MT.TNUM, MT.MTNEXT MT.MTNEXT , MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35AFh/0 | M_01-00-00-000 |
| | 6086h/0 | |
| Modbus | 532 | M_01-03-00-000 |

Description

MT.CNTL specifies the motion task control word, which is used by the MT.SET and MT.LOAD commands. The control word describes the behavior of the motion task. This parameter is a temporary value, since a motion task is only set after an MT.SET command is issued.

Since this parameter is read bitwise, it can combine multiple functions into a single word. The meaning of each bit is described in the tables below.

Table 1: Motion Task (MT) Bit Descriptions

| Bit | Meaning | Description |
|-----|---------|--|
| 0 | 0x00001 | See Table 2: MT Type. |
| 1 | 0x00002 | |
| 2 | 0x00004 | |
| 3 | 0x00008 | |
| 4 | 0x00010 | If this bit is 0, then the next MT is not executed. If this bit is 1, then the next MT is executed. |
| 5 | 0x00020 | See Table 3: Next MT Start Type. |
| 6 | 0x00040 | |
| 7 | 0x00080 | |
| 8 | 0x00100 | |
| 9 | 0x00200 | See Table 4: MT Acceleration Type. |
| 10 | 0x00400 | |
| 11 | 0x00800 | |

| Bit | Meaning | Description |
|-----|---------|--|
| 12 | 0x01000 | Activates the override functionality for a trapezoidal MT. If this bit is 1, a motion task with override functionality must be activated (see bit 5). |
| 13 | 0x02000 | If this bit is 0, then an attempt to trigger any new motion task will be accepted while this motion task is currently running. If this bit is 1, then an attempt to trigger any new motion task will be denied while this motion task is currently running. |
| 14 | 0x04000 | If this bit is set, the motion task that is supposed to be started cannot be started from velocity 0. The motion can be started if a motion task already running will be interrupted. |
| 15 | 0x08000 | Reserved. |

Table 2: MT Type

| Bits 3, 2, 1, 0 | Description |
|-----------------|--|
| 0000 | Absolute. The target position is defined by the MT.P value. |
| 1000 | Reserved. |
| 0001 | Relative to Command Position. The target position is defined as: Target position = PL.CMD + MT.P |
| 0011 | Relative to Previous Target Position. The target position is defined as: Target position = Target position of the last motion task + MT.P |
| 0101 | Reserved. |
| 0111 | Relative to Feedback Position. The target position is defined as: Target position = PL.FB + MT.P |

Table 3: Next MT Start Type

| Bits 9, 8, 7, 6, 5 | Description |
|--------------------|--|
| 00000 | Switches over to next MT after stopping. After an MT ends, the next MT starts immediately. |
| 00001 | Switches over to next MT after stopping and delay. After an MT ends, the MT following time (MT.TNEXTelapse in order to start the next MT. |
| 11000 | Switches over to the next MT at next MT speed (change on the fly).When the target position of an MT is reached, the drive has already accelerated with the acceleration ramp of the next MT to the target velocity of the next MT. Thus, the drive begins the next MT at the next MT target velocity. The MT.TNEXT setting is ignored if adjusted. |

Table 4: MT Acceleration Type

| Bits 11, 10 | Description |
|-------------|--|
| 00 | Trapezoidal acceleration and deceleration. |

| Bits 11, 10 | Description |
|-------------|---|
| 01 | <p>1:1 motion profile table motion task. The drive follows the customer motion profile table without inserting a constant velocity phase between the acceleration and deceleration process. This setting allows the usage of non-symmetric velocity profiles.</p> <p>The MT.TNUM parameter defines which table to use for the 1:1 profile handling.</p> |
| 11 | <p>Standard motion profile table motion task. The drive accelerates according to the shape of the motion profile table by stepping through the first half of the customer table. Then the drive inserts a constant velocity phase until the brake point is reached. Finally, the drive decelerates by stepping through the second half of the customer profile table.</p> <p>The MT.TNUM parameter defines which table to use for the 1:1 profile handling. This mode allows also a change on the fly between motion tasks (see Table 3 above). See "AKD Customer Profile Application Note" on the Kollmorgen web site (www.kollmorgen.com) for additional details.</p> |

Related Topics

12.2 Motion Tasks

MT.CONTINUE

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Continues a stopped motion task; active in opmode 2 (position) only. |
| Units | N/A |
| Range | N/A |
| Default Value | 0 |
| Data Type | N/A |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACCMT.V, MT.DEC, MT.TNUM, MT.MTNEXT MT.MTNEXT , MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 534 | M_01-03-00-000 |

Description

MT.CONTINUE continues a motion task that has been stopped by the DRV.STOP command.

Related Topics

12.2 Motion Tasks

MT.DEC

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets motion task deceleration; active in opmode 2 (position) only. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rps/s, rpm/s, deg/s ² , (custom units)/s ² , rad/s ² Linear: counts/s ² , mm/s ² , μm/s ² , (custom units)/s ² |
| Range | Rotary: 0.002 to 833,333.333 rps/s 0.112 to 50,000,000.000 rpm/s 0.009 to 300,000,000.000 deg/s ² 0.155 to 4,166,666.752 (custom units)/s ² 0.012 to 5,235,987.968 rad/s ² Linear: 16,000.000 to 3,579,139,408,000.000 counts/s ² 0.031*MOTOR.PITCH to 833333.333*MOTOR.PITCH mm/s ² 30.994*MOTOR.PITCH to 833333333.333*MOTOR.PITCH μm/s ² 0.155 to 4166666.667 (custom units)/s ² |
| Default Value | Rotary: 166.669 rps/s 10,000.000 rpm/s 60,000.000 deg/s ² 833.333 (custom units)/s ² 1,047.2 rad/s ² Linear: 715,840,000.000 counts/s ² 166.714*MOTOR.PITCHMOTOR.PITCH mm/s ² 166,714.191*MOTOR.PITCHMOTOR.PITCH μm/s ² 833.571 (custom units)/s ² |
| Data Type | Float |
| See Also | MT.ACC, MT.NUM, MT.P, MT.V, MT.CNTL, MT.TNUM, MT.MTNEXT, MT.TNEXT, MT.SET, MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | Profile specific SDOs/0 | M_01-00-00-000 |
| Modbus | 536 (64-bit) | M_01-03-00-000 |

Description

MT.DEC specifies the motion task deceleration and is used by the MT.SET and MT.LOAD commands. This parameter is a temporary value, since a motion task is only set after an MT.SET command is issued. The motion task deceleration is further limited by the maximum allowed acceleration, DRV.DEC.

A value of 0 for MT.DEC should not be used when setting a motion task via MT.SET because this value causes a validity check of the MT.SET command to fail.

A value of 0 for MT.DEC after an MT.LOAD command displays an empty (not initialized) motion task.

Related Topics

12.2 Motion Tasks

MT.EMERGMT

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Selects a motion task to be triggered after an emergency stop procedure; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 1 to 128 |
| Default Value | 0 |
| Data Type | N/A |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 540 | M_01-03-00-000 |

Description

MT.EMERGMT selects the motion task to be triggered after an emergency stop procedure.

A value of -1 shows that no motion task must be started after a ramp-down procedure in a closed position loop mode of operation.

Related Topics

12.2 Motion Tasks

MT.HOMEREQUIRE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Defines if the axis is required to be homed before a motion task can be executed. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 1 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-03-00-005 |

Description

This parameter defines whether or not the axis must be homed before a motion task can be executed.

MT.HOMEREQUIRE = 0: The axis does not need to be homed before a motion task can execute ("Home Done" can be either true or false before a motion task can execute).

MT.HOMEREQUIRE = 1: Homing must be complete ("Home Done" is true) before a motion task can execute.

Related Topics

12.2 Motion Tasks

MT.LIST

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Lists all initialized motion tasks in the drive; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 0 |
| Default Value | N/A |
| Data Type | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| See Also | M_01-00-00-000 |
| Start Version | N/A |

Description

MT.LIST reads every initialized motion task from the drive. A motion task consists of the following parameters: MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , and MT.TNEXT.

A motion task is considered as initialized as soon as MT.V, MT.ACC, and MT.DEC of that specific motion task have values not equal to 0.

Related Topics

12.2 Motion Tasks

MT.LOAD

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Reads/loads a motion task number from the drive; active in opmode 2 (position) only. |
| Units | N/A |
| Range | N/A |
| Default Value | 0 |
| Data Type | N/A |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 542 | M_01-03-00-000 |

Description

MT.LOAD reads out a motion task number MT.NUM from the drive. A motion task consists of the following parameters: MT.NUM, MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT. These parameters belong to the motion task number MT.NUM and are refreshed by MT.LOAD.

Related Topics

12.2 Motion Tasks

MT.MOVE

| General Information | |
|---------------------|---|
| Type | Command |
| Description | Starts a motion task; active in opmode 2 (position) only. |
| Units | N/A |
| Range | N/A |
| Default Value | 0 |
| Data Type | N/A |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 544 | M_01-03-00-000 |

Description

MT.MOVE starts a motion task. This command needs one argument in order to start a motion task. The drive must be homed, otherwise the motion task will not start (see also HOME commands).

Example

MT.MOVE 3 -> Start motion task number 3.

Related Topics

12.2 Motion Tasks

Mode 2: Start Motion Task in 10.1 Digital Inputs and Outputs (see also **Modes 3: Motion Task Select Bit** and **Mode 4: Motion Task Start Selected** in this topic)

MT.MTNEXT

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Specifies following motion task number; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 0 to 128 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35BCh/0 | M_01-00-00-000 |
| Modbus | 546 | M_01-03-00-000 |

Description

MT.MTNEXT specifies the number of the following motion task and is used by the MT.SET and MT.LOAD command. This parameter is a temporary value. A motion task is only set after an MT.SET command.

The motion task control word can be selected so that a following motion task is executed after a first motion task. This parameter displays which motion task should be started after the first motion task.

Related Topics

12.2 Motion Tasks

MT.NUM

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the motion task number; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 0 to 128 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 365Bh/0 | M_01-00-00-000 |
| Modbus | 548 | M_01-03-00-000 |

Description

MT.NUM specifies the motion task number, which is used by the MT.SET and MT.LOAD commands. This parameter is a temporary value. A motion task is only set after an MT.SET command is issued.

Related Topics

12.2 Motion Tasks

MT.P

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the motion task position; active in opmode 2 (position) only. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR |
| Range | N/A |
| Default Value | 0 |
| Data Type | Float |
| See Also | MT.NUM , MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | Profile specific SDOs/0 | M_01-00-00-000 |
| Modbus | 550 (64-bit) | M_01-03-00-000 |

Description

MT.P specifies the motion task position, which is used by the MT.SET and MT.LOAD command. Depending on the motion task control word (MT.CNTL), the MT.P command can either be the target position of the motion task or a relative distance. This parameter is a temporary value. A motion task is only set after an MT.SET command.

Related Topics

12.2 Motion Tasks

MT.PARAMS

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Shows a motion task; active in opmode 2 (position) only. |
| Units | N/A |
| Range | N/A |
| Default Value | 0 |
| Data Type | N/A |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

Description

MT.PARAMS displays a motion task. This command needs one argument in order to show a motion task. If you enter MT.PARAMS without an argument, the drive returns the current or last active motion task.

Example

MT.PARAMS 5

The drive responds as follows:

```

7   5222.000 Counts   135.000 rpm   1   550.746 rpm/s   654.458 rpm/s
0   0     0 ms

```

Related Topics

12.2 Motion Tasks

MT.SET

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Sets the motion task in the drive; active in opmode 2 (position) only. |
| Units | N/A |
| Range | N/A |
| Default Value | 0 |
| Data Type | N/A |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3431h/0 | M_01-00-00-000 |
| Modbus | 554 | M_01-03-00-000 |

Description

MT.SET sends a motion task to the drive. A motion task consists of the following parameters: MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , and MT.TNEXT.

The motion task number (MT.NUM) with the parameters above is sent to the drive only after the MT.SET command.

Related Topics

12.2 Motion Tasks

MT.TNEXT

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Specifies following motion task time; active in opmode 2 (position) only. |
| Units | ms |
| Range | 0 to 65,535 ms |
| Default Value | 0 ms |
| Data Type | Integer |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35BDh/0 | M_01-00-00-000 |
| Modbus | 556 | M_01-03-00-000 |

Description

MT.TNEXT specifies the time that must elapse before starting a following motion task. This value is used by the MT.SET and MT.LOAD command. This parameter is a temporary value. A motion task is only set after an MT.SET command.

The motion task control word can be selected so that a following motion task is executed after a first motion task and this additional delay time.

Related Topics

12.2 Motion Tasks

MT.TNUM

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the motion task customer table number; active in opmode 2 (position) only. |
| Units | N/A |
| Range | 0 to 7 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | MT.NUM , MT.P, MT.V, MT.CNTL, MT.ACC, MT.DEC, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 558 | M_01-03-00-000 |

Description

MT.TNUM specifies the customer profile table and is used by the MT.SET and MT.LOAD command. This parameter is a temporary value. A motion task is only set after an MT.SET command.

The drive can have up to eight customer specific profile tables. The drive performs an S-curve acceleration with these profile tables. The shapes of these tables have an impact on the shape of the motion task acceleration and deceleration. The motion task control word specifies if a customer profile table is used or not.

This parameter has no impact when a trapezoidal motion task acceleration and deceleration profile is selected (see Table 4: MT Acceleration Type in the MT.CNTL description).

Related Topics

12.2 Motion Tasks

MT.TNVSAVE

| General Information | |
|----------------------|--|
| Type | Command |
| Description | Saves the motion profile tables to the nonvolatile memory. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | MT.TDWNLDS, MT.TDWNLDV |
| Start Version | M_01-04-00-000 |

Description

This command saves all motion profile tables, which are used for motion tasking, to the nonvolatile memory. The nonvolatile memory for these tables can be cleared by triggering this command while no motion profile table is available in the volatile memory (empty array in the volatile memory). The nonvolatile memory for these tables can be deleted as follows:

```
-->MT.TDWNLDS 1
-->MT.TDWNLDS 2
-->MT.TNVSAVE
```

The first two commands are needed in order to delete all motion profile tables in the volatile memory. The following MT.TNVSAVE command detects that there is no data available in the volatile memory and therefore deletes the nonvolatile memory sector.

WorkBench also uses this command for the firmware download procedure.

MT.TPOSWND

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the motion task target position window; active in opmode 2 (position) only. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, custom units, 16-bit counts Linear: counts, mm, μ m, custom units, 16-bit counts |
| Range | N/A |
| Default Value | 0.5 rev |
| Data Type | Float |
| See Also | DRV.MOTIONSTAT |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35C6h/0 | M_01-00-00-000 |
| Modbus | 560 (64-bit) | M_01-03-00-000 |

Description

Within DRV.MOTIONSTAT, MT.TPOSWND is used to indicate that the target position of a motion task has been reached. DRV.MOTIONSTAT displays a "Target Position Reached" bit as soon as the following statement becomes true:

$$\text{abs}(\text{actual_position} - \text{target_position}) < \text{MT.TPOSWND}$$

Related Topics

12.2 Motion Tasks

MT.TVELWND

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the motion task target velocity window; active in opmode 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: Counts/s, mm/s, µm/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 250.000*MOTOR.PITCH mm/s 0.000 to 250,000.000*MOTOR.PITCH µm/sec 0.000 to 1,250.000 custom units/s |
| Default Value | Rotary: 60.000 rpm 1.000 rps 359.999 deg/s 5.000 custom units/s 6.283 rad/s Linear: 0.001 counts/s 1.000*MOTOR.PITCH mm/s 999.998*MOTOR.PITCH µm/sec 5.000 custom units/s |
| Data Type | Float |
| See Also | DRV.MOTIONSTAT |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3856h/0 | M_01-00-00-000 |
| Modbus | 564 | M_01-03-00-000 |

Description

Within DRV.MOTIONSTAT, MT.TVELWND is used to indicate that the target velocity of a motion task has been reached. DRV.MOTIONSTAT displays a "Target Velocity Reached" bit as soon as the following statement becomes true:

$$(\text{target velocity} - \text{MT.TVELWND}) < \text{actual velocity} < (\text{target velocity} + \text{MT})$$

Related Topics

12.2 Motion Tasks

MT.V

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the motion task velocity; active in opmode 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, μ m/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 250.000*MOTOR.PITCH mm/s 0.000 to 250,000.000*MOTOR.PITCH μ m/sec 0.000 to 1,250.000 custom units/s |
| Default Value | 0 |
| Data Type | Float |
| See Also | MT.NUM , MT.P, MT.CNTL, MT.ACC, MT.DEC, MT.TNUM, MT.MTNEXT , MT.TNEXT, MT.SET , MT.LOAD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 6081h6081h | M_01-00-00-000 |
| Modbus | 566 | M_01-03-00-000 |

Description

MT.V specifies the motion task velocity, which is used by the MT.SET and MT.LOAD command. This parameter is a temporary value. A motion task is only set after an MT.SET command. The motion task velocity is furthermore limited by VL.LIMITP or VL.LIMITN depending on the direction of the motion task.

A value of 0 should not be used when setting a motion task via MT.SET because this value causes a validity check of the MT.SET command to fail.

A value of 0 after an MT.LOAD command displays an empty (not initialized) motion task.

Related Topics

12.2 Motion Tasks

MT.VCMD

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the derivative of PL.CMD; active in opmode 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 568 | M_01-03-00-000 |

Description

MT.VCMD returns the derivative of the position loop trajectory (PL.CMD), which is therefore a velocity. MT.VCMD is updated while the drive is in DRV.OPMODE 2 and is processing the following motion types:

- Motion tasking
- Homing
- Electronic gearing
- Service motion
- External trajectory coming from a fieldbus
- External trajectory calculated from an analog input signal

Related Topics

12.2 Motion Tasks

PL Parameters

PL.CMD

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the position command directly from the entry to the position loop. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEARUNIT.ACCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, μm , (custom units), 16-bit counts |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | PL.FB |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 570 (64-bit) | M_01-03-00-000 |

Description

PL.CMD reads the position command as it is received in the position loop entry.

Related Topics

11.5 Position Loop

10.4 Analog Input

PL.ERR

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the position error present when the drive is controlling the position loop. |
| Units | counts, rad, deg, (custom units) |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | PL.FB |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | 35C5h/0 | M_01-00-00-000 |
| | Profile specific SDOs/0 | |
| Modbus | 574 (64-bit) | M_01-03-00-000 |

Description

PL.ERR reads the position error present when the drive is controlling the position loop. PL.ERR is the difference between the actual position of the motor shaft (PL.FB) and the commanded position of the drive (PL.CMD). If the drive is not in the position operating mode (DRV.OPMODE = 2), then the PL.ERR value is not generated by the drive and this parameter is read as 0.

Related Topics

11.5 Position Loop

PL.ERRFTHRESH

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the maximum position error. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, μm , (custom units), 16-bit counts |
| Range | Rotary: 0.000 to 5,123,372,000,000,005.000 counts 0.000 to 7,495,067.136 rad 0.000 to 429,436,076.032 deg 0.000 to 5,964,389.888 (custom units) 0.000 to 78,176,452,636.718 16-bit counts Linear: 0.000 to 5,123,372,000,000,005.000 counts 0.000 to 1,192,877.952*MOTOR.PITCH mm 0.000 to 1,192,878,014.464*MOTOR.PITCH μm 0.000 to 5,964,389.888 (custom units) 0.000 to 78,176,452,636.718 16-bit counts |
| Default Value | Rotary: 42,949,672,960.000 counts 62.832 rad 3,600.000 deg 50.000 (custom units) 655,360.000 16-bit counts Linear: 42,949,672,960.000 counts 10.000*MOTOR.PITCHMOTOR.PITCHmm 10,000.000*MOTOR.PITCH μm 50.000 (custom units) 655,360.000 16-bit counts |
| Data Type | Float |
| See Also | PL.ERR |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35C7h/0 | M_01-00-00-000 |
| | 6065h/0 | |
| Modbus | 580 (64-bit) | M_01-03-00-000 |

Description

This parameter sets the maximum position error. If the position error PL.ERR is larger than PL.ERRFTHRESH the drive generates a fault. If PL.ERRFTHRESH is set to 0, the maximum position error is ignored.

Example

Set position rotary units to 2 (degrees). Setting PL.ERRFTHRESH to 1000 states that is the position error is larger than 1000 degrees, the drive will generate a fault.

UNIT.PROTARY 2

PL.ERRFTHRESH 1000

Related Topics

11.5 Position Loop

PL.ERRMODE

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the type of following error warning and fault usage. |
| Units | 0- Standard following error 1-Enhanced following error |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | PL.ERR, PL.ERRFTHRESH, PL.ERRWTHRESH |
| Start Version | M_01-02-09-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 578 | M_01-03-00-000 |

Description

PL.ERRMODE sets the type of following error warning and fault usage.

Mode 0 - following error magnitude fault

In Mode 0, the values of PL.ERRFTHRESH and PL.ERRWTHRESH are compared against the value of PL.ERR. If the absolute value of PL.ERR is larger than PL.ERRWTHRESH, then a warning is generated. If the absolute value of PL.ERR is larger than PL.ERRFTHRESH, then a fault is generated.

Mode 1 - deviation from predicted trajectory fault

In Mode 1, the values of PL.ERRFTHRESH and PL.ERRWTHRESH are compared against the following value:

$$\langle \text{error} \rangle = \text{abs}(\text{PL.ERR} - [(\text{VL.CMD} - \text{VL.FF}) / \text{PL.KP}])$$

If the absolute value of $\langle \text{error} \rangle$ is larger than PL.ERRWTHRESH for a consecutive period of 100 ms, then a warning is generated. If the absolute value of $\langle \text{error} \rangle$ is larger than PL.ERRFTHRESH for a consecutive period of 100 ms, then a fault is generated.

In mode 1, if PL.KI is not 0 then the following error prediction mechanism is turned off. When the drive is disabled, the following error limit tests are turned off and the warnings are cleared. A value of 0 in PL.ERRFTHRESH or PL.ERRWTHRESH disables the respective functionality.

Example

Assuming

PL.ERRMODE = 0, PL.ERRFTHRESH=1.2, PL.ERRWTHRESH=1, then PL.ERR reads 1.1.

In this case the warning is generated, but the fault is not.

Assuming PL.ERRMODE = 0, PL.ERRFTHRESH=1.2, PL.ERRWTHRESH=1, then PL.ERR reads 1.3.

In this case the warning is generated, as well as the fault.

Related Topics

11.5 Position Loop

PL.ERRWTHRESH

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the position error warning level. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEARUNIT.ACCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, μ m, (custom units), 16-bit counts |
| Range | Rotary: 0.000 to 5,123,372,000,000,005.000 counts 0.000 to 7,495,067.136 rad 0.000 to 429,436,076.032 deg 0.000 to 5,964,389.888 (custom units) 0.000 to 78,176,452,636.718 16-bit counts Linear: 0.000 to 5,123,372,000,000,005.000 counts 0.000 to 1,192,877.952*MOTOR.PITCH mm 0.000 to 1,192,878,014.464*MOTOR.PITCH μ m 0.000 to 5,964,389.888 (custom units) 0.000 to 78,176,452,636.718 16-bit counts |
| Default Value | 0.000 deg |
| Data Type | Float |
| See Also | PL.ERR |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3483h/0 | M_01-00-00-000 |
| Modbus | 580 (64-bit) | M_01-03-00-000 |

Description

If this value is not equal 0 and the position error PL.ERR is larger than this value, the drive will generate a warning. If PL.ERRWTHRESH is set to 0 the warning is not issued.

Example

Set position rotary units to 2 degrees. If you set PL.ERRWTHRESH to 100 and the position error is larger than 100 degrees, then the drive will generate a warning.

UNIT.PROTARY 2

PL.ERRWTHRESH 100

Related Topics

11.5 Position Loop

PL.FB

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the position feedback value. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEARUNIT.ACCLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, μm , (custom units), 16-bit counts |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | FB1.OFFSET |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 6064h6064h/0 | M_01-00-00-000 |
| Modbus | 588 (64-bit) | M_01-03-00-000 |

Description

PL.FB returns the position feedback value.

Note that this value is not the pure feedback value read from the feedback device, but also includes the value of the FB1.OFFSET and an internal offset set automatically by the FW when a homing switch is actuated.

Related Topics

11.5 Position Loop

Selecting and Using Homing Modes

PL.FBSOURCE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the feedback source for the position loop. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | VL.FBSOURCE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 592 | M_01-03-00-000 |

Description

This parameter determines the feedback source that the position loop uses. A value of 0 for this parameter selects the primary feedback, a value of 1 selects the secondary feedback. If you use the secondary feedback as the source for the position loop, then FB2.MODE mode should be set as 0 (A/B signals). A/B signals are the only supported feedback type as secondary feedback into the position loop. Other settings for FB2.MODE are intended as pulse inputs or a gearing command when PL.FBSOURCE remains 0.

Related Topics

11.5 Position Loop

PL.INTINMAX

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Limits the input of the position loop integrator by setting the input saturation. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, μm , (custom units), 16-bit counts |
| Range | Rotary: 0.000 to 18,446,744,073,709.000 counts 0.000 to 26,986.052 rad 0.000 to 1,546,188.288 deg 0.000 to 21,474.836 (custom units) 0.000 to 281,474,976.710 16-bit counts Linear: 0.000 to 18,446,744,073,709.000 counts 0.000 to 4,294.968*MOTOR.PITCH mm 0.000 to 4,294,967.296*MOTOR.PITCH μm 0.000 to 21,474.836 (custom units) 0.000 to 281,474,976.710 16-bit counts |
| Default Value | Rotary: 3,999,989,760.000 counts 5.852 rad 335.275 deg 4.657 (custom units) 61,035.000 16-bit counts Linear: 3,999,989,760.000 counts 0MOTOR.PITCH mm 9MOTOR.PITCH μm 4.657 (custom units) 61,035.000 16-bit counts |
| Data Type | Float |
| See Also | PL.FB |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3481h/1 | M_01-00-00-000 |
| Modbus | 594 (64-bit) | M_01-03-00-000 |

Description

PL.INTINMAX limits the input of the position loop integrator by setting the input saturation. When used in concert with PL.INSATOUT, this variable allows you to make the position loop integrator effective near the target position. Far from the target position, however, the integrator is not dominant in the loop dynamics.

Related Topics

11.5 Position Loop

PL.INTOUTMAX

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Limits the output of the position loop integrator by setting the output saturation. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR Rotary: counts, rad, deg, (custom units), 16-bit counts Linear: counts, mm, μm , (custom units), 16-bit counts |
| Range | Rotary: 0.000 to 18,446,744,073,709.000 counts 0.000 to 26,986.052 rad 0.000 to 1,546,188.288 deg 0.000 to 21,474.836 (custom units) 0.000 to 281,474,976.710 counts 16 bit Linear: 0.000 to 18,446,744,073,709.000 counts 0.000 to 4,294.968*MOTOR.PITCH mm 0.000 to 4,294,967.296*MOTOR.PITCH μm 0.000 to 21,474.836 (custom units) 0.000 to 281,474,976.710 16-bit counts |
| Default Value | Rotary: 3,999,989,760.000 counts 5.852 rad 335.275 deg 4.657 (custom units) 61,035.000 16-bit counts Linear: 3,999,989,760.000 counts 0MOTOR.PITCH mm 9MOTOR.PITCH μm 4.657 (custom units) 61,035.000 16-bit counts |
| Data Type | Float |
| See Also | PL.INTINMAX |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3481h/2 | M_01-00-00-000 |
| Modbus | 598 (64-bit) | M_01-03-00-000 |

Description

PL.INTOUTMAX limits the output of the position loop integrator by setting the output saturation.

When used in concert with PL.INTINMAX, this variable allows you to make the position loop integrator effective near the target position. Far from the target position, however, the integrator is not dominant in the loop dynamics.

Related Topics

11.5 Position Loop

PL.KI

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the integral gain of the position loop. |
| Units | Hz |
| Range | 0 to 250 Hz |
| Default Value | 0 Hz |
| Data Type | Float |
| See Also | PL.KP, PL.KD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3480h/0 | M_01-00-00-000 |
| Modbus | 602 | M_01-03-00-000 |

Description

PL.KI sets the integral gain of the position regulator PID loop.

Related Topics

11.5 Position Loop

PL.KP

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the proportional gain of the position regulator PID loop. |
| Units | (rev/s)/rev |
| Range | 0 to 2,147,483.008 (rev/s)/rev |
| Default Value | 100 rps/rev |
| Data Type | Float |
| See Also | PL.KI, PL.KD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3542h/0 | M_01-00-00-000 |
| Modbus | 604 | M_01-03-00-000 |

Description

PL.KP sets the proportional gain of the position regulator PID loop.

Related Topics

11.5 Position Loop

PL.MODP1

| General Information | |
|----------------------|--|
| Type | R/W parameter |
| Description | Sets modulo range parameter. |
| Units | Depends on UNIT.PROTARY and UNIT.PLINEAR |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3637h/0 | M_01-00-00-000 |
| Modbus | 606 (64-bit) | M_01-03-00-000 |

Description

This parameter is either the beginning or the end of the modulo range, depending on whether this value is smaller or larger than PL.MODP2. If you set PL.MODP1 equal to PL.MODP2, an error message occurs.

| Condition | Beginning of the modulo-range | End of the modulo-range |
|---------------------|-------------------------------|-------------------------|
| PL.MODP1 < PL.MODP2 | PL.MODP1 | PL.MODP2 |
| PL.MODP2 < PL.MODP1 | PL.MODP2 | PL.MODP1 |

Related Topics

11.5 Position Loop

PL.MODP2

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the beginning or end modulo range parameter. |
| Units | Depends on UNIT.PROTARY and UNIT.PLINEAR. |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3638h/0 | M_01-00-00-000 |
| Modbus | 610 (64-bit) | M_01-03-00-000 |

Description

This parameter is either the beginning or the end of the modulo range, depending on whether this value is smaller or larger than PL.MODP1.

| Condition | Beginning of the modulo range | End of the modulo range |
|---------------------|-------------------------------|-------------------------|
| PL.MODP1 < PL.MODP2 | PL.MODP1 | PL.MODP2 |
| PL.MODP2 < PL.MODP1 | PL.MODP2 | PL.MODP1 |

Related Topics

11.5 Position Loop

PL.MODPDIR

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the direction for absolute motion tasks. |
| Units | N/A |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3430h/0 | M_01-00-00-000 |
| Modbus | 614 | M_01-03-00-000 |

Description

This parameter defines the direction of an absolute motion task when the modulo position has been activated. For more details about absolute motion tasks, see .12.2 Motion Tasks. For absolute motion tasks, you can only select a target position within the modulo range.

PL.MODPDIR Settings

| Value | Motion | Description |
|-------|-------------------|--|
| 0 | Inside Range | The motor moves in a negative direction if the target position of the absolute motion task is less than the current position. The motor moves in positive direction if the target position of the absolute motion task is greater than the current position. |
| 1 | Positive | The motor always moves in a positive direction relative to the target position of the absolute motion task. |
| 2 | Negative | The motor always moves in a negative direction relative to the target position of the absolute motion task. |
| 3 | Shortest Distance | The motor always moves the shortest distance in order to reach the target position within the modulo-range. |

PL.MODPEN

| General Information | |
|----------------------|------------------------------|
| Type | R/W Parameter |
| Description | Enables the modulo position. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

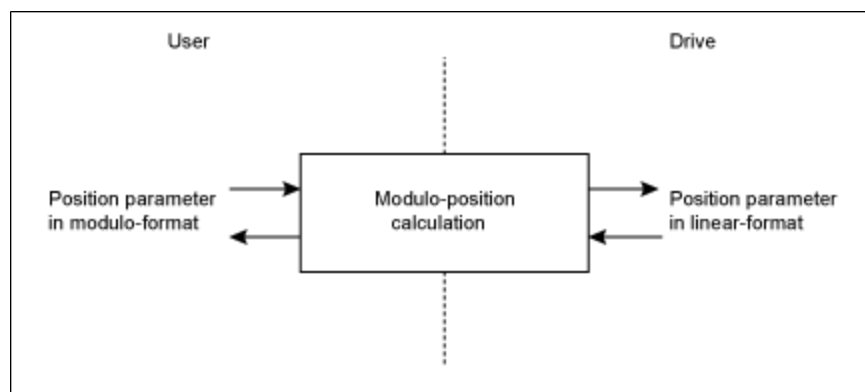
| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35CFh/0 | M_01-00-00-000 |
| Modbus | 616 | M_01-03-00-000 |

Description

A value of 0 disables the modulo-position and a value of 1 enables the modulo-position feature. The modulo-position feature can be used for circular applications such as round tables.

The position loop of the drive uses always a linear position variable but the data-exchange between the user and the drive uses the modulo-position calculation in order to convert values from linear format into modulo format and vice versa.

The following figure shows the interface between the user and the drive for PL.MODPEN=1:



Related Topics

11.5 Position Loop

PLS Parameters and Commands

PLS.EN

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Enables programmable limit switch (PLS). |
| Units | N/A |
| Range | 0 to 255 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | PLS.MODE, PLS.RESET, PLS.STATE, PLS.UNITS, PLS.P1 to PLS.P8, PLS.WIDTH1 to PLS.WIDTH8, PLS.T1 to PLS.T8 |
| Start Version | M_01-02-03-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 34A3h/1 | M_01-02-03-000 |
| Modbus | 618 | M_01-03-00-000 |

Description

PLS.EN is a bit variable which determines the mode of an individual PLS. Eight PLSs are available in the drive.

Example

| Bit Value | Behavior |
|------------------|----------------|
| Bit 0 = 0 | Disables PLS 1 |
| Bit 0 = 1 | Enables PLS 1 |
| Bit 7 = 0 | Disables PLS 8 |
| Bit 7 = 1 | Enables PLS 8 |

Related Topics

10.8 Programmable Limit Switch

PLS.MODE

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Selects programmable limit switch mode. |
| Units | N/A |
| Range | 0 to 255 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | PLS.EN, PLS.RESET, PLS.STATE, PLS.UNITS, PLS.P1 to PLS.P8, PLS.WIDTH1 to PLS.WIDTH8, PLS.T1 to PLS.T8 |
| Start Version | M_01-02-03-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 34A3h/3 | M_01-02-03-000 |
| Modbus | 620 | M_01-03-00-000 |

Description

PLS.MODE is a bit variable which determines the mode of an individual PLS. Eight PLSs are available in the drive.

Example

| Bit Value | Behavior |
|------------------|--|
| Bit 0 = 0 | PLS 1 is monitored continuously. |
| Bit 0 = 1 | PLS 1 is monitored until it is triggered once (single-shot method). The PLS observation can be re-armed using the PLS.RESET command. |
| Bit 7 = 0 | PLS 8 is monitored continuously. |
| Bit 7 = 1 | PLS 8 is monitored until it is triggered once (single-shot method). The PLS observation can be re-armed using the PLS.RESET command. |

Related Topics

10.8 Programmable Limit Switch

PLS.P1 TO PLS.P8

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the trigger point for programmable limit switches. |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR |
| Range | N/A |
| Default Value | 0 |
| Data Type | Float |
| See Also | UNIT.PROTARY |
| Start Version | M_01-02-03-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 34A0h/1 | PLS.P1 |
| | 34A0h/2 | PLS.P2 |
| | 34A0h/3 | PLS.P3 |
| | 34A0h/4 | PLS.P4 |
| | 34A0h/5 | PLS.P5 |
| | 34A0h/6 | PLS.P6 |
| | 34A0h/7 | PLS.P7 |
| | 34A0h/8 | PLS.P8 |
| Modbus | 622 (64-Bit) | PLS.P1 |
| | 626 (64-Bit) | PLS.P2 |
| | 630 (64-Bit) | PLS.P3 |
| | 634 (64-Bit) | PLS.P4 |
| | 638 (64-Bit) | PLS.P5 |
| | 642 (64-Bit) | PLS.P6 |
| | 646 (64-Bit) | PLS.P7 |
| | 650 (64-Bit) | PLS.P8 |

Description

PLS.P1 to PLS.P8 define the trigger point of the PLS. For further information about how these parameters affect PLS behavior, see the PLS.UNITS parameter description.

Related Topics

10.8 Programmable Limit Switch

PLS.RESET

| General Information | |
|----------------------|--|
| Type | W/O Parameter |
| Description | Resets programmable limit switch. |
| Units | N/A |
| Range | 0 to 255 |
| Default Value | N/A |
| Data Type | Integer |
| See Also | PLS.EN, PLS.MODE, PLS.STATE, PLS.UNITS, PLS.Px (x=1...8), PLS.WIDTHx (x=1...8), PLS.Tx (x=1...8) |
| Start Version | M_01-02-03-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 34A3h | M_01-02-03-000 |
| Modbus | 654 | M_01-03-00-000 |

Description

This parameter is a bit variable and is used in order to re-arm the corresponding PLS.STATE observation for another single-shot PLS use (see also PLS.MODE).

Example

| Bit Value | Behavior |
|------------------|--|
| Bit 0 = 0 | The PLS 1 observation (PLS.STATE bit 0) is not re-armed. |
| Bit 0 = 1 | The PLS 1 observation (PLS.STATE bit 0) is re-armed. |
| Bit 7 = 0 | The PLS 8 observation (PLS.STATE bit 7) is not re-armed. |
| Bit 7 = 1 | The PLS 8 observation (PLS.STATE bit 7) is re-armed. |

Related Topics

10.8 Programmable Limit Switch

PLS.STATE

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the programmable limit switch state. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | Integer |
| See Also | PLS.EN, PLS.RESET, PLS.UNITS, PLS.MODE, PLS.P1 TO PLS.P8, PLS.WIDTH1 TO PLS.WIDTH8, PLS.T1 TO PLS.T8 |
| Start Version | M_01-02-03-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 34A3h/4 | M_01-02-03-000 |
| Modbus | 656 | M_01-03-00-000 |

Description

This parameter is a bit variable and displays the current status of the individual programmable limit switches.

Example

Bit 0 = 0: Programmable Limit Switch 1 (PLS 1) is not active.

Bit 0 = 1: Programmable Limit Switch 1 (PLS 1) is active.

Bit 7 = 0: Programmable Limit Switch 8 (PLS 8) is not active.

Bit 7 = 1: Programmable Limit Switch 8 (PLS 8) is not active.

Related Topics

10.8 Programmable Limit Switch

PLS.T1 TO PLS.T8

| General Information | |
|----------------------|---|
| Type | R/W parameter |
| Description | Sets programmable limit switch time |
| Units | ms |
| Range | 0 to 65,536 ms |
| Default Value | 500 ms |
| Data Type | Integer |
| See Also | PLS.EN, PLS.RESET, PLS.STATE, PLS.UNITS, PLS.MODE, PLS.WIDTH1 TO PLS.WIDTH8, PLS.P1 TO PLS.P8 |
| Start Version | M_01-02-03-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 34A2h/1 | PLS.T1 |
| | 34A2h/2 | PLS.T2 |
| | 34A2h/3 | PLS.T3 |
| | 34A2h/4 | PLS.T4 |
| | 34A2h/5 | PLS.T5 |
| | 34A2h/6 | PLS.T6 |
| | 34A2h/7 | PLS.T7 |
| | 34A2h/8 | PLS.T8 |
| Modbus | 658 | PLS.T1 |
| | 660 | PLS.T2 |
| | 662 | PLS.T3 |
| | 664 | PLS.T4 |
| | 666 | PLS.T5 |
| | 668 | PLS.T6 |
| | 670 | PLS.T7 |
| | 672 | PLS.T8 |

Description

These parameters define the time of the PLS pulse for time-based PLS handling.

For further information about the PLS functionality, especially the meaning of the PLS.T1 to PLS.T8 parameter, refer to the PLS.UNITS parameter.

Related Topics

10.8 Programmable Limit Switch

PLS.UNITS

| General Information | |
|---------------------|--|
| Type | R/W parameter |
| Description | Sets programmable limit switch (PLS) units. |
| Units | N/A |
| Range | 0 to 255 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | PLS.EN, PLS.RESET, PLS.STATE, PLS.MODE, PLS.P1 TO PLS.P8, PLS.WIDTH1 TO PLS.WIDTH8, PLS.T1 TO PLS.T8 |
| Start Version | M_01-02-03-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 34A4h/0 | M_01-02-03-000 |
| Modbus | 674 | M_01-03-00-000 |

Description

PLS.UNITS is a bit variable which determines the behavior of the eight PLSs available in the drive. This parameter is used to select the units for the PLS pulse.

Examples

| Bit Value | Behavior |
|---|---|
| Bit 0 = 0 Position-based PLS handling. | The PLS.STATE parameter displays an active PLS 1 when the position is within the range of PLS.P1 + PLS.WIDTH1 ($PLS.P1 \leq PL.FB \leq PLS.P1 + PLS.WIDTH1$). When the parameter PLS.WIDTH1 has been set to the value of 0, this bit will be activated as soon as $PLS.FB \geq PL.P1$. |
| Bit 0 = 1 Time-based PLS handling. | After PLS.P1 is crossed, the PLS.STATE parameter displays an active PLS 1 for a PLS.T1 ms period of time. |
| Bit 7 = 0 Position-based PLS handling. | The PLS.STATE parameter displays an active PLS 8 when the position is within the range of PLS.P8 + PLS.WIDTH8 ($PLS.P8 \leq PL.FB \leq PLS.P8 + PLS.WIDTH8$). When the parameter PLS.WIDTH8 has been set to the value of 0, this bit will be activated as soon as $PLS.FB \geq PL.P8$. |
| Bit 7 = 1 Time-based PLS handling. | After PLS.P8 has been crossed, the PLS.STATE parameter displays an active PLS 8 for a PLS.T8 ms period of time. |

Continuous position-based PLS handling

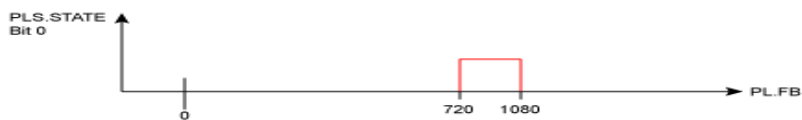
PLS.P1 = 720

PLS.WIDTH1 = 360

PLS.UNITS bit 0 (for PLS 1) = low; PLS.T1 is not considered.

PLS.EN bit 0 (for PLS 1) = high

PLS.MODE bit 0 (for PLS 1) = low



Time-based PLS handling

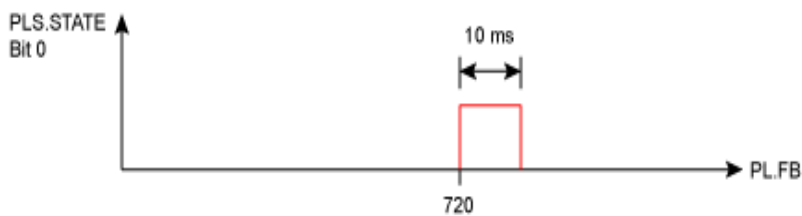
PLS.P1 = 720

PLS.T1 = 10

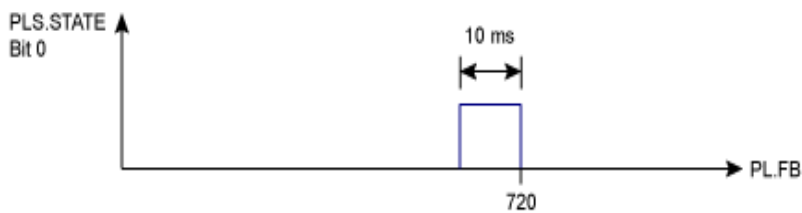
PLS.UNITS bit 0 (for PLS 1) = low; PLS.WIDTH1 is not considered.

PLS.EN bit 0 (for PLS 1) = high

PLS.MODE bit 0 (for PLS 1) = low



Positive direction of travel



Negative direction of travel

Related Topics

10.8 Programmable Limit Switch

PLS.WIDTH1 TO PLS.WIDTH8

| General Information | |
|----------------------|---|
| Type | R/W parameter |
| Description | Programmable Limit Switch Width |
| Units | Depends on UNIT.PROTARY or UNIT.PLINEAR |
| Range | N/A |
| Default Value | 0 |
| Data Type | Float |
| See Also | PLS.EN, PLS.RESET, PLS.STATE, PLS.UNITS, PLS.MODE, PLS.P1 TO PLS.P8, PLS.T1 TO PLS T8 |
| Start Version | M_01-02-03-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | 34A1h/1 PLS.WIDTH1 | M_01-02-03-000 |
| | 34A1h/2 PLS.WIDTH2 | |
| | 34A1h/3 PLS.WIDTH3 | |
| | 34A1h/4 PLS.WIDTH4 | |
| | 34A1h/5 PLS.WIDTH5 | |
| | 34A1h/6 PLS.WIDTH6 | |
| | 34A1h/7 PLS.WIDTH7 | |
| | 34A1h/8 PLS.WIDTH8 | |
| Modbus | 676 (64-Bit) PLS.WIDTH1 | M_01-03-00-000 |
| | 680 (64-Bit) PLS.WIDTH2 | |
| | 684 (64-Bit) PLS.WIDTH3 | |
| | 688 (64-Bit) PLS.WIDTH4 | |
| | 692 (64-Bit) PLS.WIDTH5 | |
| | 696 (64-Bit) PLS.WIDTH6 | |
| | 700 (64-Bit) PLS.WIDTH7 | |
| | 704 (64-Bit) PLS.WIDTH8 | |

Description

These parameter define the width of the PLS pulse for position-based PLS handling. For further information about the PLS functionality, especially the meaning of the PLS.WIDTH1 to PLS.WIDTH8 parameter, refer to the PLS.UNITS parameter.

Related Topics

10.8 Programmable Limit Switch

This page intentionally left blank.

REC Parameters and Commands

REC.ACTIVE

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Indicates if data recording is in progress (active). |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | N/A |
| Data Type | Boolean |
| See Also | REC.DONE, REC.OFF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 708 | M_01-03-00-000 |

Description

REC.ACTIVE indicates whether or not data recording is in progress. Recording is in progress if the trigger was met and the recorder is recording all data.

Related Topics

15 Scope

REC.CH1 to REC.CH6

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets recording channels 1 to 6. |
| Units | N/A |
| Range | N/A |
| Default Value | CH1 = IL.FB CH2 = IL.CMD CH3 = VL.FB CH4 = Empty CH5 = Empty CH6 = Empty |
| Data Type | String |
| See Also | REC.TRIG |
| Start Version | M_01-00-00-000 |

Description

REC.CHx specifies the recording channels.

There are 3 options to set the recording channels values:

- Set 0, CLR, or CLEAR. This setting clears the recording channel.
- Set one of the recordable commands. The list of recordable commands can be obtain by executing REC.RECPRMLIST.
- Set an internal value or variable of the drive (same as for DRV.MEMADDR input).

Related Topics

15 Scope

REC.DONE

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Checks whether or not the recorder has finished recording. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | N/A |
| Data Type | Boolean |
| See Also | REC.ACTIVE, REC.OFF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 710 | M_01-03-00-000 |

Description

REC.DONE indicates that the recorder has finished recording. This value is reset to 0 when the recorder trigger is set. The drive also resets this value when the recording has finished or when REC.OFF is executed.

Related Topics

15 Scope

REC.GAP

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Specifies the gap between consecutive samples. |
| Units | N/A |
| Range | 1 to 65,535 |
| Default Value | 1 |
| Data Type | Integer |
| See Also | REC.TRIG |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 712 | M_01-03-00-000 |

Description

REC.GAP specifies the gap between consecutive samples. The recording base rate is 16 kHz, thus a gap of 1 means that a sample is recorded every 62.5 μ s.

Related Topics

15 Scope

REC.NUMPOINTS

| General Information | |
|----------------------|--------------------------------------|
| Type | R/W Parameter |
| Description | Sets the number of points to record. |
| Units | N/A |
| Range | 1 to 65,535 |
| Default Value | 1,000 |
| Data Type | Integer |
| See Also | REC.TRIG |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 714 | M_01-03-00-000 |

Description

REC.NUMPOINTS specifies the number of points (samples) to record.

Related Topics

15 Scope

REC.OFF

| General Information | |
|----------------------|---------------------------------|
| Type | R/W Parameter |
| Description | Turns the recorder OFF. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | REC.ACTIVE, REC.DONE, REC.READY |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 716 | M_01-03-00-000 |

Description

REC.OFF turns the recorder off. In order to set the recorder again, the recorder must first be armed and then a trigger set.

Related Topics

15 Scope

REC.RECPRMLIST

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the list of recordable parameters. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | REC.CH1 to REC.CH6 |
| Start Version | M_01-00-00-000 |

Description

This command returns the list of recordable parameters. You can use a recordable parameter as an input to any of the recording channels.

Note that an internal address or a registered variable can be used as input to any of the channels in addition to the list.

Related Topics

15 Scope

REC.RETRIEVE

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Transfers all the recorded data to the communication channel. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | String |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

Description

REC.RETRIEVE causes the drive to transfer all the recorded data to the communication channel.

Example

The following format is the retrieve reply format (for N samples, G sample gap, and M parameters, where $M \leq 6$):

```
Recording
<N>, <G>
<parameter name 1> ... <parameter name M>
Value11 ... Value1M
Value N1 ... ValueNM
```

Related Topics

15 Scope

REC.RETRIEVEDATA

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Retrieves the recorded data without the header. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | REC.RETRIEVE, REC.RETRIEVEHDR, REC.RETRIEVESIZE |
| Start Version | M_01-00-00-000 |

Description

REC.RETRIEVEDATA retrieves a section of recorded data according to REC.RETRIEVESIZE from the received index; if no index is received, the drive retrieves the data from next section. An index is supplied to enable multiple retrieves and to give better control on the buffer in case of overflow. If no index or a negative value is present, then the index is ignored.

WorkBench uses this parameter to retrieve the data continuously for real time recoding.

The size of the data returned by this command depends on the number set by REC.RETRIEVESIZE.

Use REC.RETRIEVE for complete recording information view.

Notes:

- If REC.RETRIEVESIZE is larger than the buffer size, then it simply returns the whole buffer (no error).
- If the index is received, the data will be continuously returned starting from the given index (default starting index is 0).
- If the index is out of the bounds of the buffer, then it will be ignored.
- If recorder is active and REC.STOPTYPE==0, then this parameter returns an error.
- If REC.STOPTYPE==1, then this parameter returns the next section of data in the buffer (even if it reached the end of the buffer, it will return to the beginning of the buffer and add the data from index 0.)
- If REC.STOPTYPE==1 and the retrieve is too slow (gets overrun by the recorder), an overflow error message is returned instead of the retrieved data.
- If REC.STOPTYPE==0 and no index is received, continuously send the sections of data until the end of the buffer is reached. Then, return to the beginning of buffer and continue.
- A new REC.TRIG command automatically sets the index to 0.

Example

The following example retrieves data from index 100 in the size of 10 (hence places 100 to 109 in the buffer)

```
REC.NUMPOINTS 1000
REC.RETRIVESIZE 10
REC.TRIG
REC.RETRIEVEDATA 100
```

Related Topics

15 Scope

REC.RETRIEVEFRMT

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the format for recorded data output. |
| Units | N/A |
| Range | 0 to 1; 0 = Standard format, 1 = Internal format (high speed) |
| Default Value | 1 |
| Data Type | Integer |
| See Also | REC.RETRIEVE, REC.RETRIEVEDATA |
| Start Version | M_01-00-00-000 |

Description

Recorded data is transferred to the communication channel in one of two formats: standard or internal high speed. The standard (slower) format cannot be used for continuous recording, but is more easily read. The high speed format allows continuous data recording (needed for autotuning). WorkBench supports both formats.

Example

The following recorder data is in standard format:

```
10,1
IL.FB,VL.CMD,VL.FB
-0.086,0.000,2.661
0.000,0.000,3.605
0.029,0.000,-0.486
```

The following recorder data is in internal format:

```
10,1
IL.FB,VL.CMD,VL.FB
F3-0x56,F30x0,F30xA65
F30x0,F30x0,F30xE15
F30x1D,F30x0,F3-0x1E6
```

Related Topics

15 Scope

REC.RETRIEVEHDR

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Retrieves the recorded header without the data. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | REC.RETRIEVE, REC.RETRIEVEDATA |
| Start Version | M_01-00-00-000 |

Description

This command retrieves the recorded header without the data of the recording.

WorkBench uses this parameter to retrieve the header once before continuously reading the data for RT recoding.

Use REC.RETRIEVE for complete recording information view.

Related Topics

15 Scope

REC.RETRIEVESIZE

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the number of samples that REC.RETRIEVEDATA returns. |
| Units | recorder samples |
| Range | 0 to 65,535 recorder samples |
| Default Value | 1,000 recorder samples |
| Data Type | Integer |
| See Also | REC.RETRIEVEDATA, REC.RETRIEVEHDR |
| Start Version | M_01-00-00-000 |

Description

This parameter sets the number of samples that REC.RETRIEVEDATA returns.

WorkBench also uses this parameter to set the number of samples returned when retrieving the data continuously for RT recoding.

Use REC.RETRIEVE for the complete recording information view.

Related Topics

15 Scope

REC.STOPTYPE

| General Information | |
|----------------------|------------------------------------|
| Type | R/W Parameter |
| Description | Sets the recorder stop type. |
| Units | N/A |
| Range | 0 or 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | REC.RETRIEVEDATA, REC.RETRIEVESIZE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 718 | M_01-03-00-000 |

Description

This parameter sets the stop type for the recording.

0 = Recorder runs, continuously filling the recording circular buffer.

1 = Recorder fills in the buffer once.

To stop RT recording, execute REC.OFF.

Related Topics

15 Scope

REC.TRIG

| General Information | |
|----------------------|------------------------|
| Type | Command |
| Description | Triggers the recorder. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | REC.RETRIEVE, REC.OFF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 720 | M_01-03-00-000 |

Description

REC.TRIG starts the trigger according to the trigger type defined by REC.TRIGTYPE.

REC.TRIG sets the value of REC.DONE to 0.

After calling REC.TRIG, the data that was recorded by previous recording is deleted and cannot be retrieved.

No REC parameters can be set after a call to REC.TRIG until the recorder has finished or until REC.OFF is executed.

Related Topics

15 Scope

REC.TRIGPARAM

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the parameter that triggers the recorder. |
| Units | N/A |
| Range | N/A |
| Default Value | IL.FB |
| Data Type | String |
| See Also | REC.TRIG |
| Start Version | M_01-00-00-000 |

Description

REC.TRIGPARAM sets the parameter on which the recorder triggers.

This parameter is only used when REC.TRIGTYPE = 2.

Input values are:

1. One of the set drive parameters list that can be set as a trigger. The available parameters for trigger are: PL.ERR, PL.CMD, PL.FB, VL.CMD, VL.FB, IL.CMD, and IL.FB.
2. Internal value or variable of the drive (same as for DRV.MEMADDR input).

Related Topics

15 Scope

REC.TRIGPOS

| General Information | |
|---------------------|--|
| Type | R/W Parameter |
| Description | Sets the trigger position in the recording buffer. |
| Units | % |
| Range | 1 to 100% |
| Default Value | 10% |
| Data Type | Integer |
| See Also | REC.TRIG, REC.NUMPOINTS |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 722 | M_01-03-00-000 |

Description

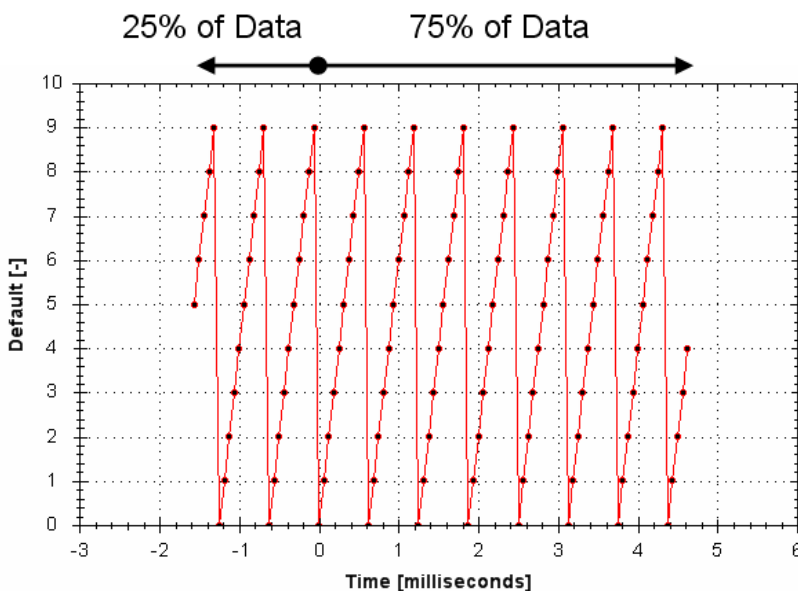
REC.TRIGPOS sets the trigger position in the recording buffer. The recording buffer size is defined by REC.NUMPOINTS. The input value is a percentage of the buffer (that is, a value of 25 means saving 25% of the buffer data before the trigger occurs and 75% after it occurs). This parameter is only used when REC.TRIGTYPE = 2 or 3.

Trigger Position

Trigger position (REC.TRIGPOS) allows you to collect data that occurs before the trigger occurs. In some instances, you may want to see the conditions prior to the trigger. Trigger position lets you control how much signal is collected before the trigger condition occurred .

Trigger position is specified in units of percent (%). If you specify a trigger position of x% , then x% of the data is before 0 ms in the data time and 100-x% (the rest of the data) is at or greater than 0 ms. In the figure below, the trigger position is set to 25% (REC.TRIGPOS 25).

In the WorkBench scope, the 0 time point is clear. When collecting the data using REC.RETRIEVE or similar commands, the time is not returned, so some caution should be used when the trigger point is important to understand.



Related Topics

15 Scope

REC.TRIGPRMLIST

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the list of possible trigger parameters. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | REC.TRIGPARAM |
| Start Version | M_01-00-00-000 |

Description

This command returns the list of trigger parameters. Each one of those parameters can serve as the trigger parameter (input to REC.TRIGPARAM).

Note that an internal address or a registered variable can be used as input to REC.TRIGPARAM in addition to the list that this parameter returns.

Related Topics

15 Scope

REC.TRIGSLOPE

| General Information | |
|----------------------|------------------------------|
| Type | R/W Parameter |
| Description | Sets the trigger slope. |
| Units | 0 = Negative 1 = Positive |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | REC.TRIG, REC.NUMPOINTS |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 726 | M_01-03-00-000 |

Description

REC.TRIGSLOPE sets the recorder trigger slope. This parameter is only used when REC.TRIGTYPE = 2 or 3.

Related Topics

15 Scope

REC.TRIGTYPE

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the trigger type. |
| Units | 0 = immediate 1 = command 2 = parameter 3 = boolean |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | REC.TRIG, REC.TRIGPARAM, REC.TRIGVAL, REC.TRIGSLOPE REC.TRIGPOS |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 728 | M_01-03-00-000 |

Description

REC.TRIGTYPE sets the type of trigger.

Input values are as follows:

| Value | Description |
|-------|---|
| 0 | Recording starts immediately |
| 1 | Recording starts on the next command executed through the TCP/IP. The trigger location in the buffer is set according to REC.TRIGPOS. |
| 2 | Recording starts per the values of REC.TRIGPARAM, REC.TRIGVAL, REC.TRIGSLOPE, and REC.TRIGPOS. |
| 3 | Recording starts when the value of REC.TRIGPARAM is 0 for REC.TRIGSLOPE = 0 or 1 for REC.TRIGSLOPE = 1 |

Related Topics

15 Scope

REC.TRIGVAL

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the trigger value. |
| Units | The units of the parameter are chosen according to the unit type. |
| Range | 0 to 2 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | REC.TRIG, REC.TRIGPARAM, REC.TRIGVAL, REC.TRIGSLOPE, REC.TRIGPOS |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 730 (64-bit) | M_01-03-00-000 |

Description

REC.TRIGVAL is the value that must be met by REC.TRIGPARAM for the trigger to occur. The units of this parameter are set according to the units of REC.TRIGPARAM.

Related Topics

15 Scope

REGEN Parameters

REGEN.POWER

| General Information | |
|---------------------|--|
| Type | R/O parameter |
| Description | Reads regen resistor's calculated power. |
| Units | Watt |
| Range | N/A |
| Default Value | N/A |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3416h/0 | M_01-00-00-000 |
| Modbus | 734 (64-bit) | M_01-03-00-000 |

Description

This parameter reads regen resistor's calculated power, which is determined as follows:

$$(V^2 / R) * DutyCycle$$

Related Topics

7.2 Regeneration

REGEN.REXT

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the external, user-defined regen resistor resistance. |
| Units | Ω |
| Range | 0 to 255 Ω |
| Default Value | 0 Ω |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 35C2h/0 | M_01-00-00-000 |
| Modbus | 738 | M_01-03-00-000 |

Description

REGEN.REXT sets the external user-defined regen resistor resistance. This variable is needed for the regen resistor temperature estimation algorithm.

Related Topics

7.2 Regeneration

REGEN.TEXT

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the external regen resistor thermal protection time constant. |
| Units | s |
| Range | 0.1 to 1,200 s |
| Default Value | 100 s |
| Data Type | Float |
| See Also | REGEN.WATTEXT, REGEN.REXT |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3415h/0 | M_01-00-00-000 |
| Modbus | 740 | M_01-03-00-000 |

Description

REGEN.TEXT is a thermal time constant used to protect an external regeneration (regen) resistor from overheating and failing. Its value is the time-to-fault when input power steps from 0 to 150% of REGEN.WATTEXT. The drive's regen resistor protection algorithm continuously calculates the power dissipated in the resistor and processes that power value through a single pole low pass filter to model the regen resistor's thermal inertia. When the filtered regen power on the output of the filter exceeds REGEN.WATTEXT, a fault occurs. REGEN.TEXT sets the time constant of this thermal inertia filter.

REGEN.TEXT can often be found directly on power resistor data sheets. On the data sheets, find the peak overload curve and then find the safe allowed time to be at 150% of the regen resistor's continuous power rating. Another way regen resistor peak overload capability is often specified is by giving the energy rating in joules of the resistor. If you have the energy rating E then:

$$\text{REGEN.TEXT} = (1.1) * (\text{joule limit} / \text{REGEN.WATTEXT})$$

Example

The external regen resistor is rated for 250 W continuous, is 33 ohm, and has a joule rating of 500 joules. To use this resistor, the drive settings become:

REGEN.TYPE = -1 (External Regen)

REGEN.REXT = 33

REGEN.WATTEXT = 250

REGEN.TEXT = $(1.1) * (500 \text{ j}) / (250 \text{ W}) = 2.2 \text{ sec}$

Related Topics

7.2 Regeneration

REGEN.TYPE

| General Information | |
|---|-------------------------------|
| Type | NV Parameter |
| Function | Sets the regen resistor type. |
| WorkBench Location (Screen/Dialog Box) | Power/Regen Resistor Type |
| Units | N/A |
| Range | -1 to 0 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3412h/0 | M_01-00-00-000 |
| Modbus | 742 | M_01-03-00-000 |

Description

You can specify a user-defined external regen resistor, select an internal regen resistor, or choose from a list of predefined regen resistors. The values for REGEN.TYPE are shown below:

| Type | Description |
|------|--------------------------------------|
| -1 | External user-defined regen resistor |
| 0 | Internal regen resistor |

If you specify a user-defined regen resistor, then you must also define this resistor's resistance (REGEN.REXT), heatup time (REGEN.REXT), and power (REGEN.WATTEXT).

Related Topics

7.2 Regeneration

REGEN.WATTEXT

| General Information | |
|----------------------|---|
| Type | R/W parameter |
| Description | Sets the regen resistor's power fault level for an external regen resistor. |
| Units | W |
| Range | 0 to 62,000 W |
| Default Value | 1000 W |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3414h/0 | M_01-00-00-000 |
| Modbus | 744 | M_01-03-00-000 |

Description

Sets the regen resistor's power fault level for an external regen resistor (when REGEN.TYPE = -1). Above this fault level, the regen resistor's PWM will be 0 and a fault will be issued.

Related Topics

7.2 Regeneration

This page intentionally left blank.

SM Parameters

SM.I1

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets service motion current 1; active in opmode 0 (torque) only. |
| Units | A |
| Range | –Drive peak current to +Drive peak current |
| Default Value | 0.025 · Drive peak current |
| Data Type | Float |
| See Also | SM.ACCTYPE, SM.I2, SM.MODE, SM.MOVE, SM.T1, SM.T2, SM.V1, SM.V2, SM.VPM1, SM.VPM2 |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 746 | M_01-03-00-000 |

Description

SM.I1 defines the current that is used in service motion modes 0 and 1 (see SM.MODE).

Related Topics

12.4 Service Motion

SM.I2

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets service motion current 2; active in opmode 0 (torque) only. |
| Units | A |
| Range | -Drive peak current to +Drive peak current |
| Default Value | 0.025 · Drive peak current |
| Data Type | Float |
| See Also | SM.ACCTYPE, SM.I1, SM.MODE, SM.MOVE, SM.T1, SM.T2, SM.V1, SM.V2, SM.VPM1, SM.VPM2 |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 748 | M_01-03-00-000 |

Description

SM.I2 defines the current that is used in service motion mode 1 (see SM.MODE).

Related Topics

12.4 Service Motion

SM.MODE

| General Information | |
|---------------------|--|
| Type | R/W Parameter |
| Description | Sets the service motion mode. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | SM.I1, SM.I2, SM.MOVE, SM.T1 SM.T2, SM.V1, SM.V2, DRV.ACCDRV.DEC |
| Start Version | M_01-01-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 750 | M_01-03-00-000 |

Description

SM.MODE defines the mode of service motion for each loop. Two types of service motion are available :

1. A constant motion in one direction (endless or for a certain amount of time).
2. An alternating motion.

The possible modes for this parameter are described in the following table:

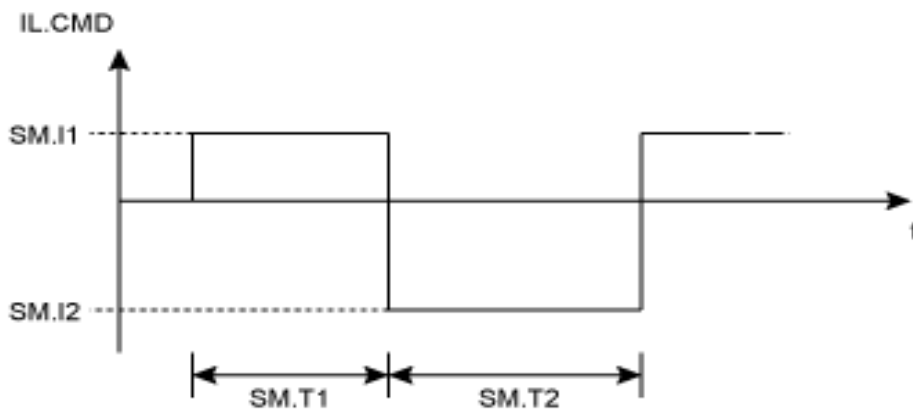
| SM.MODE | Description | Requirements |
|---------|--|--|
| 0 | <p>Constant motion in closed current loop mode of operation.</p> <ul style="list-style-type: none"> • DRV.OPMODE 0: The drive generates a constant current command value (SM.I1) for a certain amount of time (if SM.T1>0) or endless (if SM.T1=0). The drive will not generate any ramps in this mode of operation. • DRV.OPMODE 1 or 2: The drive generates a constant velocity command value (SM.V1) for a certain amount of time (if SM.T1>0) or endless (if SM.T1=0). The drive generates acceleration and deceleration ramps according to the DRV.ACC and DRV.DEC setting in this mode of operation. <p>The service motion can be stopped by using the DRV.STOP command.</p> | <p>DRV.OPMODE = 0, 1, or 2 DRV.CMDSOURCE = 0</p> |

| SM.MODE | Description | Requirements |
|---------|---|--|
| 1 | <ul style="list-style-type: none"> • DRV.OPMODE 0: The drive generates a current command value (SM.I1) for a certain amount of time (SM.T1). Afterwards the drive generates a current command value (SM.I2) for another certain amount of time (SM.T2). This sequence is repeated as long as a DRV.STOP command occurs. The drive will not generate any ramps in this mode of operation. • DRV.OPMODE 1 or 2: The drive generates a velocity command value (SM.V1) for a certain amount of time (SM.T1). Afterwards the drive generates a velocity command value (SM.V2) for another certain amount of time (SM.T2). This sequence is repeated as long as a DRV.STOP command occurs. The drive will generate an acceleration and deceleration ramps according to the DRV.ACC and DRV.DEC setting in this mode of operation. | <p>DRV.OPMODE = 0, 1, or 2 DRV.CMDSOURCE = 0</p> |

Ramps

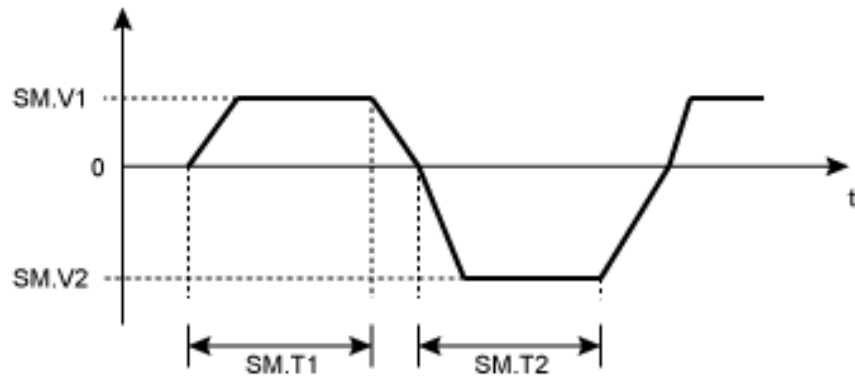
The drive uses DRV.ACC and DRV.DEC for the ramps in DRV.OPMODE 1 (closed velocity) and 2 (closed position). The drive does not generate any ramps in service motion mode 0 and 1.

Service Motion for DRV.OPMODE 0 and SM.MODE 1



Service motion for DRV.OPMODE 1 or 2 and SM.MODE 1

DRV.OPMODE 1: VL.CMD
DRV.OPMODE 2: d/dt (PL.CMD)



The deceleration process from SM.V1 or SM.V2 to 0 is not included in SM.T1 and SM.T2, respectively. SM.T1 and SM.T2 start as soon as the command value has reached the velocity 0.

Related Topics

12.4 Service Motion

SM.MOVE

| General Information | |
|---------------------|----------------------------|
| Type | Command |
| Description | Starts the service motion. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | SM.MODE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 752 | M_01-03-00-000 |

Description

This command starts the service motion that has been selected by the SM.MODE parameter.

Related Topics

12.4 Service Motion

SM.T1

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the service motion time 1. |
| Units | ms |
| Range | 0 to 65,535 ms |
| Default Value | 500 ms |
| Data Type | Integer |
| See Also | SM.I1, SM.I2, SM.MODE, SM.MOVE, SM.T2, SM.V1, SM.V2, SM.VPM1, SM.VPM2 |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 754 | M_01-03-00-000 |

Description

SM.T1 defines the time of the service motion that is used in all service motion modes (see SM.MODE). For an alternating service motion mode, SM.T1 may not be set to 0.

Related Topics

12.4 Service Motion

SM.T2

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the service motion time 2. |
| Units | ms |
| Range | 1 to 65,535 ms |
| Default Value | 500 ms |
| Data Type | Integer |
| See Also | SM.I1, SM.I2, SM.MODE, SM.MOVE, SM.T1, SM.V1, SM.V2, SM.VPM1, SM.VPM2 |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 756 | M_01-03-00-000 |

Description

SM.T2 defines the time of the service motion that is used in service motion modes 1, 3, and 5 (see SM.MODE).

Related Topics

12.4 Service Motion

SM.V1

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets service motion velocity 1; active in opmode 1 (velocity) and 2 (position). |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: Counts/s, mm/s, $\mu\text{m/s}$, custom units/s |
| Range | Rotary: -15,000.000 to 15,000.000 rpm -250.000 to 250.000 rps -90,000.000 to 90,000.000 deg/s -1,250.000 to 1,250.000 custom units/s -1,570.796 to 1,570.796 rad/s Linear: -1,073,741,824,000.000 to 1,073,741,824,000.000 counts/s -250.000*MOTOR.PITCH to 250.000*MOTOR.PITCH mm/s -250,000.000*MOTOR.PITCH to 250,000.000*MOTOR.PITCH $\mu\text{m/s}$ -1,250.000 to 1,250.000 custom units/s |
| Default Value | Rotary: 60.000 rpm 1.000 rps 359.999 deg/s 5.000 custom units/s 6.283 rad/s Linear: 0.001 Counts/s 1.000*MOTOR.PITCH mm/s 999.998*MOTOR.PITCH $\mu\text{m/sec}$ 5.000 custom units/s |
| Data Type | Float |
| See Also | SM.I1, SM.I2, SM.MODE, SM.MOVE, SM.T1, SM.T2, SM.V2 |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 758 | M_01-03-00-000 |

Description

SM.V1 defines the velocity that is used in service motion modes 0 and 1 (see SM.MODE) in the closed velocity and position mode of operation.

Related Topics

12.4 Service Motion

SM.V2

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets service motion velocity 2; active in opmode 1 (velocity) and 2 (position). |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, μ m/s, custom units/s |
| Range | Rotary: -15,000.000 to 15,000.000 rpm -250.000 to 250.000 rps -90,000.000 to 90,000.000 deg/s -1,250.000 to 1,250.000 custom units/s -1,570.796 to 1,570.796 rad/s Linear: -1,073,741,824,000.000 to 1,073,741,824,000.000 counts/s -250.000*MOTOR.PITCH to 250.000*MOTOR.PITCH mm/s -250,000.000*MOTOR.PITCH to 250,000.000*MOTOR.PITCH μ m/s -1,250.000 to 1,250.000 custom units/s |
| Default Value | Rotary: -60.000 rpm -1.000 rps -359.999 deg/s -5.000 custom units/s -6.283 rad/s Linear: -0.001 counts/s -1.000*MOTOR.PITCH mm/s -999.998*MOTOR.PITCH μ m/sec -5.000 custom units/s |
| Data Type | Float |
| See Also | SM.I1, SM.I2, SM.MODE, SM.MOVE, SM.T1, SM.T2, SM.V1 |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 760 | M_01-03-00-000 |

Description

SM.V2 defines the velocity that is used in service motion mode 1 (see SM.MODE) in the closed velocity and position mode of operation.

Related Topics

12.4 Service Motion

STO Parameters

STO.STATE

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Returns the status of the safe torque off. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | N/A |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 762 | M_01-03-00-000 |

Description

STO.STATE returns the status of the safe torque off.

- 1 - Safe torque on (no safe torque off fault).
- 0 - Safe torque off (safe torque off fault).

Related Topics

10.7.5 Limits

10.13 Safe Torque Off (STO)

SWLS Parameters

SWLS.EN

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Enables and disables software travel limit switches. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | DRV.MOTIONSTAT |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 764 | M_01-03-00-000 |

Description

This parameter enables the software travel limit switches. The software limit switches are only active if the axis is homed.

Example

Bit 0 = 0: Disable SWLS.LIMIT0

Bit 0 = 1: Enable SWLS.LIMIT0

Bit 1 = 0: Disable SWLS.LIMIT1

Bit 1 = 1: Enable SWLS.LIMIT1

Related Topics

10.7.5 Limits

12.1 Homing

HOME Parameters

SWLS.LIMIT0

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the position of the software travel limit switch 0. |
| Units | Position units |
| Range | -9,007,199,254,740,992 to 9,007,199,254,740,991 |
| Default Value | 0 |
| Data Type | S64 |
| See Also | UNIT.PROTARY, UNIT.PLINEAR |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | Profile specific SDOs/1 | M_01-00-00-000 |
| Modbus | 766 (64-bit) | M_01-03-00-000 |

Description

This parameter sets the compare register for the software limit switch 0. This value can be either the lower or the upper software limit switch register, depending on the configuration of the software limit switches. Whichever switch is set largest is the positive limit switch; the other switch becomes the negative limit switch. These switches can be used in addition to hardware limit switches. The software limit switches are only active if the axis is homed. For more information about homing, please refer to the HOME Parameters and DRV.MOTIONSTAT.

Related Topics

10.7.5 Limits

12.1 Homing

HOME Parameters

SWLS.LIMIT1

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the position of the software travel limit switch 0. |
| Units | Position units |
| Range | -9,007,199,254,740,992 to 9,007,199,254,740,991 |
| Default Value | 1,048,576.000 counts, 16-bit (firmware versions M_01-02-00-000 and above) 68,719,476,736 counts (for firmware version M_01-01-00-000) |
| Data Type | S64 |
| See Also | UNIT.PROTARY, UNIT.PLINEAR |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | Profile specific SDOs/2 | M_01-00-00-000 |
| Modbus | 770 (64-bit) | M_01-03-00-000 |

Description

This parameter sets the compare register for the software limit switch 1. This value can be either the lower or the upper software limit switch register, depending on the configuration of the software limit switches. Whichever switch is set largest is the positive limit switch; the other switch becomes the negative limit switch. These switches can be used in addition to hardware limit switches. The software limit switches are only active if the axis is homed. For more information about homing, please refer to the HOME Parameters and DRV.MOTIONSTAT.

Related Topics

10.7.5 Limits

12.1 Homing

HOME Parameters

SWLS.STATE

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the actual status of software limit switches. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 774 | M_01-03-00-000 |

Description

This parameter reads the status word of the software limit switches. The status word indicates the result of the compare between the software limit switch compare register and the actual position of the position loop.

Example

Bit 0 = 0: SWLS.LIMIT0 is not active.

Bit 0 = 1: SWLS.LIMIT0 is active.

Bit 1 = 0: SWLS.LIMIT1 is not active.

Bit 1 = 1: SWLS.LIMIT1 is active.

Bits 2 to 7 are currently not in use.

Related Topics

10.7.5 Limits

12.1 Homing

HOME Parameters

This page intentionally left blank.

UNIT Parameters

UNIT.ACCLINEAR

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the linear acceleration/deceleration units. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | DRV.ACC, DRV.DEC, MOTOR.TYPE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 776 | M_01-03-00-000 |

Description

UNIT.ACCLINEAR sets the units type for the acceleration and deceleration parameters, when the motor type (MOTOR.TYPE) is linear.

| Type | Description |
|------|--|
| 0 | [custom units]/s ² |
| 1 | millimeters per second squared (mm/s ²) |
| 2 | micrometers per second squared ($\mu\text{m/s}^2$) |
| 3 | Feedback counts/s ² |

Related Topics

9 Selecting Units for Your Application

UNIT.ACCROTARY

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the rotary acceleration/deceleration units. |
| Units | rpm/s, rps/s, deg/s ² , [custom units]/s ² |
| Range | 0 to 3 rpm/s |
| Default Value | 0 rpm/s |
| Data Type | Integer |
| See Also | DRV.ACC, "DRV.DEC" (=> p. 369), MOTOR.TYPE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3659h/0 | M_01-00-00-000 |
| Modbus | 778 | M_01-03-00-000 |

Description

UNIT.ACCROTARY sets the acceleration/deceleration units when the motor type (MOTOR.TYPE) is rotary.

| Type | Description |
|------|-------------------------------|
| 0 | rpm/s |
| 1 | rps/s |
| 2 | deg/s ² |
| 3 | (custom units)/s ² |

Related Topics

9 Selecting Units for Your Application

UNIT.LABEL

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets user-defined name for user-defined position units. |
| Units | N/A |
| Range | Maximum 16 characters, no spaces |
| Default Value | custom units |
| Data Type | String |
| See Also | UNIT.PLINEAR, UNIT.POUT |
| Start Version | M_01-00-00-000 |

Description

If you define a special position unit with UNIT.PLINEAR and UNIT.POUT, then you can give this unit a descriptive name. You can name the unit anything you wish, as long as the name is limited to 16 characters and includes no spaces. The label used for velocity and acceleration are in terms of this descriptive name.

This parameter is descriptive only and does not influence drive internal functions in any way.

Related Topics

9 Selecting Units for Your Application

UNIT.PIN

| General Information | |
|----------------------|---------------------------------------|
| Type | NV Parameter |
| Description | Sets gear IN for the unit conversion. |
| Units | User units |
| Range | 0 to 4,294,967,295 |
| Default Value | 100 |
| Data Type | Integer |
| See Also | UNIT.POUT |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | 35CAh/0 | M_01-00-00-000 |
| | Profile specific SDOs/1 | |
| Modbus | 780 | M_01-03-00-000 |

Description

UNIT.PIN is used in conjunction with UNIT.POUT to set application specific units. This parameter is used as follows in the drive unit conversion:

- For position, this parameter sets the units as [custom units]/rev.
- For velocity, this parameter sets the units as [custom units]/s.
- For acceleration/deceleration, this parameter sets the units as [custom units]/s².

Related Topics

9 Selecting Units for Your Application

UNIT.PLINEAR

| General Information | |
|----------------------|---------------------------------|
| Type | NV Parameter |
| Description | Sets the linear position units. |
| Units | N/A |
| Range | 0 to 4 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | PL.FB, PL.CMD, MOTOR.TYPE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 782 | M_01-03-00-000 |

Description

UNIT.PLINEAR sets the units type for the position parameters when the motor type (MOTOR.TYPE) is linear.

| Type | Description |
|------|-------------------------------|
| 0 | 32-bit counts |
| 1 | Millimeters (mm) |
| 2 | Micrometers (μm) |
| 3 | (PLINEAR/POUT) per revolution |
| 4 | 16-bit counts |

Related Topics

9 Selecting Units for Your Application

UNIT.POUT

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets gear out for the unit conversion. |
| Units | User units. |
| Range | 0 to 4,294,967,295 |
| Default Value | 20 |
| Data Type | Integer |
| See Also | UNIT.PLINEAR |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | 35CBh/0 | M_01-00-00-000 |
| | Profile specific SDOs/2 | |
| Modbus | 784 | M_01-03-00-000 |

Description

UNIT.POUT is used in conjunction with UNIT.PLINEAR to set application specific units in UNIT.POUT. This parameter is used as follows in the drive unit conversion:

- For position, this parameter sets the units as [custom units]/rev.
- For velocity, this parameter sets the units as [custom units]/s.
- For acceleration/deceleration, this parameter sets the units as [custom units]/s².

Related Topics

9 Selecting Units for Your Application

UNIT.PROTARY

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the position units when the motor type (MOTOR.TYPE) is rotary. |
| Units | counts, rad, deg, custom units, 16-bit counts |
| Range | 0 to 4 |
| Default Value | 4 16-bit counts (for firmware versions M_01-02-00-000 and above) 0 counts (for firmware version M_01-01-00-000) |
| Data Type | Integer |
| See Also | PL.FB, PL.CMD, MOTOR.TYPE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3660h/0 | M_01-00-00-000 |
| Modbus | 786 | M_01-03-00-000 |

Description

UNIT.PROTARY sets the position units when the motor type (MOTOR.TYPE) is rotary.

| Value | Units |
|-------|---------------|
| 0 | counts |
| 1 | radians |
| 2 | degrees |
| 3 | custom units |
| 4 | 16-bit counts |

Related Topics

9 Selecting Units for Your Application

UNIT.VLINEAR

| General Information | |
|----------------------|------------------------------------|
| Type | NV Parameter |
| Description | Sets the linear velocity units. |
| Units | N/A |
| Range | 0 to 3 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | VL.FB, VL.CMDU, VL.CMD, MOTOR.TYPE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 788 | M_01-03-00-000 |

Description

UNIT.VLINEAR sets the units type for the velocity parameters when the motor type (MOTOR.TYPE) is linear.

| Type | Description |
|------|---------------------------|
| 0 | (custom units) per second |
| 1 | Micrometers per second |
| 2 | Millimeters per second |
| 3 | Counts per second |

Related Topics

9 Selecting Units for Your Application

UNIT.VROTARY

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the velocity units when the motor type (MOTOR.TYPE) is rotary. |
| Units | rpm, rps, deg/s, (custom units)/s |
| Range | 0 to 3 |
| Default Value | 0 rpm |
| Data Type | Integer |
| See Also | VL.FB, VL.CMDU, VL.CMD, MOTOR.TYPE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 365Fh/0 | M_01-00-00-000 |
| Modbus | 790 | M_01-03-00-000 |

Description

UNIT.VROTARY sets the velocity units when the motor type (MOTOR.TYPE) is rotary.

| Value | Units |
|-------|------------------|
| 0 | rpm |
| 1 | rps |
| 2 | deg/s |
| 3 | (custom units)/s |

Related Topics

9 Selecting Units for Your Application

VBUS Parameters

VBUS.OVFTHRESH

| General Information | |
|----------------------|-------------------------------------|
| Type | R/O Parameter |
| Description | Reads the over voltage fault level. |
| Units | Vdc |
| Range | 0 to 900 Vdc |
| Default Value | N/A |
| Data Type | Integer |
| See Also | VBUS.UVFTHRESH |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 794 | M_01-03-00-000 |

Description

VBUS.OVFTHRESH reads the over voltage fault level for the DC bus.

This value is read from the drive EEPROM and varies according to the drive type.

Related Topics

7.2 Regeneration

VBUS.OVWTHRESH

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets voltage level for over voltage warning. |
| Units | Vdc |
| Range | 0 to 900 Vdc |
| Default Value | 0 Vdc (warning disabled) |
| Data Type | U16 |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 796 | M_01-03-00-000 |

Description

If VBUS.VALUE value exceeds VBUS.OVWTHRESH, then a warning is generated.

Related Topics

7.2 Regeneration

VBUS.RMSLIMIT

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the limit for the bus capacitors load. |
| Units | Vrms |
| Range | N/A |
| Default Value | N/A |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 798 | M_01-03-00-000 |

Description

This parameter reads the limit of the bus capacitor load. When the bus capacitor loads exceeds this limit, the drive generates fault F503.

Excessive bus capacitor load may indicate a disconnected main supply phase.

Related Topics

7.2 Regeneration

VBUS.UVFTHRESH

| General Information | |
|----------------------|-------------------------------------|
| Type | R/W Parameter |
| Description | Sets the under voltage fault level. |
| Units | Vdc |
| Range | 90 to 420 Vdc |
| Default Value | 90 Vdc |
| Data Type | Integer |
| See Also | VBUS.OVFTHRESH |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 800 | M_01-03-00-000 |

Description

VBUS.UVFTHRESH sets the undervoltage fault level of the DC bus.

The default value is read from the EEPROM, but can be modified by the user and stored on the NV RAM. This value varies according to drive type.

Related Topics

7.2 Regeneration

VBUS.UVMODE

| General Information | |
|----------------------|-----------------------------------|
| Type | N/V Parameter |
| Description | Indicates undervoltage (UV) mode. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 1 |
| Data Type | Boolean |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 802 | M_01-03-00-000 |

Description

This parameter indicates undervoltage (UV) mode.

When VBUS.UVMODE = 0, an undervoltage fault is issued whenever the DC bus goes below the undervoltage threshold.

When VBUS.UVMODE = 1, an undervoltage fault is issued whenever the DC bus goes below the under voltage threshold and the controller attempts to enable the drive (software or hardware enable).

Related Topics

7.2 Regeneration

VBUS.UVWTHRESH

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets voltage level for undervoltage warning. |
| Units | Vdc |
| Range | 0 to 900 Vdc |
| Default Value | 10 volts above the default value of the under voltage fault threshold (VBUS.UVFTHRESH). The default value of VBUS.UVFTHRESH is hardware dependent. |
| Data Type | U16 |
| See Also | VBUS.UVFTHRESH |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 804 | M_01-03-00-000 |

Description

If VBUS.VALUE value drops below VBUS.UVWTHRESH, then a warning is generated.

Related Topics

7.2 Regeneration

VBUS.VALUE

| General Information | |
|----------------------|-----------------------|
| Type | R/O Parameter |
| Description | Reads DC bus voltage. |
| Units | Vdc |
| Range | 0 to 900 Vdc |
| Default Value | N/A |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 361Ah/0 | M_01-00-00-000 |
| Modbus | 806 | M_01-03-00-000 |

Description

VBUS.VALUE reads the DC bus voltage.

Related Topics

7.2 Regeneration

7.2 Regeneration

VL Parameters

VL.ARPF1 TO VL.ARPF4

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the natural frequency of the pole (denominator) of anti-resonance (AR) filters 1, 2, 3, and 4; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Hz |
| Range | 5 to 5,000 Hz |
| Default Value | 500 Hz |
| Data Type | Float |
| See Also | VL.ARPQ1 TO VL.ARPQ4, VL.ARZF1 TO VL.ARZF4, Sets the Q of the zero (numerator) of anti-resonance filter #1; active in opmodes 1 (velocity) and 2 (position) only. |
| Start Version | M_01-02-00-000 |

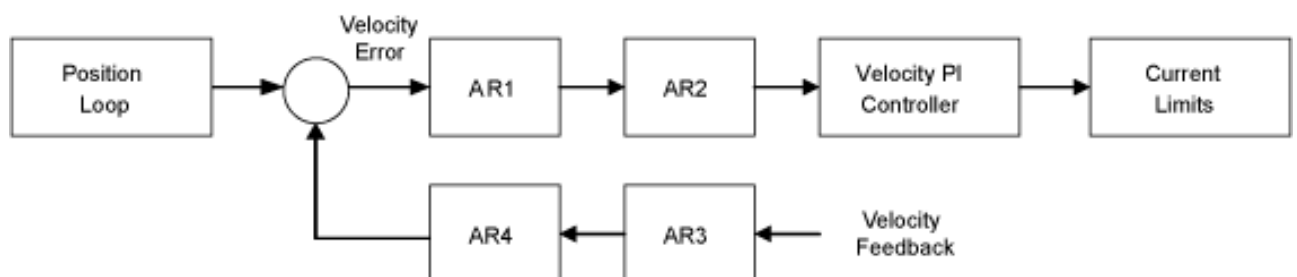
| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|------------------|----------------------|
| EtherCAT COE and CANopen | 3406h/1 VL.ARPF1 | M_01-02-00-000 |
| | 3406h/2 VL.ARPF2 | |
| | 3406h/3 VL.ARPF3 | |
| | 3406h/4 VL.ARPF4 | |
| Modbus | 808 VL.ARPF1 | M_01-03-00-000 |
| | 810 VL.ARPF2 | |
| | 812 VL.ARPF3 | |
| | 814 VL.ARPF4 | |

Description

VL.ARPF1 sets the natural frequency of the pole (denominator) of AR filter 1. This value is F_p in the approximate transfer function of the filter:

$$ARx(s) = [s^2/(2\pi F_z)^2 + s/(Q_z 2\pi F_z) + 1] / [s^2/(2\pi F_p)^2 + s/(Q_p 2\pi F_p) + 1]$$

The following block diagram describes the AR filter function; note that AR1 and AR2 are in the forward path, while AR3 and AR4 are applied to feedback:



AR1, AR2, AR3, and AR4 are used in velocity and position mode, but are disabled in torque mode.

Discrete time transfer function (applies to all AR filters)

The velocity loop compensation is actually implemented as a digital discrete time system function on the DSP. The continuous time transfer function is converted to the discrete time domain by a backward Euler mapping:

$$s \approx (1-z^{-1})/t, \text{ where } t = 62.5 \mu\text{s}$$

The poles are prewarped to F_p and the zeros are prewarped to F_z .

Related Topics

11.4 Velocity Loop

VL.ARPQ1 TO VL.ARPQ4

| General Information | |
|---------------------|---|
| Type | R/W Parameter |
| Description | Sets the Q of the pole (denominator) of anti-resonance (AR) filter 1; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | None |
| Range | 0.2 to 20 |
| Default Value | 0.5 |
| Data Type | Float |
| See Also | VL.ARPF1 TO VL.ARPF4, VL.ARZF1 TO VL.ARZF4, VL.ARZQ1 TO VL.ARZQ4 |
| Start Version | M_01-02-00-000 |

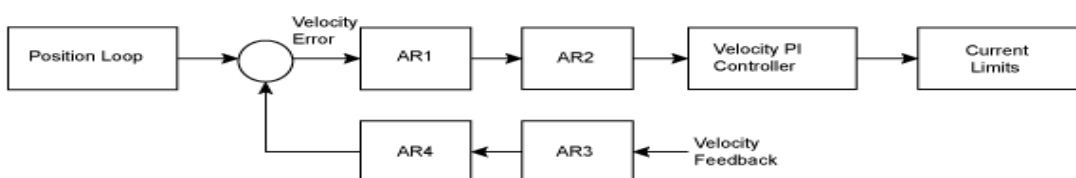
| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|------------------|----------------------|
| EtherCAT COE and CANopen | 3406h/5 VL.ARPQ1 | M_01-02-00-000 |
| | 3406h/6 VL.ARPQ2 | |
| | 3406h/7 VL.ARPQ3 | |
| | 3406h/8 VL.ARPQ4 | |
| Modbus | 816 VL.ARPQ1 | M_01-03-00-000 |
| | 818 VL.ARPQ2 | |
| | 820 VL.ARPQ3 | |
| | 822 VL.ARPQ4 | |

Description

VL.ARPQ1 sets the Q (quality factor) of the pole (denominator) of AR filter 1. This value is Q_p in the approximate transfer function of the filter:

$$AR_x(s) = [s^2/(2\pi F_z)^2 + s/(Q_z 2\pi F_z) + 1] / [s^2/(2\pi F_p)^2 + s/(Q_p 2\pi F_p) + 1]$$

The following block diagram describes the AR filter function; note that AR1 and AR2 are in the forward path, while AR3 and AR4 are applied to feedback:



AR1, AR2, AR3, and AR4 are used in velocity and position mode, but are disabled in torque mode.

Discrete time transfer function (applies to all AR filters)

The velocity loop compensation is actually implemented as a digital discrete time system function on the DSP. The continuous time transfer function is converted to the discrete time domain by a backward Euler mapping:

$$s \approx (1-z^{-1})/t, \text{ where } t = 62.5 \mu\text{s}$$

The poles are prewarped to F_p and the zeros are prewarped to F_z .

Related Topics

11.4 Velocity Loop

VL.ARTYPE1 TO VL.ARTYPE4

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Indicates the method used to calculate BiQuad coefficients; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | N/A |
| Range | 0 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|---------------------|----------------------|
| EtherCAT COE and CANopen | 3405h/1 VL.ARTYPE1 | M_01-00-00-000 |
| | 3405h/2 VL.A-RTYPE2 | |
| | 3405h/3 VL.A-RTYPE3 | |
| | 3405h/4 VL.A-RTYPE4 | |
| Modbus | 824 VL.ARTYPE1 | M_01-03-00-000 |
| | 826 VL.ARTYPE2 | |
| | 828 VL.ARTYPE3 | |
| | 830 VL.ARTYPE4 | |

Description

These parameters indicate the method used to calculate the biquad coefficients VL.ARPFx, VL.ARPQx, VL.ARZFx, and VL.ARZQx. A value of 0 indicates that the coefficients are set directly. This parameter has no effect on the filter itself, but is only used to determine the original design parameters. Currently, only the value of 0 is supported.

Related Topics

11.4 Velocity Loop

VL.ARZF1 TO VL.ARZF4

| General Information | |
|---------------------|--|
| Type | R/W Parameter |
| Description | Sets the natural frequency of the zero (numerator) of anti-resonance (AR)filter 1; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Hz |
| Range | 5 to 5,000 Hz |
| Default Value | 500 Hz |
| Data Type | Float |
| See Also | VL.ARPF1 TO VL.ARPF4, VL.ARPQ1 TO VL.ARPQ4, VL.ARZQ1 TO VL.ARZQ4 |
| Start Version | M_01-02-00-000 |

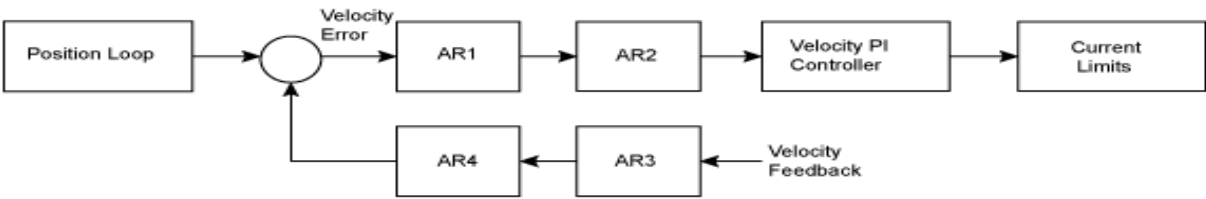
| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|------------------|----------------------|
| EtherCAT COE and CANopen | 3406h/9 VL.ARZF1 | M_01-02-00-000 |
| | 3406h/A VL.ARZF2 | |
| | 3406h/B VL.ARZF3 | |
| | 3406h/C VL.ARZF4 | |
| Modbus | 832 VL.ARZF1 | M_01-03-00-000 |
| | 834 VL.ARZF2 | |
| | 836 VL.ARZF3 | |
| | 838 VL.ARZF4 | |

Description

VL.ARZF1 sets the natural frequency of the zero (numerator) of AR filter 1. This value is F_z in the approximate transfer function of the filter:

$$ARx(s) = [s^2/(2\pi F_z)^2 + s/(Q_z 2\pi F_z) + 1] / [s^2/(2\pi F_p)^2 + s/(Q_p 2\pi F_p) + 1]$$

The following block diagram describes the AR filter function; note that AR1 and AR2 are in the forward path, while AR3 and AR4 are applied to feedback:



AR1, AR2, AR3, and AR4 are used in velocity and position mode, but are disabled in torque mode.

Discrete time transfer function (applies to all AR filters)

The velocity loop compensation is actually implemented as a digital discrete time system function on the DSP. The continuous time transfer function is converted to the discrete time domain by a backward Euler mapping:

$$s \approx (1-z^{-1})/t, \text{ where } t = 62.5 \mu s$$

The poles are prewarped to F_p and the zeros are prewarped to F_z .

Related Topics

11.4 Velocity Loop

VL.ARZQ1 TO VL.ARZQ4

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the Q of the zero (numerator) of anti-resonance filter #1; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | N/A |
| Range | 0.1 to 5 |
| Default Value | 0.5 |
| Data Type | Float |
| See Also | VL.ARPF1 TO VL.ARPF4, VL.ARPQ1 TO VL.ARPQ4, VL.ARZF1 TO VL.ARZF4 |
| Start Version | M_01-02-00-000 |

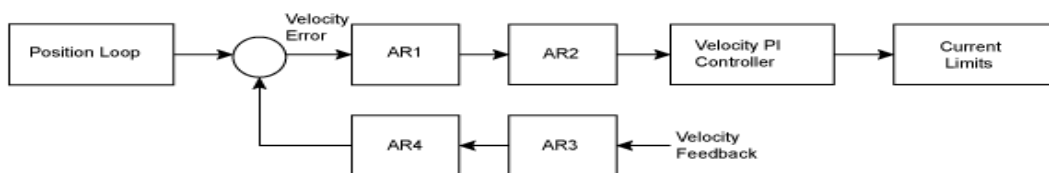
| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------|----------------------|
| EtherCAT COE and CANopen | 3406h VL.ARZQ1 | M_01-02-00-000 |
| | 3406h/E VL.ARZQ2 | |
| | 3406h VL.ARZQ3 | |
| | 3406h/10 VL.ARZQ4 | |
| Modbus | 832 VL.ARZQ1 | M_01-03-00-000 |
| | 834 VL.ARZQ2 | |
| | 836 VL.ARZQ3 | |
| | 838 VL.ARZQ4 | |

Description

VL.ARZQ1 sets the Q (quality factor) of the zero (numerator) of AR filter 1. This value is Q_z in the approximate transfer function of the filter:

$$AR1(s) = [s^2/(2\pi F_z)^2 + s/(Q_z 2\pi F_z) + 1] / [s^2/(2\pi F_p)^2 + s/(Q_p 2\pi F_p) + 1]$$

The following block diagram describes the AR filter function; note that AR1 and AR2 are in the forward path, while AR3 and AR4 are applied to feedback:



AR1, AR2, AR3 and AR4 are used in velocity and position mode, but are disabled in torque mode.

Discrete time transfer function (applies to all AR filters)

The velocity loop compensation is actually implemented as a digital discrete time system function on the DSP. The continuous time transfer function is converted to the discrete time domain by a backward Euler mapping:

$$s \approx (1-z^{-1})/t, \text{ where } t = 62.5 \mu\text{s}.$$

The poles are prewarped to F_p and the zeros are prewarped to F_z .

Related Topics

11.4 Velocity Loop

VL.BUSFF

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Displays the velocity loop feedforward value injected by the field-bus; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEARUNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, μ m/s, (custom units)/s |
| Range | 0.0 to VL.LIMITP |
| Default Value | 0.0 |
| Data Type | Float |
| See Also | VL.FF, VL.KBUSFF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 60B1h/0 | M_01-00-00-000 |
| Modbus | 848 | M_01-03-00-000 |

Description

This parameter displays the velocity loop feedforward value injected by the fieldbus.

Related Topics

11.4 Velocity Loop

VL.CMD

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the actual velocity command; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEARUNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, $\mu\text{m/s}$, (custom units)/s |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | VL.FB, VL.CMDU, VL.LIMITP, VL.LIMITN |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 606Bh/0 | M_01-00-00-000 |
| Modbus | 850 | M_01-03-00-000 |

Description

VL.CMD returns the actual velocity command as it is received in the velocity loop entry after all velocity limits (such as VL.LIMITN and VL.LIMITP). See velocity loop design diagram for more details.

Related Topics

- 10.4 Analog Input
- 11.4 Velocity Loop
- 11.5 Position Loop
- Block Diagrams

VL.CMDU

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the user velocity command; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEARUNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, μ m/s, custom units/s |
| Range | Rotary -15,000.000 to 15,000.000 rpm -250.000 to 250.000 rps -90000.000 to 90000.000 deg/s -1250.000 to 1250.000 custom units/s -1570.796 to 1570.796 rad/s Linear -1,073,741,824,000.000 to 1,073,741,824,000.000 counts/s -8,000.000 to 8,000.000 mm/s -8,000,000.000 to 8,000,000.000 μ m/s -1,250.000 to 1,250.000 custom units/s |
| Default Value | 0 |
| Data Type | Float |
| See Also | VL.FB, VL.CMD, DRV.OPMODE, DRV.CMDSOURCE, VL.LIMITN, VL.LIMITP |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|-------------------------|----------------------|
| EtherCAT COE and CANopen | Profile specific SDOs/0 | M_01-00-00-000 |
| Modbus | 852 | M_01-03-00-000 |

Description

VL.CMDU sets the user velocity command.

When DRV.OPMODE is set to 1 (velocity loop) and DRV.CMDSOURCE is set to 0 (TCP/IP channel), then setting this value when the drive is enabled will cause the drive to rotate at the required velocity.

Related Topics

11.4 Velocity Loop

VL.ERR

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Sets the velocity error; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, $\mu\text{m/s}$, (custom units)/s |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | VL.CMD, VL.FB |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3407h/4 | M_01-00-00-000 |
| Modbus | 854 | M_01-03-00-000 |

Description

VL.ERR sets the velocity error. It is calculated in the velocity loop as the difference between VL.CMD and VL.FB.

Related Topics

11.4 Velocity Loop

VL.FB

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the velocity feedback; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEARUNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, μ m/s, (custom units)/s |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | VL.CMDU |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|-----------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3618h/0 | M_01-00-00-000 |
| | 606Ch/0 | |
| Modbus | 856 | M_01-03-00-000 |

Description

VL.FB returns the velocity feedback as it is received in the velocity loop, after passing through Filter 3 and Filter 4.

Related Topics

11.4 Velocity Loop

VL.FBFILTER

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Filters VL.FB value; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, $\mu\text{m/s}$, (custom units)/s |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | VL.FB |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3407h/1 | M_01-00-00-000 |
| Modbus | 858 | M_01-03-00-000 |

Description

This parameter returns the same value as VL.FB, filtered through a 10 Hz filter.

Related Topics

11.4 Velocity Loop

VL.FBSOURCE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets feedback source for the velocity loop; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Boolean |
| See Also | PL.FBSOURCE |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 860 | M_01-03-00-000 |

Description

This parameter determines the feedback source to be used by the velocity loop. A value of 0 selects the primary feedback, 1 selects the secondary feedback.

Related Topics

11.4 Velocity Loop

VL.FBUNFILTERED

| General Information | |
|----------------------|---|
| Type | R/O Parameter |
| Description | Reads the velocity feedback. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR, UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, $\mu\text{m/s}$, (custom units)/s |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | VL.FB, VL.FBFILTER |
| Start Version | M_01-03-06-000 |

Description

VL.FBUNFILTERED reads the raw velocity feedback before any filters affect the value of this feedback.

Related Topics

Velocity Loop

VL.FF

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Displays the velocity loop overall feedforward value; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.ACCROTARY or UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, μ m/s, (custom units)/s |
| Range | 0 to VL.LIMITP |
| Default Value | 0 |
| Data Type | Float |
| See Also | VL.KBUSFF, VL.KVFF, VL.ACFFGAIN |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 862 | M_01-03-00-000 |

Description

This parameter displays the velocity loop overall feedforward value.

Related Topics

11.4 Velocity Loop

VL.GENMODE

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Selects mode of velocity generation (Observer, d/dt); active in opmodes 1 (velocity) and 2 (position) only. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 864 | M_01-03-00-000 |

Description

This parameter is used to select the velocity generator mode.

| Mode | Description |
|------|--|
| 0 | d/dt mode: The derivative of the mechanical angle of the drive is fed to a first order low pass. |
| 1 | Luenberger Observer mode |

Related Topics

11.4 Velocity Loop

VL.KBUSFF

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the velocity loop acceleration feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | NA |
| Range | 0.0 to 2.0 |
| Default Value | 0.0 |
| Data Type | Float |
| See Also | VL.BUSFF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3416h | M_01-00-00-000 |
| Modbus | 866 | M_01-03-00-000 |

Description

This parameter sets the gain for the acceleration feedforward (a scaled second derivative of the position command is added to the velocity command value).

The nominal feedforward value can be multiplied by this gain value.

This will have affect only when using position mode (DRV.OPMODE = 2).

Related Topics

11.4 Velocity Loop

VL.KI

| General Information | |
|---------------------|---|
| Type | NV Parameter |
| Description | Sets the velocity loop integral gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Hz |
| Range | 0 to 1,000 Hz |
| Default Value | 160 Hz |
| Data Type | Float |
| See Also | VL.KP |
| Start Version | M_01-00-00-000 |

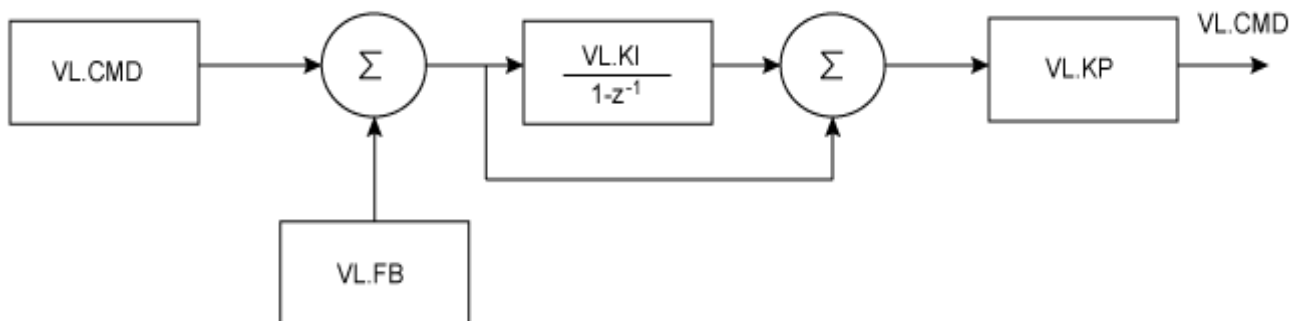
| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 354Dh/0 | M_01-00-00-000 |
| Modbus | 868 | M_01-03-00-000 |

Description

VL.KI sets the integral gain of the velocity loop.

A factor of 2π is included in the time calculation, therefore a PI velocity loop with a constant error of 1 rps in which VL.KI is set to 160 and VL.KP is set to 1, will take $(1000/160)*2\pi$ ms to increase the integral gain to 1. Therefore, the total gain is 2 at this time (see velocity loop structure below).

Velocity Loop Structure



Related Topics

11.4 Velocity Loop

VL.KO

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Scales the observer velocity signal; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Internal |
| Range | 0 to 65,535 |
| Default Value | 0 |
| Data Type | Float |
| See Also | VL.MODEL |
| Start Version | M_01-00-01-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 870 | M_01-03-00-000 |

Description

VL.KO is used to scale the observer model to match the load. When VL.KO is tuned properly, Reads the observer velocity signal; active in opmodes 1 (velocity) and 2 (position) only. will match VL.FB, except when there is an unspecified offset between the two.

Related Topics

11.4 Velocity Loop

VL.KP

| General Information | |
|---------------------|---|
| Type | NV Parameter |
| Description | Sets velocity loop proportional gain for the PI controller; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | A/(rad/sec) |
| Range | 0.001 to 2,147,483.008 |
| Default Value | 1 |
| Data Type | Float |
| See Also | VL.KI |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3548h/0 | M_01-00-00-000 |
| Modbus | 872 | M_01-03-00-000 |

Description

VL.KP sets the proportional gain of the velocity loop.

The idealized velocity loop bandwidth in Hz is:

Rotary motor:

$$\text{Bandwidth (Hz)} = \text{VL.KP} * K_t / (2\pi * J_m)$$

Where:

K_t = motor torque constant, in units of Nm/Arms

J_m = motor inertia, in units of kg*m²

Linear motor:

$$\text{Bandwidth (Hz)} = \text{VL.KP} * K_t / (\text{Motor Pitch (mm)} * J_m)$$

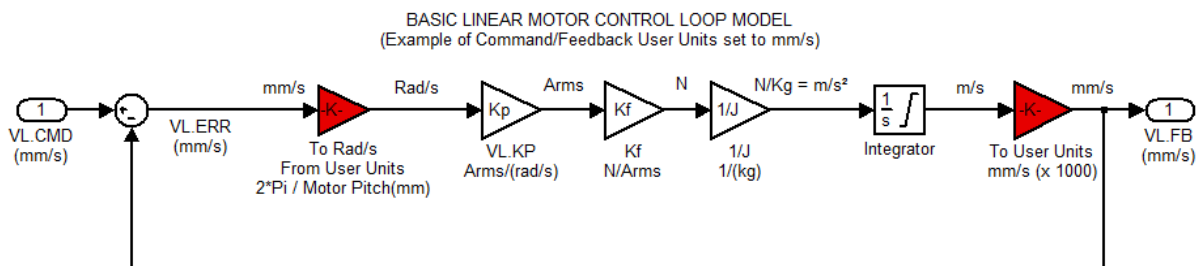
Where:

K_t = motor torque constant, in units of Nm/Arms

J_m = motor inertia, in units of kg

The drive uses the same control loop for both linear and rotary motors. VL.KP units are in Arms/(rad/s). If you want to tune in units of Arms/(mm/s), then you must manually convert the units.

The diagram below shows how linear motors are implemented at the control loop level.



The red blocks are automatically handled at the drive level.

2π radians is the linear equivalent of one full mechanical revolution of a rotary motor - and is equal to the MOTOR.PITCH of a linear motor.

Example

To convert VL.KP = 0.320 Arms/(rad/s) to Arms/(mm/s), where MOTOR.PITCH is 32 mm:

$$VL.KP = 0.320 \text{ Arm /rad/s} * (2\pi \text{ rad} / 32\text{mm MOTOR.PITCH})$$

$$VL.KP = 0.32 * 2\pi / 32 = 0.063 \text{ Arms / (mm/s)}$$

Related Topics

Velocity Controller Environment Block Diagram

VL.KVFF

| General Information | |
|---------------------|---|
| Type | R/W Parameter |
| Description | Sets the velocity loop velocity feedforward gain value; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | NA |
| Range | 0.0 to 2.0 |
| Default Value | 0.0 |
| Data Type | Float |
| See Also | VL.FF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3407h | M_01-00-00-000 |
| | 354Bh/0 | |
| Modbus | 874 | M_01-03-00-000 |

Description

This parameter sets the gain for the velocity feedforward (a scaled derivative of the position command is added to the velocity command value). The nominal feedforward value can be multiplied by this gain value.

This parameter is only used in the position mode (DRV.OPMODE = 2).

Related Topics

11.4 Velocity Loop

VL.LIMITN

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the velocity lower limit; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, μ m/s, custom units/s |
| Range | Rotary: -15,000.000 to 0.000 rpm -250.000 to 0.000 rps -90,000.000 to 0.000 deg/s -1,250.000 to 0.000 custom units/s -1570.796 to 0.000 rad/s Linear: -1,073,741,824,000.000 to 0.000 counts/s -250.000*MOTOR.PITCH to 0.000 mm/s -250,000.000*MOTOR.PITCH to 0.000 μ m/sec -1,250.000 to 0.000 custom units/s |
| Default Value | Rotary: -3,000.000 rpm -50.000 rps -18,000.002 deg/s -250.000 (custom units)/s -314.159 rad/s Linear: -0.050 counts/s -50*MOTOR.PITCH mm/s -50,000.004*MOTOR.PITCH μ m/sec -250.000 custom units/s |
| Data Type | Float |
| See Also | VL.LIMITP, VL.CMD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3623h/0 | M_01-00-00-000 |
| Modbus | 876 | M_01-03-00-000 |

Description

VL.LIMITN sets the velocity command negative limit.

If the input to the velocity loop is lower than VL.LIMITN, then the actual velocity command VL.CMD is limited by the value of VL.LIMITN.

Related Topics

11.4 Velocity Loop

VL.LIMITP

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the velocity high limit; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, μ m/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 250.000*MOTOR.PITCH mm/sec 0.000 to 250,000.000*MOTOR.PITCH μ m/s 0.000 to 1,250.000 custom units/s |
| Default Value | Rotary: 3,000.000 rpm 50.000 rps 18,000.002 deg/s 250.000 (custom units)/s 314.159 rad/s Linear: 0.050 counts/s 50.000*MOTOR.PITCH mm/sec 50,000.004*MOTOR.PITCH μ m/sec 250.000 custom units/s |
| Data Type | Float |
| See Also | VL.LIMITN, VL.CMD |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3622h/0 | M_01-00-00-000 |
| Modbus | 878 | M_01-03-00-000 |

Description

VL.LIMITP sets the velocity command positive limit.

If the input to the velocity loop is higher than VL.LIMITP, then the actual velocity command VL.CMD is limited by the value of VL.LIMITP.

Related Topics

11.4 Velocity Loop

VL.LMJR

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets the ratio of the estimated load moment of inertia relative to the motor moment of inertia; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | NA |
| Range | 0 to 100.0 |
| Default Value | 0 |
| Data Type | Float |
| See Also | IL.KACCFF, IL.FF |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 880 | M_01-03-00-000 |

Description

This parameter is used in the internal calculation of the current loop acceleration feed forward gain value.

Related Topics

11.4 Velocity Loop

VL.MODEL

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads the observer velocity signal; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR, UNIT.ACCLINEAR Rotary: rpm, rps, deg/s, (custom units)/s, rad/s Linear: counts/s, mm/s, μ m/s, (custom units)/s |
| Range | N/A |
| Default Value | N/A |
| Data Type | Float |
| See Also | VL.FB, VL.KO |
| Start Version | M_01-00-01-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 882 | M_01-03-00-000 |

Description

VL.MODEL is the observer velocity output. When VL.KO is tuned properly, VL.MODEL will match VL.FB, except when there is an unspecified offset between the two.

Related Topics

11.4 Velocity Loop

VL.OBSBW

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the bandwidth of the observer in Hz. |
| Units | Hz |
| Range | 10 to 4,000 Hz |
| Default Value | 30 Hz |
| Data Type | Float |
| See Also | N/A |
| Start Version | M_01-03-00-004 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 884 | M_01-03-00-000 |

Description

This parameter sets the bandwidth of the observer in Hz. The observer passes the velocity feedback through a PID control loop that behaves like a low-pass filter with a bandwidth of VL.OBSBW.

Related Topics

11.4 Velocity Loop

VL.OBSMODE

| General Information | |
|----------------------|-----------------------------------|
| Type | NV Parameter |
| Description | Sets the observer operating mode. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | Integer |
| See Also | N/A |
| Start Version | M_01-03-00-004 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 886 | M_01-03-00-000 |

Description

This parameter sets the observer operating mode. When VL.OBSMODE = 0, the observer is not part of the control loop; that is, VL.FB is used as the velocity feedback signal to the velocity loop. When VL.OBSMODE = 1, the observer is part of the control loop; VL.MODEL is used as the velocity feedback signal.

Related Topics

11.4 Velocity Loop

VL.THRESH

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the over speed fault value; active in opmodes 1 (velocity) and 2 (position) only. |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, $\mu\text{m/s}$, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 deg/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 250.000*MOTOR.PITCH mm/s 0.000 to 250,000.000*MOTOR.PITCHMOTOR.PITCH $\mu\text{m/s}$ 0.000 to 1,250.000 custom units/s |
| Default Value | Rotary: 3,600 rpm 60 rps 21,600.000 deg/s 300.000 custom units/s 376.991 rad/s Linear: 0.060 counts/s 60.000*MOTOR.PITCH mm/s 60,000.04*MOTOR.PITCHMOTOR.PITCH $\mu\text{m/s}$ 300.000 custom units/s |
| Data Type | Float |
| See Also | VL.CMD, VL.CMDU |
| Start Version | M_01-00-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3627h/0 | M_01-00-00-000 |
| Modbus | 888 | M_01-03-00-000 |

Description

VL.THRESH sets the threshold for the velocity over which an over speed fault is generated. The value is considered as an absolute value, hence it applies for both negative and positive velocities.

Example

VL.THRESH is set to 600 rpm. A velocity (VL.FB) of 700 rpm will generate an over speed fault.

Related Topics

11.4 Velocity Loop

WS Parameters

WS.ARM

| General Information | |
|---------------------|--|
| Type | Command |
| Description | Sets wake and shake to start at the next drive enable. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3494h/6 | M_01-00-00-000 |
| Modbus | 890 | M_01-03-00-000 |

Description

This command sets wake and shake to start at the next drive enable. Feedback type is not relevant for this command. If WS.STATE is 0 and the drive is disabled, then WS.STATE will change to 1 after issuing WS.ARM. With this command, wake and shake can be repeated if desired.

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.DISARM

| General Information | |
|----------------------|---|
| Type | Command |
| Description | Cancels ARM requests and resets wake and shake to the IDLE state. |
| Units | N/A |
| Range | N/A |
| Default Value | N/A |
| Data Type | N/A |
| See Also | N/A |
| Start Version | M_01-04-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | N/A | |
| Modbus | N/A | |

Description

This command disables wake and shake immediately. Feedback type is not relevant for this command. If WS.ARM has been issued, the request to execute the wake and shake algorithm at the next enable is cancelled. WS.STATE is set to IDLE.

Related Topics

8.3.9 Using Wake and Shake (WS))

WS.DISTMAX

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets maximum movement allowed for wake and shake. |
| Units | deg (position units) |
| Range | 0 to 90 deg |
| Default Value | 15 deg |
| Data Type | S64 |
| See Also | N/A |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3494h/2 | M_01-00-00-000 |
| Modbus | 892 (64-bit) | M_01-03-00-000 |

Description

This parameter sets the maximum movement that is allowed for finding commutation. If this value is too small, FF475, "Wake and Shake. Too much movement", may occur before wake and shake is finished. The bigger this value, the more movement is allowed for wake and shake. This value is application dependent.

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.DISTMIN

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the minimum movement required for wake and shake. |
| Units | Actual position units |
| Range | 0 to 90 deg |
| Default Value | 1 deg |
| Data Type | S64 |
| See Also | N/A |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 36D1h/0 | M_01-00-00-000 |
| Modbus | 896 (64-bit) | M_01-03-00-000 |

Description

This parameter sets the minimum movement that is required for commutation finding. If this value is too small, the the commutation finding might fail if too little current is used. The larger this value, the more movement is needed in order to avoid F473: "Wake and Shake: Too little movement".

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.IMAX

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets maximum current used for wake and shake. |
| Units | Arms |
| Range | 0 to (lower value of MOTOR.IPEAK and DRV.IPEAK) Arms |
| Default Value | (half of maximum) Arms |
| Data Type | U16 |
| See Also | MOTOR.IPEAK, DRV.IPEAK |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3494h/1 | M_01-00-00-000 |
| Modbus | 900 | M_01-03-00-000 |

Description

This parameter defines the maximum current used for wake and shake. If the selected current is too low, the minimum required movement may not occur. If the selected current is too high, the movement may be too fast (over-speed) or too large (over maximum movement).

The maximum of this parameter is the lower value of MOTOR.IPEAK and DRV.IPEAK. The default value of this parameter is the half of its maximum. This value depends on the specific application.

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.MODE

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the method used for wake and shake. |
| Units | N/A |
| Range | 0 to 1 |
| Default Value | 0 |
| Data Type | U8 |
| See Also | N/A |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|----------|----------------|----------------------|
| Modbus | 902 | M_01-03-00-000 |

Description

This parameter sets the method used for finding commutation.

0 = Standard wake and shake

Two iterations are used to find the correct angle in this mode. Coarse (current mode) and fine (velocity mode) iterations are done in a loop (WS.NUMLOOPS times). The average angle of all loops is calculated and used.

1 = Commutation alignment by fixed commutation vector (Zero Method)

The motor poles are set to 0, current mode is activated, and WS.IMAX is applied. The angle in which the motor settles is used for commutation. Other settings are restored (such as motor poles and operation mode).

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.NUMLOOPS

| General Information | |
|----------------------|--|
| Type | R/W Parameter |
| Description | Sets the number of repetitions for wake and shake. |
| Units | counts |
| Range | 0 to 20 counts |
| Default Value | 5 counts |
| Data Type | U8 |
| See Also | N/A |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 36E2h/0 | M_01-00-00-000 |
| Modbus | 904 | M_01-03-00-000 |

Description

This parameter sets the maximum number of wake and shake repetitions. MOTOR.PHASE is calculated as mean value of all wake and shake repetitions.

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.STATE

| General Information | |
|----------------------|--|
| Type | R/O Parameter |
| Description | Reads wake and shake status |
| Units | N/A |
| Range | N/A |
| Default Value | Only valid before the first enable occurs. 11 - for feedback types that do not require wake and shake 1 - for feedback types that require wake and shake |
| Data Type | U8 |
| See Also | N/A |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3494h/5 | M_01-00-00-000 |
| Modbus | 906 | M_01-03-00-000 |

Description

WS switches different current vectors and records position feedback in order to establish commutation alignment.

WS.STATE 0 = wake and shake successful (DONE).

WS.STATE 1 = wake and shake configured and will be done at next enable (ARMED).

WS.STATE 2 = wake and shake running. (ACTIVE)

WS.STATE 10 = error occurred during wake and shake (ERROR).

WS.STATE 11 = wake and shake not required (IDLE).

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.T

| General Information | |
|----------------------|---|
| Type | R/W Parameter |
| Description | Sets wake and shake current-vector appliance time |
| Units | ms |
| Range | 1 to 200 ms |
| Default Value | 2 ms |
| Data Type | U8 |
| See Also | WS.IMAX, WS.DISTMAX |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 36D0h/0 | M_01-00-00-000 |
| Modbus | 908 | M_01-03-00-000 |

Description

This parameter defines the duration for each different current-vector while the coarse angle calculation. The move distance is proportional to the WS.T and WS.IMAX value.

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.TDELAY1

| General Information | |
|----------------------|---------------------------------|
| Type | NV Parameter |
| Description | Delay for wake and shake timing |
| Units | ms |
| Range | 0 to 200 ms |
| Default Value | 5 ms |
| Data Type | U8 |
| See Also | N/A |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3683h/0 | M_01-00-00-000 |
| Modbus | 910 | M_01-03-00-000 |

Description

WS.TDELAY1 defines the delay time of the wake and shake function. This time is a delay time between the switching of different current vectors during the wake and shake procedure. This time should be increased in the case of movement interferences between single current vectors.

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.TDELAY2

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Sets the delay for wake and shake timing. |
| Units | ms |
| Range | 0 to 200 ms |
| Default Value | 50 ms |
| Data Type | U8 |
| See Also | N/A |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3685h/0 | M_01-00-00-000 |
| Modbus | 912 | M_01-03-00-000 |

Description

WS.TDELAY2 defines the delay between switching from coarse angle calculation to fine angle calculation during the wake and shake procedure. This time should be increased in the case of interferences between the coarse calculation done in current mode and the fine calculation done in velocity mode. Choosing too large a value increases the wake and shake duration.

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.TDELAY3

| General Information | |
|----------------------|--|
| Type | NV Parameter |
| Description | Sets the delay for wake and shake between loops in mode 0. |
| Units | ms |
| Range | 0 to 2,000 ms |
| Default Value | 100 ms |
| Data Type | U16 |
| See Also | N/A |
| Start Version | M_01-01-00-102, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|---------------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3494h/3 | M_01-00-00-000 |
| Modbus | 914 | M_01-03-00-000 |

Description

WS.TDELAY3 defines the delay between complete loops in mode 0 only. Decreasing this value makes the wake and shake procedure faster, but may lead to problems if the motor moves too long. Increasing this value will make wake and shake significantly longer.

Related Topics

8.3.9 Using Wake and Shake (WS)

WS.VTHRESH

| General Information | |
|----------------------|---|
| Type | NV Parameter |
| Description | Defines the maximum allowed velocity for Wake & Shake |
| Units | Depends on UNIT.VROTARY or UNIT.VLINEARUNIT.ACCLINEAR Rotary: rpm, rps, deg/s, custom units/s, rad/s Linear: counts/s, mm/s, μ m/s, custom units/s |
| Range | Rotary: 0.000 to 15,000.000 rpm 0.000 to 250.000 rps 0.000 to 90,000.000 degree/s 0.000 to 1,250.000 custom units/s 0.000 to 1,570.796 rad/s Linear: 0.000 to 1,073,741,824,000.000 counts/s 0.000 to 8,000.000 mm/s 0.000 to 8,000,000.000 μ m/s 0.000 to 1,250.000 custom units/s |
| Default Value | 100 rpm |
| Data Type | U16 |
| See Also | N/A |
| Start Version | M_01-01-00-101, M_01-02-00-000 |

| Fieldbus | Index/Subindex | Object Start Version |
|--------------------------|----------------|----------------------|
| EtherCAT COE and CANopen | 3494h/4 | M_01-00-00-000 |
| Modbus | 916 | M_01-03-00-000 |

Description

This parameter defines the maximum allowed velocity that occurs while commutation finding is active. This supervision runs in real time, but only while wake and shake is active (WS.STATE 2 or greater, for Mode 0). If at any time while wake and shake is running a velocity higher than this value is detected, fault F478 is generated. Setting WS.VTHRESH to zero disables this feature. For Mode 1, WS.VTHRESH is only used after the initial phase-finding.

Related Topics

8.3.9 Using Wake and Shake (WS)

This page intentionally left blank.

Sales and Service

We are committed to quality customer service. In order to serve in the most effective way, please contact your local sales representative for assistance.

If you are unaware of your local sales representative, please contact us.

Europe

Kollmorgen Customer Support Europe

Internet www.kollmorgen.com
E-Mail technik@kollmorgen.com
Tel.: +49(0)2102 - 9394 - 0
Fax: +49(0)2102 - 9394 - 3155

North America

Kollmorgen Customer Support North America

Internet www.kollmorgen.com
E-Mail support@kollmorgen.com
Tel.: +1 - 540 - 633 - 3545
Fax: +1 - 540 - 639 - 4162



25 Index

1

| | |
|------------|-----|
| 1000h | 163 |
| 1001h | 164 |
| 1002h | 165 |
| 1003h | 166 |
| 1005h | 167 |
| 1006h | 167 |
| 1008h | 168 |
| 1009h | 168 |
| 100Ah | 168 |
| 100Ch | 169 |
| 100Dh | 169 |
| 1010h | 170 |
| 1014h | 171 |
| 1016h | 171 |
| 1017h | 172 |
| 1018h | 172 |
| 1026h | 174 |
| 1400-1403h | 185 |
| 1600-1603h | 186 |
| 1800-1803h | 188 |
| 1A00-1A03h | 190 |

2

| | |
|------------|-----|
| 2014-2017h | 175 |
| 2018h | 176 |
| 2026h | 177 |
| 204Ch | 200 |
| 20A0h | 178 |
| 20A1h | 178 |
| 20A2h | 178 |
| 20A3h | 179 |
| 20A4h | 179 |
| 20A5h | 180 |
| 20A6h | 180 |
| 20B8h | 181 |

6

| | |
|-------|-----|
| 6040h | 195 |
| 6041h | 196 |
| 6060h | 198 |
| 6061h | 199 |
| 6063h | 207 |
| 6064h | 208 |
| 6065h | 208 |
| 606Ch | 203 |
| 6071h | 205 |
| 6073h | 205 |
| 6077h | 206 |
| 607Ah | 219 |

| | |
|-------|-----|
| 607Ch | 214 |
| 607Dh | 219 |
| 6081h | 220 |
| 6083h | 220 |
| 6084h | 220 |
| 608Fh | 201 |
| 6092h | 202 |
| 6098h | 214 |
| 6099h | 216 |
| 609Ah | 216 |
| 60C0h | 209 |
| 60C1h | 210 |
| 60C2h | 211 |
| 60C4h | 212 |
| 60F4h | 208 |
| 60FDh | 182 |
| 60FEh | 182 |
| 60FFh | 204 |
| 6502h | 183 |

A

| | |
|----------------------------|---------|
| Abbreviations | 60, 137 |
| Acceleration ramp | 359 |
| AL Event | 80 |
| Analog inputs | 275 |
| Analog output | 103 |
| Analog setpoints | 275 |
| Anti-resonance filter | 662 |
| Aux. supply 24V, interface | 267 |

B

| | |
|-----------------------------|-----|
| Basic data types | 147 |
| Basic drive setup | 34 |
| Basic testing | 236 |
| Blink | 362 |
| Block diagrams | |
| current loop | 282 |
| position and velocity loops | 282 |
| Brake view | 80 |

C

| | |
|-----------------------------------|--------|
| CANopen over EtherCAT | 84 |
| Capture | 81 |
| Check communications | 40, 43 |
| Command buffer | 96 |
| Command source | 366 |
| Communication Objects | 149 |
| Connect to another drive | 40 |
| Connected and disconnected states | 38 |
| Connection diagram | |
| AKD-x00306 to x00606 | 264 |
| AKD-x01206 | 265 |
| AKD-x01206 & AKD-xzzz07 | 266 |
| Connection Diagrams | 263 |

| | | | |
|--|-----------------|---|-----------------|
| Control word | 195 | Fieldbus Parameters | 96 |
| Control Word Commands | 85 | Foldback | 77 |
| Controlled Stop | 115 | G | |
| current loop | 121 | General definitions | 163 |
| Current loop gain scheduling | 122 | General objects | 163 |
| Cycle Time | | H | |
| Adjust | 89 | Heartbeat | 158 |
| Max.Values | 89 | Homing | 134 |
| Cyclical Values | 88 | Homing Mode | 214 |
| D | | I | |
| Data types | 147 | I/O-Connection | 104, 273 |
| DC bus link, interface | 270 | Inputs | |
| deceleration | 369 | Analog | 275 |
| Device control | 192 | Digital | 99 |
| Device Not Shown | 38, 46 | Enable | 102 |
| Dig. Encoder emulation, interface | 277 | Programmable | 102 |
| Digital inputs | 99 | Interpolated position mode | 209 |
| Digital Inputs and Outputs | 88 | Interrupt Event | 80 |
| Direct DC mains | 56 | Invalid Firmware | 259 |
| Disable | 373 | J | |
| Disconnected state | 38 | Jog Move | 153 |
| Display Codes | 34 | K | |
| Downloading Firmware | 259 | KAS IDE | 77 |
| Drive Motion Status | 153 | L | |
| Dynamic Braking | 117 | Latch Words | 91 |
| E | | M | |
| Electronic gearing | 106 | Macros | 219 |
| Emergency Messages | 253, 160 | Mailbox | 92 |
| Emergency Object | 151 | Mains supply, interface | 271 |
| emergency timeout | 375 | Manufacturer specific objects | 175 |
| Emulated Encoder Output | 277 | master/slave | 74 |
| ENABLE | 102 | Mixed data types | 148 |
| Enable/Disable | 111 | Modbus | 6 |
| encoder emulation | 74 | Modbus Dynamic Mapping | 9 |
| Error Messages | | Modulo position | 127 |
| parameter and command | 250 | Motion Tasks | 144 |
| EtherCAT onboard | 65 | Motor | 64 |
| EtherCAT Profile | 78 | Motor interface | 268 |
| Examples | 236 | N | |
| Examples, setup | 236 | Network Management Object | 150 |
| Examples, special applications | 249 | Nodeguard | 157 |
| Extended data types | 149 | Non-Plug and Play Feedback Devices | 76 |
| F | | | |
| Factor Groups | 199 | | |
| Fault messages | 240 | | |
| Faults | 364 | | |
| clearing | 249 | | |
| Feedback 1 | 66 | | |
| Feedback 2 | 73 | | |
| Feedback Connection | 279 | | |
| Fieldbus | 96 | | |

O

| | |
|-----------------------------------|------------|
| Object Dictionary | 223 |
| Operating mode | 198 |
| parameters and commands active in | 225, |
| | 265 |
| Operation Modes | 88 |
| Outputs | |
| Analog | 274 |
| Overview | 50 |

P

| | |
|-----------------------------------|----------------|
| Parameter Comparer | 221 |
| Parameter Load/Save | 221 |
| PDO configuration | 184 |
| PDO Mapping | 87 |
| Performance Servo Tuner | 160 |
| advanced | 164 |
| Phase run-up | 82 |
| Position Control Function | 207 |
| Position Loop | 126 |
| Power | 56 |
| Process Data Object | 155 |
| Profile Position Mode | 218 |
| Profile specific objects | 182 |
| Profile torque mode | 205 |
| Profile Velocity Mode | 203 |
| Programmable Limit Switch | 109 |
| Prohibited Use | 62, 139 |
| Pulse Direction, interface | 278 |

R

| | |
|----------------------------------|------------|
| Receive PDOs | 185 |
| Regen resistor, interface | 269 |
| Regeneration | 58 |
| Registration moves | 149 |
| Response monitoring | 169 |

S

| | |
|----------------------------------|------------|
| Safe Torque Off (STO) | 118 |
| Safety Instructions | |
| Electrical Installation | 64, 141 |
| General | 62, 139 |
| Save On Disconnect | 157 |
| Save On Exit | 156 |
| Save On Firmware Download | 158 |
| Save Options | 156 |
| scope | 205 |
| channels | 205 |
| settings and presets | 212 |
| time-base | 205 |
| SDO abort codes | 154 |
| Service Data Object | 152 |
| Service Motion | 152 |

| | |
|-------------------------------|------------|
| Settings | 52 |
| Setup | 66 |
| Signed Integer | 148 |
| Slave Register | 79 |
| Slider Tuning | 160 |
| Status machine | 192 |
| Status Machine | 84 |
| Status word | 196 |
| Status Word | 86 |
| Synchronization | 90 |
| Synchronization Object | 150 |

T

| | |
|---------------------------|----------------|
| Target group | 59, 136 |
| TCP/IP | 48 |
| Terminal | 218 |
| Time Stamp Object | 150 |
| Transmission modes | 156 |
| Transmit PDOs | 188 |
| Trigger modes | 156 |
| Trouble Shooting | 257 |
| Tuning | 189 |
| TwinCAT | 66 |

U

| | |
|-------------------------|----------------|
| Units | 84 |
| Unsigned Integer | 147 |
| Up/Down Input | 278 |
| Use as directed | 62, 139 |

V

| | |
|----------------------|------------|
| Velocity Loop | 124 |
|----------------------|------------|

W

| | |
|-------------------------------|------------|
| Wake and shake (WS) | 68 |
| Warnings | 240 |
| Watch | 52 |
| Wizard | 35 |
| Workbench over TwinCAT | 71 |

This page intentionally left blank.

Sales and Service

We are committed to quality customer service. In order to serve in the most effective way, please contact your local sales representative for assistance.

If you are unaware of your local sales representative, please contact us.

Europe

Kollmorgen Customer Support Europe

Internet www.kollmorgen.com
E-Mail technik@kollmorgen.com
Tel.: +49(0)2102 - 9394 - 0
Fax: +49(0)2102 - 9394 - 3155

North America

Kollmorgen Customer Support North America

Internet www.kollmorgen.com
E-Mail support@kollmorgen.com
Tel.: +1 - 540 - 633 - 3545
Fax: +1 - 540 - 639 - 4162

