CD SynqNet

Installation and User Manual

Revision No: 2.7
Date: 07 September 2004
# Table Of Contents

0. **REVISION HISTORY** ........................................................................................................... 7

1. **CONVENTIONS** ........................................................................................................... 9

2. **PRODUCT OVERVIEW** .................................................................................................. 9
   2.1 **DESCRIPTION** ........................................................................................................ 9
   2.2 **PRODUCT OPTIONS** .............................................................................................. 9
   2.3 **PART NUMBER** ...................................................................................................... 10

3. **PRODUCT BLOCK DIAGRAM** ..................................................................................... 11

4. **FUNCTIONAL CAPABILITIES** ..................................................................................... 12
   4.1 **SERVO CONTROL** .................................................................................................. 12
   4.2 **MOTOR TYPE** ....................................................................................................... 12
   4.3 **MOTOR FEEDBACK** ............................................................................................... 12
      4.3.1 **Incremental Encoder** ....................................................................................... 12
      4.3.2 **Sine Encoder** .................................................................................................. 13
      4.3.3 **Resolver** ......................................................................................................... 14
      4.3.4 **Connector** ....................................................................................................... 15
   4.4 **I/O 16**
      4.4.1 **I/O Summary** .................................................................................................. 16
      4.4.2 **I/O Electrical Characteristics** ........................................................................... 18
      4.4.3 **I/O Connectors** .............................................................................................. 22
   4.5 **COMMUNICATIONS** ............................................................................................... 24
      4.5.1 **SynqNet** ........................................................................................................ 24
      4.5.2 **Serial RS-232** ................................................................................................ 25
   4.6 **SWITCHES** ............................................................................................................. 25
   4.7 **PROTECTION** .......................................................................................................... 25
   4.8 **SECONDARY ENCODER** ......................................................................................... 26
      4.8.1 **Connector** ....................................................................................................... 26
      4.8.2 **Pin-Out** ........................................................................................................... 26

5. **ELECTRICAL SPECIFICATIONS** .................................................................................. 27
   5.1 **300V MODEL** ........................................................................................................ 27
      5.1.1 **Electrical Specification** .................................................................................... 27
      5.1.2 **Regen Information** ........................................................................................ 28
   5.2 **600V OPTION** ....................................................................................................... 28
      5.2.1 **Electrical Specification** .................................................................................... 28
      5.2.2 **Regen Information** ........................................................................................ 29

6. **MOUNTING** .................................................................................................................. 30
   6.1 **300V MODEL** ........................................................................................................ 30
      6.1.1 **Hardware Specifications** .................................................................................. 30
      6.1.2 **Outline Dimensions** ....................................................................................... 31
   6.2 **600V MODEL** ........................................................................................................ 31
      6.2.1 **Hardware Specifications** .................................................................................. 31
      6.2.2 **Outline Dimensions** ....................................................................................... 32

7. **WIRING DIAGRAM** ...................................................................................................... 36
   7.1 **300V MODEL: 3A, 6A, 10A** ................................................................................... 36
   7.2 **600V MODEL AND 20A 300V MODEL** .................................................................. 37
7.3 **SYNQNET CABLES** ........................................................................................................... 38

8. **OPERATIONAL USE** ........................................................................................................ 38

8.1 **POWERING UP** ............................................................................................................. 38
8.2 **CURRENT SCALING** ....................................................................................................... 38
8.3 **DRIVE CONFIGURATION** ............................................................................................... 38
  8.3.1 **Current Loop Gain** .................................................................................................... 38
  8.3.2 **Back-EMF Compensation** .......................................................................................... 39
  8.3.3 **Phase Angle Advance** ................................................................................................. 40
8.4 **MEMORY DESCRIPTIONS** .............................................................................................. 40
8.5 **INDICATORS** .................................................................................................................. 41
  8.5.1 **Drive Status** ................................................................................................................ 42
  8.5.2 **SynqNet™ Status** ....................................................................................................... 43
8.6 **FIRMWARE UPDATE** ........................................................................................................ 43
  8.6.1 **Terminology** ................................................................................................................ 44
  8.6.2 **Important Files** ........................................................................................................... 44
  8.6.3 **Preparations** ................................................................................................................ 44
  8.6.4 **Update Drive Firmware** .............................................................................................. 47
  8.6.5 **Resuming Operation** .................................................................................................. 50
8.7 **ABSOLUTE POSITION** .................................................................................................. 50
8.8 **COMMUTATION INITIALIZATION WITHOUT HALLS** .................................................. 51
  8.8.1 **Overview** .................................................................................................................... 51
  8.8.2 **The Process** ................................................................................................................ 51
8.9 **COMMUTATION INITIALIZATION WITH HALLS** .......................................................... 52
  8.9.1 **Reset Parameters** ....................................................................................................... 53
  8.9.2 **Identify the direction of motor phases** ....................................................................... 53
  8.9.3 **Set MFBDIR bit 2** ........................................................................................................ 53
  8.9.4 **Align the halls to motor phase (set MPHASE)** .............................................................. 53
  8.9.5 **Set MFBDIR bit 0** ......................................................................................................... 54
  8.9.6 **Set MFBDIR bit 1** ......................................................................................................... 54
  8.9.7 **Operate the system** ...................................................................................................... 54
8.10 **CONSIDERATIONS FOR WORKING WITH ENDat SINE ENCODERS** ......................... 55
  8.10.1 **Setting the Encoder Type** ......................................................................................... 55
  8.10.2 **Equivalent Counts per Revolution** ............................................................................. 55
  8.10.3 **Hardware Absolute Position** .................................................................................... 55
  8.10.4 **Absolute Position Mode** ........................................................................................... 55
  8.10.5 **Position Feedback Offset** .......................................................................................... 55
  8.10.6 **Sine/Cosine Calibration** ............................................................................................. 56
8.11 **SINE/COSINE CALIBRATION** .................................................................................... 56
  8.11.1 **Overview** .................................................................................................................. 56
  8.11.2 **The Process** ................................................................................................................ 56
8.12 **DRIVE ENABLE** ............................................................................................................ 57
8.13 **OPERATING MODES** ..................................................................................................... 57
8.14 **DIVIDE-BY-N** .............................................................................................................. 57
  8.14.1 **Synchronizing the Position Feedback** ...................................................................... 57
  8.14.2 **DBN Setup** ................................................................................................................ 57
  8.14.3 **Effect of Direction** ..................................................................................................... 57
  8.14.4 **Calculating the DBN Position Error** ......................................................................... 58
8.15 **HOMING** ....................................................................................................................... 58
8.16 **FAULTS AND WARNINGS** ............................................................................................ 58
  8.16.1 **Warning** .................................................................................................................... 58
  8.16.2 **Fault Status Word** ...................................................................................................... 58
  8.16.3 **Reading Faults Over SynqNet** .................................................................................. 60
  8.16.4 **Reading Warnings Over SynqNet** ............................................................................. 60
8.16.5 Clearing Faults........................................................................................................... 61
8.16.6 Fault History............................................................................................................. 61
8.17 ANALOG INPUTS.......................................................................................................... 61
8.17.1 Drive Parameters.................................................................................................... 61
8.17.2 Direct Commands.................................................................................................. 61
8.18 DIRECT COMMANDS................................................................................................. 61
8.18.1 Direct Command Syntax....................................................................................... 61
8.18.2 Table of Direct Command Codes ...................................................................... 62
8.18.3 Accessing Analog Inputs Using Direct Commands............................................. 64
8.18.4 Examples................................................................................................................ 64
8.19 REAL TIME MONITORING ....................................................................................... 65
8.19.1 Analog Value Monitoring..................................................................................... 65
8.19.2 Setting up Analog Input Monitoring.................................................................... 66
8.19.3 Viewing Monitored Data on MotionScope......................................................... 66

9. INSTRUCTION SET ......................................................................................................... 69
9.1 TABLE OF CD SYNQNET PARAMETERS ................................................................... 69
9.1.1 NOP....................................................................................................................... 69
9.1.2 MBEMFCOMP....................................................................................................... 69
9.1.3 DICON T ................................................................................................................. 70
9.1.4 DIPEAK.................................................................................................................. 70
9.1.5 ICON T .................................................................................................................. 70
9.1.6 ILIM ....................................................................................................................... 70
9.1.7 IMAX...................................................................................................................... 71
9.1.8 MENCRES............................................................................................................. 71
9.1.9 MENCOFF............................................................................................................. 72
9.1.10 MICON T .............................................................................................................. 72
9.1.11 MIPEAK............................................................................................................... 72
9.1.12 MKT .................................................................................................................... 73
9.1.13 MLGAING.......................................................................................................... 73
9.1.14 MLGAINGP....................................................................................................... 74
9.1.15 MLMIN ............................................................................................................... 74
9.1.16 MPOLES ............................................................................................................. 74
9.1.17 MPOLES............................................................................................................. 75
9.1.18 MSPEED............................................................................................................. 75
9.1.19 MTANGLC......................................................................................................... 75
9.1.20 MTANGLP......................................................................................................... 76
9.1.21 MVANGLF......................................................................................................... 76
9.1.22 MVANGLH....................................................................................................... 77
9.1.23 PWMFRQ.......................................................................................................... 77
9.1.24 VBUS .................................................................................................................. 77
9.1.25 VER...................................................................................................................... 78
9.1.26 PFB...................................................................................................................... 78
9.1.27 DIVN.................................................................................................................... 78
9.1.28 DIVNDIR............................................................................................................ 78
9.1.29 DIVNSTART...................................................................................................... 78
9.1.30 DIVNSTOP........................................................................................................ 79
9.1.31 MHINVA............................................................................................................ 79
9.1.32 MHINVB............................................................................................................ 79
9.1.33 MHINVC............................................................................................................ 79
9.1.34 IN80................................................................................................................... 80
9.1.35 NOP..................................................................................................................... 80
9.1.36 MPITCH.............................................................................................................. 80
9.1.37 MENCTYPE ....................................................................................................... 81
9.1.38 REMOTE ................................................................. 81
9.1.39 DIVNspd ............................................................... 81
9.1.40 MOTORTYPE ......................................................... 81
9.1.41 VLIM ........................................................................ 82
9.1.42 MFBDIR ................................................................. 82
9.1.43 ANOFF1 ................................................................. 82
9.1.44 ANOFF2 ................................................................. 83
9.1.45 ANIN1 ................................................................. 83
9.1.46 ANIN2 ................................................................. 83
9.1.47 VMAX ................................................................. 84
9.1.48 WNSERR .............................................................. 84
9.1.49 INITGAIN .............................................................. 85
9.1.50 IENCSTART ......................................................... 85
9.1.51 MJ ................................................................. 85
9.1.52 UVMODE ............................................................. 85
9.1.53 UVTIME .............................................................. 86
9.1.54 UVRECOVER ......................................................... 86
9.1.55 THERM ............................................................... 86
9.1.56 THERMODE .......................................................... 86
9.1.57 THERMTYPE ....................................................... 87
9.1.58 IZERO ................................................................. 87
9.1.59 ZERO ................................................................. 87
9.1.60 SININITST .......................................................... 87
9.1.61 SINPARAM1 ......................................................... 88
9.1.62 SINPARAM2 ......................................................... 88
9.1.63 SINPARAM3 ......................................................... 88
9.1.64 SINPARAM4 ......................................................... 89
9.1.65 ABSPOSMOD ....................................................... 89
9.1.66 MSININT ............................................................. 89
9.1.67 PFBOFF ............................................................... 90
9.1.68 Feedback Type .................................................... 90
9.1.69 Halls ................................................................. 90
9.1.70 ANZERO1 ............................................................ 91
9.1.71 ANZERO2 ............................................................ 91
9.1.72 HSAVE .............................................................. 91
9.1.73 SININIT ............................................................. 92
9.1.74 ENCSTART ......................................................... 92
9.1.75 HWPOS .............................................................. 92
9.1.76 RDRS ................................................................. 92
9.1.77 RESBW .............................................................. 93
9.1.78 PRD ................................................................. 93

9.2 ORDER OF PARAMETER DOWNLOAD .......................... 93
9.3 ERROR CODES .......................................................... 94
9.4 ADDITIONAL ERROR CODES ....................................... 95
9.5 SYNQNET CYCLIC STATUS BITS ................................. 96

Table of Figures
FIGURE 8-1: MEMORY OPERATIONS .................................. 41
FIGURE 8-2: MOTIONLINK MAIN SCREEN ........................ 44
FIGURE 8-3: MOTIONLINK DRIVE BACKUP SCREEN .......... 45
FIGURE 8-4: LOCATION OF JUMPER FOR EMBER .................. 46
FIGURE 8-5: RESET ERROR MESSAGE ............................. 47
FIGURE 8-6: IGNITE28xx MAIN SCREEN ............................. 47
FIGURE 8-7: VM3 SCREEN SHOWING MONITORED DATA .......................................................... 67
FIGURE 8-8: SELECTING TRACES IN MOTIONSCOPE .................................................... 68
FIGURE 8-9: DEFINING NEW TRACES IN MOTIONSCOPE ............................................. 68
## 0. Revision History

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Creation</td>
</tr>
<tr>
<td>0.3</td>
<td>Added LED definitions</td>
</tr>
<tr>
<td></td>
<td>Refined I/O definition</td>
</tr>
<tr>
<td>0.4</td>
<td>Removed Resolver and Sine Encoder options</td>
</tr>
<tr>
<td>0.5</td>
<td>Part number: Encoder only</td>
</tr>
<tr>
<td></td>
<td>Motor type: linear motors</td>
</tr>
<tr>
<td></td>
<td>RJ-45 connector pin-out was added</td>
</tr>
<tr>
<td></td>
<td>More description of DGND signal for Fast Output</td>
</tr>
<tr>
<td></td>
<td>Analog supply failure is detected</td>
</tr>
<tr>
<td>0.6</td>
<td>Information on Fault Relay</td>
</tr>
<tr>
<td></td>
<td>Description on operating DBN</td>
</tr>
<tr>
<td></td>
<td>Updated parameter table</td>
</tr>
<tr>
<td></td>
<td>Added table of Direct Commands</td>
</tr>
<tr>
<td></td>
<td>Added more descriptions to the instruction set</td>
</tr>
<tr>
<td></td>
<td>Added more drive status indications</td>
</tr>
<tr>
<td></td>
<td>Added “SynqNet communication fault” to list of drive faults</td>
</tr>
<tr>
<td></td>
<td>Defined order in which parameters must be written</td>
</tr>
<tr>
<td>1.0</td>
<td>Additional functionality for MFBDIR (Firmware 0.1.1)</td>
</tr>
<tr>
<td></td>
<td>Added description of accessing analog inputs</td>
</tr>
<tr>
<td></td>
<td>Setting proper commutation</td>
</tr>
<tr>
<td></td>
<td>More complete description of Faults, Analog Inputs, Direct Commands, Real-time monitoring, including examples and screen captures</td>
</tr>
<tr>
<td></td>
<td>Added firmware upgrade procedure</td>
</tr>
<tr>
<td>1.1</td>
<td>Added wiring diagram</td>
</tr>
<tr>
<td>1.2</td>
<td>Added Powering Up description</td>
</tr>
<tr>
<td></td>
<td>Clarification on incremental encoder input frequency</td>
</tr>
<tr>
<td></td>
<td>Added information on 600V product</td>
</tr>
<tr>
<td></td>
<td>Added Mounting specifications</td>
</tr>
<tr>
<td></td>
<td>Added some wiring diagram info (wiring diagram for CD600 is still missing)</td>
</tr>
<tr>
<td></td>
<td>Additional direct commands (0x32 to 0x34) and parameters (0x2A to 0x3A)</td>
</tr>
<tr>
<td></td>
<td>Description of Commutation Initialization (Wake-no-shake)</td>
</tr>
<tr>
<td>1.3</td>
<td>Updated electrical schematics of general purpose digital outputs</td>
</tr>
<tr>
<td></td>
<td>Front panel I/O connector pin-out table updated: fault relay has no polarity</td>
</tr>
<tr>
<td></td>
<td>Direct commands: added examples of reading and writing drive parameters</td>
</tr>
<tr>
<td></td>
<td>Firmware Update section: Added description of Software EMBER mechanism</td>
</tr>
<tr>
<td></td>
<td>Moved description of Indicators to section on Operational Use</td>
</tr>
<tr>
<td></td>
<td>Wiring diagram for CD600</td>
</tr>
<tr>
<td>2.0</td>
<td>Brake output</td>
</tr>
<tr>
<td></td>
<td>Sine Encoder support</td>
</tr>
<tr>
<td>2.1</td>
<td>Updated EnDat operation information</td>
</tr>
<tr>
<td></td>
<td>Updated drive status LED description</td>
</tr>
<tr>
<td></td>
<td>Added fault code display priority</td>
</tr>
<tr>
<td>2.2</td>
<td>Encoder supply current: 400mA</td>
</tr>
<tr>
<td></td>
<td>Feedback type parameter (0x43): read-only, and not read/write</td>
</tr>
<tr>
<td></td>
<td>Real-time monitor on actual torque: index 10 and not 3. It always was 10 – the documentation was incorrect</td>
</tr>
<tr>
<td></td>
<td>Direct Commands 0x32 through 0x38 changed to drive Parameters, starting with 0x45</td>
</tr>
<tr>
<td></td>
<td>Added drive parameter to read Halls (0x44)</td>
</tr>
<tr>
<td></td>
<td>Added Halls States to cyclic status flags:</td>
</tr>
<tr>
<td></td>
<td>Hall A: Bit 3</td>
</tr>
<tr>
<td></td>
<td>Hall B: Bit 4</td>
</tr>
<tr>
<td></td>
<td>Hall C: Bit 5</td>
</tr>
<tr>
<td></td>
<td>Added Amp Active to bit 11 of cyclic status flags</td>
</tr>
<tr>
<td></td>
<td>Re-define cyclic status flags Amp_Power bit (bit 7): Set when drive is ready to receive AmpEnable (no faults and Remote Enable high)</td>
</tr>
<tr>
<td></td>
<td>Add Feedback Faults to upper word faults status word</td>
</tr>
<tr>
<td>2.3</td>
<td>Change to CD600 Electrical Specifications (OV and Regen)</td>
</tr>
<tr>
<td></td>
<td>Added detailed description of current loop parameters</td>
</tr>
<tr>
<td>Section</td>
<td>Changes</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>2.4</td>
<td>Added additional parameters 0x4D(RESBW), 0x4C(PRDF), added additional Direct commands 0x4A, 0x40, 0x41, and 0x42. Updated fault word (direct command 0x06). Operational ranges for parameters were updated.</td>
</tr>
<tr>
<td>2.5</td>
<td>Update Firmware Update section to show screen captures of new IGNITE. Added description of current loop gain. Correction to MFBDIR range.</td>
</tr>
<tr>
<td>2.7</td>
<td>Updated Warning indication. Updated Display description to describe decimal point on power up. Added fault descriptions to Display description.</td>
</tr>
</tbody>
</table>
1. Conventions

**Warning** identifies hazards that could result in personal injury or death.

**Caution** identifies hazards that could result in personal injury or equipment damage.

**Note** identifies information critical to the user’s understanding or use of the equipment.

2. Product Overview

2.1 Description

The **CD SynqNet** is a SynqNet™ servo drive based on the ServoSTAR™ CD. The power stage is the same as that of the CD, while the control stage is a dedicated SynqNet™ design. The **CD SynqNet** is designed as a Torque drive, with torque command being provided at high servo update rates from the SynqNet™ controller. Extensive I/O support is provided, with both function-specific inputs, such as Limit Switches, Home and Brake, and numerous general purpose I/Os. The **CD SynqNet** provides divide-by-N capability, enabling it to be programmed to generate an output pulse every N counts of the drive’s encoder counter register. Incremental encoder feedback is supported at hardware revision level 0, while additional types are added in later revisions.

2.2 Product Options

The **CD SynqNet** is available in both 300V and 600V models. The 300V model is based on a power stage that can take single- or three-phase power in the range 110-230VAC L-L. The 600V model is based on a power stage that can take 380VAC Line-to-Line or 480VAC Line-to-Line power. All models require separate logic power.

In addition to the differences in the power ratings, these models differ in the following characteristics:

- Electrical specifications
- Mounting
- Wiring diagram
2.3 Part Number

The following diagram illustrates the part number:

![Part Number Diagram](image)

**Note:** Hardware revision 0 supports encoder feedback only. Sine encoder and resolver feedback are available on hardware revision 1 and higher.
3. Product Block Diagram
4. Functional Capabilities

4.1 Servo Control

<table>
<thead>
<tr>
<th>Control loops</th>
<th>SynqNet Command</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Torque Mode</td>
</tr>
<tr>
<td></td>
<td>Velocity Mode (future option)</td>
</tr>
<tr>
<td>PWM switching frequency</td>
<td>16Khz for 3A/300V and 6A/300V drive</td>
</tr>
<tr>
<td></td>
<td>10A/300V and all 600V drives</td>
</tr>
</tbody>
</table>

4.2 Motor Type

The CD SynqNet drives Linear or Rotary motors. The **MOTORTYPE** parameter is used to define the motor type being driven.

4.3 Motor Feedback

4.3.1 Incremental Encoder

4.3.1.1 Description

The encoder interface includes three groups of wires:

1. A/B (and complements) lines make up the encoder quadrature signals. The signals are received differentially through line receivers.
2. The narrow Index pulse normally appears once per revolution and indicates a known physical position of the shaft. This pulse is received differentially through a line receiver.
3. Commutation signals provide information representing the approximate absolute location of the motor shaft. These signals come either from Hall effect devices in the motor, or commutation tracks on the encoder. From this information, the motor can sinusoidally commutate forward until the index signal is detected - at which time, true position is known. These signals are isolated by an opto-coupler and can be differential or open-collector type signals.

Separate power for encoder and commutation signals

Commutation initialization: either from Commutation signals or WNS (Selectable, using the **MENCTYPE** drive parameter)
### 4.3.1.2 Specifications

<table>
<thead>
<tr>
<th>Encoder Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Required Signals</strong></td>
</tr>
<tr>
<td>Types: A/B with Hall Channels</td>
</tr>
<tr>
<td>A/B/Index with Hall Channels</td>
</tr>
<tr>
<td><strong>Signal Type: A-quad-B and Marker Halls</strong></td>
</tr>
<tr>
<td>Differential: do not connect single-ended</td>
</tr>
<tr>
<td>Differential or Open Collector</td>
</tr>
<tr>
<td><strong>System Voltage</strong></td>
</tr>
<tr>
<td><strong>Maximum Input Frequency</strong></td>
</tr>
<tr>
<td><strong>Maximum Cable Length</strong></td>
</tr>
<tr>
<td><strong>Maximum Line Count</strong></td>
</tr>
<tr>
<td><strong>Maximum Supply Current from SERVOSTAR</strong></td>
</tr>
<tr>
<td><strong>Protection</strong></td>
</tr>
</tbody>
</table>

### 4.3.2 Sine Encoder

#### 4.3.2.1 Description

The CD SynqNet supports Heidenhain EnDat single- and multi-turn absolute encoders. The EnDat encoder interface consists of the following signals:

- **Sine and Cosine (A/B).** The A/B (and complements) lines make up the encoder quadrature signals. The signals are received differentially at 1V peak-to-peak amplitudes before being processed by the interpolation circuitry.

- **EnDat (SSI serial data).** Absolute position information is stored in the encoder and is serially communicated to the drive upon power up and after a feedback loss fault. The data is received synchronously by a clock signal provided by the drive. Absolute position is known immediately; therefore, an index signal is not needed.
### 4.3.2.2 Specifications

<table>
<thead>
<tr>
<th>Sine Encoder Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required Signals</td>
</tr>
<tr>
<td>Signal Level:</td>
</tr>
<tr>
<td>Signal Type:</td>
</tr>
<tr>
<td>A, B</td>
</tr>
<tr>
<td>EnDat (Data/Clock)</td>
</tr>
<tr>
<td>System Voltage</td>
</tr>
<tr>
<td>Maximum Supply Current from the SERVOSTAR for encoders</td>
</tr>
<tr>
<td>Maximum Input Frequency</td>
</tr>
<tr>
<td>Maximum Cable Length</td>
</tr>
<tr>
<td>Maximum Line Count per Motor</td>
</tr>
<tr>
<td>Electrical Cycle</td>
</tr>
<tr>
<td>Protection</td>
</tr>
<tr>
<td>Maximum Interpolation (input)</td>
</tr>
</tbody>
</table>

### 4.3.3 Resolver

#### 4.3.3.1 Description

The CD SynqNet can use single-speed (two poles) resolver feedback to monitor the motor shaft position. A resolver can be thought of as a transformer whose output is unique for any given shaft position (an absolute position feedback). The transformer is driven with a sine wave reference signal. Two AC signals are returned from the resolver into the Sine and Cosine inputs.

#### 4.3.3.2 Specifications

<table>
<thead>
<tr>
<th>Type</th>
<th>Control Transmitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformation Ratio</td>
<td>0.47 (dependant on the Resolver itself)</td>
</tr>
<tr>
<td>Modulation Frequency</td>
<td>8 kHz</td>
</tr>
<tr>
<td>Input Voltage (From Drive)</td>
<td>4.25 VAC</td>
</tr>
<tr>
<td>Max DC Resistance</td>
<td>120 Ohms (stator)</td>
</tr>
<tr>
<td>Max Drive Current</td>
<td>55mA AC-RMS</td>
</tr>
<tr>
<td>Output Voltage (To Drive)</td>
<td>2 VAC</td>
</tr>
<tr>
<td>Accuracy</td>
<td></td>
</tr>
<tr>
<td>ResBW = 300</td>
<td>+/-6.71 ArcMin</td>
</tr>
<tr>
<td>ResBW = 600</td>
<td>+/-5.87 ArcMin</td>
</tr>
</tbody>
</table>
Repeatability

<table>
<thead>
<tr>
<th>ResBW = 300</th>
<th>+/-3.13 ArcMin</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResBW = 600</td>
<td>+/-5.6 ArcMin</td>
</tr>
</tbody>
</table>

4.3.4 Connector
Female, 25-pin D-sub.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Resolver</th>
<th>Encoder</th>
<th>Sine Encoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sine High</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>Sine Low</td>
<td>/A</td>
<td>/A</td>
</tr>
<tr>
<td>3</td>
<td>Shield</td>
<td>Shield</td>
<td>Shield</td>
</tr>
<tr>
<td>4</td>
<td>Cosine High</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>Cosine Low</td>
<td>/B</td>
<td>/B</td>
</tr>
<tr>
<td>6</td>
<td>Shield</td>
<td>Shield</td>
<td>Shield</td>
</tr>
<tr>
<td>7</td>
<td>E5V Return</td>
<td>E5V Return</td>
<td>E5V Return</td>
</tr>
<tr>
<td>8</td>
<td>E5V Return</td>
<td>E5V Return</td>
<td>E5V Return</td>
</tr>
<tr>
<td>9</td>
<td>H1B</td>
<td>EnDat /Data</td>
<td>EnDat /Clock</td>
</tr>
<tr>
<td>10</td>
<td>H2B</td>
<td>EnDat /Data</td>
<td>EnDat /Clock</td>
</tr>
<tr>
<td>11</td>
<td>H3B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Shield</td>
<td>Shield</td>
<td>Shield</td>
</tr>
<tr>
<td>13</td>
<td>Thermostat High</td>
<td>Thermostat High</td>
<td>Thermostat High</td>
</tr>
<tr>
<td>14</td>
<td>Shield</td>
<td>Shield</td>
<td>Shield</td>
</tr>
<tr>
<td>15</td>
<td>Ref. High Out</td>
<td>Index</td>
<td>Index</td>
</tr>
<tr>
<td>16</td>
<td>Ref. Low Out</td>
<td>/Index</td>
<td>/Index</td>
</tr>
<tr>
<td>17</td>
<td>Shield</td>
<td>Shield</td>
<td>Shield</td>
</tr>
<tr>
<td>18</td>
<td>E5V Supply</td>
<td>E5V Supply</td>
<td>E5V Supply</td>
</tr>
<tr>
<td>19</td>
<td>E5V Supply</td>
<td>E5V Supply</td>
<td>E5V Supply</td>
</tr>
<tr>
<td>20</td>
<td>E5V Supply</td>
<td>E5V Supply</td>
<td>E5V Supply</td>
</tr>
<tr>
<td>21</td>
<td>Shield</td>
<td>Shield</td>
<td>Shield</td>
</tr>
<tr>
<td>22</td>
<td>H1A</td>
<td>EnDat Data</td>
<td>EnDat Clock</td>
</tr>
<tr>
<td>23</td>
<td>H2A</td>
<td>EnDat Clock</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>H3A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Thermostat Low</td>
<td>Thermostat Low</td>
<td>Thermostat Low</td>
</tr>
</tbody>
</table>
4.4 I/O

This section describes the control and status signals that pass between the CD SynqNet and the motion controller. The signals are organized in two connectors, one on the front panel, and the other on top (referred to as the Extended I/O Connector). “Provision For” comment means population only if needed or requested.

4.4.1 I/O Summary

4.4.1.1 Fault Relay

The CD SynqNet drive provides a dry-contact fault relay. This relay is closed as long as no faults exist, and open otherwise. The relay is controlled directly by the SynqNet FPGA, being connected to the FPGA NODE_ALARM output. The relay terminals are accessible on the front panel I/O connector.

The relay can carry a maximum of 1A. It closes within 3 milli-seconds, and opens within 2 milli-seconds.

The fault relay operation has to be enabled in order to function. This is done in MotionConsole by enabling the Node Alarm in the SqNode Summary window. See MoCon screen capture below.

4.4.1.2 Limits and Home

- Positive limit
- Negative limit
- Home

These three signals are connected directly to the SynqNet™ FPGA, for processing by SynqNet™ without DSP interaction. Limits and Home are connected to the front panel I/O connector.
4.4.1.3 Remote Enable

The Remote Enable signal is connected to the DSP, and the DSP thus controls the Enable state of the drive. This is in addition to the SynqNet™ Enable signal. The Remote Enable signal is connected to the front panel I/O connector.

4.4.1.4 Brake Output

An electromechanical relay is assembled internally for use as a brake output. This is a dry-contact relay. It can carry up to 2A, and is driven directly from the SynqNet Brake Apply output. An anti-Lenz diode is assembled; the connection to the brake output is thus polarized.

The Brake operation has to be enabled in order to function. This is done in MotionConsole by setting the Brake Mode to Delay in the Motor Summary window.

The power supply for the brake must be separate from that of the drive’s logic power.

4.4.1.5 General Purpose I/O

- 8 general purpose opto-isolated digital inputs. All of these inputs are bi-directional, i.e. the reference signal can be 5-24V or GND.
- 4 general purpose opto-isolated uni-directional digital outputs.

These I/O signals are wired to the SynqNet™ FPGA, and are controlled directly over SynqNet™ without interference by the DSP.

The 8 inputs and the 4 uni-directional outputs are connected to the extended I/O connector.

4.4.1.6 Divide-By-N Output

Divide-by-N generates an output pulse every N counts of a motor’s encoder counter register. The divisor is programmable. The following capabilities are supported for this function; more capabilities may be supported in future versions of this product.

- Enable/Disable: turn divide by n on and off
- Start position: If div-by-N is enabled, start generating pulses from this position
- Stop position: Stop generating pulses at this position
- Direction (forward or reverse): Indicates whether divide by n should work in the forward or reverse direction
- Period (number of encoder counts between divide by n pulses): The range is 1..65535.

The pulse width is fixed at 13μsec.

An output is dedicated to the Divide-by-N feature. The output circuit contains a fast opto-coupler, enabling generation of signals having sub-microsecond rise time. The Divide-by-N output appears on the extended I/O connector.

4.4.1.7 RS422 I/O

Two differential RS422 I/O ports are provided. The direction is selectable through SynqNet™. These I/O points appear on the Extended I/O connector.
4.4.1.8 Analog Inputs
2 differential analog inputs, connected to the DSP. The analog inputs have 12-bit resolution. The input voltage range is ±10V, which is scaled to the DSP’s input voltage range of 0 – 3V. The analog inputs have an accuracy of ±5% or better.

One analog input is connected to the front panel connector, and the other is connected to the Extended I/O connector.

4.4.2 I/O Electrical Characteristics

4.4.2.1 Limits, Home and Enable Digital Inputs

The user connects to the signals marked “HW_Ena” (Enable), “CW”, “CCW”, “Home”. The signals labeled “Front_CREF” is the reference signal for the I/O, and can be either 5-24V or GND – the inputs are bipolar.
4.4.2.2 General Purpose Digital Inputs

The following diagram shows the electrical interface for 4 of the 8 General Purpose digital inputs. The interface for the other 4 is exactly the same. On this diagram, the inputs are labeled In4 through In7.

The user connects to signals labeled “In4”, “In5”, “In6”, “In7”, and the reference signal is connected to “COM_In4to7”. The reference can be either 5-24V or GND – the inputs are bi-polar.

Similar circuitry applies to the other 4 digital inputs.
4.4.2.3 Brake Output

The following circuit diagram shows the electrical interface for the brake output.

The user connects to the signals labeled “Brake+” and “Brake-”.

![Circuit Diagram for Brake Output](image-url)
4.4.2.4 General Purpose Digital Outputs

The following circuit diagram shows the electrical interface for the four uni-directional general purpose digital outputs.

![Circuit Diagram](image)

The user connects to the signals labeled $DOUT2$, $DOUT3$, $DOUT4$ and $DOUT5$. Each pair of outputs share a common ground line.
4.4.2.5 Divide-by-N Output

The following circuit diagram shows the electrical interface for the Divide-by-N digital output.

The output is an open-collector stage. The pull-up resistor may be un-assembled if the customer’s device has its own pull-up resistor. VCC is +5VDC. The DGND is the common for the output collector of this output. The input stage DGND at the user end has to be connected to this DGND.

4.4.2.6 Analog Inputs

The following circuit diagram shows the electrical interface for the differential analog inputs.

The analog inputs may be in the range ±10Vdc.

4.4.3 I/O Connectors

4.4.3.1 Front Panel I/O

Weidmuller BL3.5/13, 13-pin. Mating connector is supplied with the drive.
All front panel digital /O is wired to the SynqNet™ FPGA, for direct processing through SynqNet™. Analog I/O is wired to the DSP.

<table>
<thead>
<tr>
<th>Pin no</th>
<th>Function</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Shield</td>
<td>Shield</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Analog in+</td>
<td>Differential analog input</td>
<td>±10Vdc</td>
</tr>
<tr>
<td>3</td>
<td>Analog in-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>AGND</td>
<td>Analog ground</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Fault relay</td>
<td>Fault relay dry contact</td>
<td>1 Amp</td>
</tr>
<tr>
<td>6</td>
<td>Fault relay</td>
<td>Fault relay dry contact</td>
<td>No polarity</td>
</tr>
<tr>
<td>7</td>
<td>CREF</td>
<td>Common rail for Digital inputs and outputs</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Enable</td>
<td>Remote Enable</td>
<td>Wired to DSP, sampled at 62.5μsec</td>
</tr>
<tr>
<td>9</td>
<td>CW</td>
<td>Positive limit</td>
<td>5-24V; Wired to SynqNet™ FPGA</td>
</tr>
<tr>
<td>10</td>
<td>CCW</td>
<td>Negative limit</td>
<td>5-24V; Wired to SynqNet™ FPGA</td>
</tr>
<tr>
<td>11</td>
<td>HOME</td>
<td>Home input</td>
<td>5-24V; Wired to SynqNet™ FPGA</td>
</tr>
<tr>
<td>12</td>
<td>Brake+</td>
<td>Brake relay positive terminal</td>
<td>Dry-contact for brake control</td>
</tr>
<tr>
<td>13</td>
<td>Brake-</td>
<td>Brake relay negative terminal</td>
<td></td>
</tr>
</tbody>
</table>

### 4.4.3.2 Extended I/O Connector

Male, 25-pin D-sub.

All digital Extended I/O is wired to the SynqNet FPGA, for direct processing by SynqNet™ without interaction from the DSP. Digital inputs on this connector are numbered IN4 through IN11, in consideration of 3 digital inputs that are on the front panel I/O connector. Digital outputs are similarly numbered OUT2 through OUT5.

<table>
<thead>
<tr>
<th>Pin no</th>
<th>Function</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Analog in+</td>
<td>Differential analog input</td>
<td>±10Vdc</td>
</tr>
<tr>
<td>14</td>
<td>Analog in-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>AGND</td>
<td>Analog ground</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>IN4</td>
<td>Digital Input #4</td>
<td>5V – 24V, Bi- polar Wired to SynqNet™ FPGA</td>
</tr>
<tr>
<td>3</td>
<td>IN5</td>
<td>Digital Input #5</td>
<td>5V – 24V, Bi-polar Wired to SynqNet™ FPGA</td>
</tr>
<tr>
<td>16</td>
<td>IN6</td>
<td>Digital Input #6</td>
<td>5V – 24V, Bi- polar Wired to SynqNet™ FPGA</td>
</tr>
<tr>
<td>4</td>
<td>IN7</td>
<td>Digital Input #7</td>
<td>5V – 24V, Bi- polar Wired to SynqNet™ FPGA</td>
</tr>
<tr>
<td>17</td>
<td>Common IN4 to IN7</td>
<td>Common IN4 to IN7</td>
<td>Common for Inputs 4 to 7</td>
</tr>
<tr>
<td>5</td>
<td>IN8</td>
<td>Digital Input #8</td>
<td>5V – 24V, Bi- polar Wired to SynqNet™ FPGA</td>
</tr>
<tr>
<td>18</td>
<td>IN9</td>
<td>Digital Input #9</td>
<td>5V – 24V, Bi- polar Wired to SynqNet™ FPGA</td>
</tr>
<tr>
<td>6</td>
<td>IN10</td>
<td>Digital Input #10</td>
<td>5V – 24V, Bi- polar Wired to SynqNet™ FPGA</td>
</tr>
</tbody>
</table>
### Pin no | Function | Description | Comments
--- | --- | --- | -----
19 | IN11 | Digital Input #11 | 5V – 24V, Bi-polar
| | | | Wired to SynqNet™ FPGA
7 | Common IN8 to IN11 | Common IN8 to IN11 | Common for Inputs 8 to 11
20 | OUT2 | Digital output #2 | Open collector
| | | | Wired to SynqNet™ FPGA
8 | OUT3 | Digital output #3 | Open collector
| | | | Wired to SynqNet™ FPGA
21 | Out common 2 to 3 | Out common 2 to 3 | Common user ground for OUT2 and OUT3
9 | OUT4 | Digital output #4 | Open collector
| | | | Wired to SynqNet™ FPGA
22 | OUT5 | Digital output #5 | Open collector
| | | | Wired to SynqNet™ FPGA
10 | Out common 4 to 5 | Out common 4 to 5 | Common user ground for OUT4 and OUT5
23 | Diff_IO_1+ | Differential RS422 I/O (high) | Direction (In or Out) programmable through SynqNet™.
| 11 | Diff_IO_1- | Differential RS422 I/O (low) | Direction (In or Out) programmable through SynqNet™.
24 | Diff_IO_2+ | Differential RS422 I/O (high) | Direction (In or Out) programmable through SynqNet™.
| 12 | Diff_IO_2- | Differential RS422 I/O (low) | Direction (In or Out) programmable through SynqNet™.
25 | DIV_BY_N | Fast output, used for Divide-by-N signal | Open collector, with internal pull-up that may be dis-assembled.
13 | DGND | Digital Ground | The DGND is the common for the output collector of this output. The input stage DGND at the user end has to be connected to this DGND.

### 4.5 Communications

#### 4.5.1 SynqNet

##### 4.5.1.1 Connector

Shielded RJ-45 connectors are used for the SynqNet™ connection. There are two connectors, one for SynqNet™ IN and one for SynqNet™ OUT. The connectors have built-in LEDs to display SynqNet™ link status (see section on Indicators).

##### 4.5.1.2 Pin-Out

<table>
<thead>
<tr>
<th>Pin no</th>
<th>RJ45 In</th>
<th>RJ45 Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TD2+</td>
<td>TD1+</td>
</tr>
<tr>
<td>2</td>
<td>TVDD</td>
<td>TVDD</td>
</tr>
<tr>
<td>3</td>
<td>TD2-</td>
<td>TD1-</td>
</tr>
<tr>
<td>4</td>
<td>RD2+</td>
<td>RD1+</td>
</tr>
<tr>
<td>5</td>
<td>TVDD</td>
<td>TVDD</td>
</tr>
<tr>
<td>6</td>
<td>RD2-</td>
<td>RD1-</td>
</tr>
<tr>
<td>7</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>8</td>
<td>DGND</td>
<td>DGND</td>
</tr>
</tbody>
</table>
4.5.2 Serial RS-232

4.5.2.1 Connector
Male, 9-pin D-sub.

4.5.2.2 Pin-Out

<table>
<thead>
<tr>
<th>Pin no</th>
<th>Function</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N.C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>RxD</td>
<td>Receive</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TxD</td>
<td>Transmit</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>N.C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DGND</td>
<td>Ground</td>
<td>Must be connected in order to equalize potential between controller and drive.</td>
</tr>
<tr>
<td>6</td>
<td>N.C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>N.C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>N.C.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>N.C.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.6 Switches
The CD SynqNet™ drive is equipped with 1 10-pole rotary switch. The switch is connected to the SynqNet™ FPGA and its use is application specific. The switch is mounted on the top of the drive.

4.7 Protection
This section describes the faults that the drive will react to.

<table>
<thead>
<tr>
<th>Fault</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Over-current</td>
<td>The amplifier will be shut off if the current exceeds 53A. The over-current fault can be cleared only by cycling the power</td>
</tr>
<tr>
<td>Over voltage</td>
<td>The drive is disabled when the bus voltage exceeds 430V.</td>
</tr>
<tr>
<td>Under voltage</td>
<td>The drive is disabled when the bus voltage drops below 90Vdc and recovers at 120Vdc.</td>
</tr>
<tr>
<td>Drive over-temperature</td>
<td>Temperature will be monitored once on the entire module, with the probe assembled on the heat sink. At temperatures exceeding 80°C, the drive will be disabled.</td>
</tr>
<tr>
<td>Motor over-temperature</td>
<td>The drive is disabled if the motor over-heats. This capability requires the existence of a thermostat in the motor.</td>
</tr>
<tr>
<td>Current foldback (I2t)</td>
<td>I2t is used for protection of both the motor and the drive, with the limit being set for whichever of the devices is weaker. This is a warning: the current folds back but the drive will not be automatically disabled when foldback occurs.</td>
</tr>
<tr>
<td>Feedback loss</td>
<td>The drive will be disabled if the feedback cable is disconnected, or if any one of the feedback wires breaks. In some cases this is determined in software, and so the fault is detected in some cases only if the motor is turning.</td>
</tr>
<tr>
<td>Positive analog supply fail</td>
<td>Failure in +12V supply</td>
</tr>
</tbody>
</table>
Negative analog supply fail | Failure in -12V supply

4.8 Secondary Encoder
A secondary encoder input exists on the CD SynqNet™ drive. The encoder signals are connected internally to the FPGA.

4.8.1 Connector
Female, 9-pin D-sub

4.8.2 Pin-Out

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Input + (High)</td>
</tr>
<tr>
<td>2</td>
<td>A Input - (Low)</td>
</tr>
<tr>
<td>3</td>
<td>DC Common</td>
</tr>
<tr>
<td>4</td>
<td>B Input + (High)</td>
</tr>
<tr>
<td>5</td>
<td>B Input - (Low)</td>
</tr>
<tr>
<td>6</td>
<td>Shield Connection</td>
</tr>
<tr>
<td>7</td>
<td>E5V Supply</td>
</tr>
<tr>
<td>8</td>
<td>Index +</td>
</tr>
<tr>
<td>9</td>
<td>Index -</td>
</tr>
</tbody>
</table>
5. Electrical Specifications

5.1 300V Model

<table>
<thead>
<tr>
<th>Product Model</th>
<th>Lx03</th>
<th>Lx06</th>
<th>Lx10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage (VACₗₗ) Nominal ±10%</td>
<td>110 to 230</td>
<td>230</td>
<td></td>
</tr>
<tr>
<td>115VAC</td>
<td>1φ or 3φ</td>
<td>3φ only</td>
<td></td>
</tr>
<tr>
<td>230VAC</td>
<td>1φ or 3φ</td>
<td>3φ only</td>
<td></td>
</tr>
<tr>
<td>Line Frequency</td>
<td>47-63</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Main Input Power**

<table>
<thead>
<tr>
<th></th>
<th>Lx03</th>
<th>Lx06</th>
<th>Lx10</th>
</tr>
</thead>
<tbody>
<tr>
<td>KVA at 115</td>
<td>0.44 (1φ)</td>
<td>0.89 (1φ)</td>
<td>2.4 (3φ only)</td>
</tr>
<tr>
<td></td>
<td>0.6 (3φ)</td>
<td>1.1 (3φ)</td>
<td></td>
</tr>
<tr>
<td>KVA at 230 VAC</td>
<td>0.88 (1φ)</td>
<td>1.8 (1φ)</td>
<td>4.6 (3φ only)</td>
</tr>
<tr>
<td></td>
<td>1.4 (3φ)</td>
<td>2.8 (3φ)</td>
<td></td>
</tr>
<tr>
<td>Continuous Current (amps)</td>
<td>6.2 (1φ)</td>
<td>10 (1φ)</td>
<td>13 (3φ only)</td>
</tr>
<tr>
<td></td>
<td>4 (3φ)</td>
<td>7.8 (3φ)</td>
<td></td>
</tr>
<tr>
<td>Peak Current (amps) for 500 msec</td>
<td>18.6 (1φ)</td>
<td>30 (1φ)</td>
<td>26 (3φ only)</td>
</tr>
<tr>
<td></td>
<td>12 (3φ)</td>
<td>23.4 (3φ)</td>
<td></td>
</tr>
<tr>
<td>Peak Current (amps) for 2 Sec</td>
<td>12.4 (1φ)</td>
<td>20 (1φ)</td>
<td>26 (3φ only)</td>
</tr>
<tr>
<td></td>
<td>8 (3φ)</td>
<td>15.6 (3φ)</td>
<td></td>
</tr>
<tr>
<td>Line Fuses (FRN-R, LPN, or equivalent)</td>
<td>10</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>

**Logic Input Power**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>+24 VDC Ext. Logic Voltage (volts)</td>
<td>22 to 27</td>
</tr>
<tr>
<td>+24 VDC Ext. Logic Current (amps sink)</td>
<td>600mA</td>
</tr>
<tr>
<td>+24 VDC Ext. Logic Current (amps max surge)</td>
<td>2A for 5msec, and then 1.5A for 7msec</td>
</tr>
</tbody>
</table>

**SoftStart**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Surge Current (amps)</td>
<td>30</td>
</tr>
<tr>
<td>Max. Charge Time (sec)</td>
<td>0.25</td>
</tr>
</tbody>
</table>

**Protection Functions**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault Contact Rating</td>
<td>1A</td>
</tr>
<tr>
<td>Fault Contact Closing Period (mSec)</td>
<td>Close = 3 mS, Open = 2 mS</td>
</tr>
<tr>
<td>OverTemperature trip</td>
<td>80°C (176°F)</td>
</tr>
</tbody>
</table>

**Rated Main Output (Ma, Mb, Mc)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Power (KVA) at 115 VAC Line Input (45°C (113°F) Ambient)</td>
<td>0.35 (1φ)</td>
</tr>
<tr>
<td></td>
<td>0.6 (3φ)</td>
</tr>
<tr>
<td></td>
<td>0.7 (1φ)</td>
</tr>
<tr>
<td></td>
<td>1.1 (3φ)</td>
</tr>
<tr>
<td>Continuous Power (KVA) at 230 VAC Line Input (45°C (113°F) Ambient)</td>
<td>0.7 (1φ)</td>
</tr>
<tr>
<td></td>
<td>1.1 (3φ)</td>
</tr>
<tr>
<td></td>
<td>1.4 (1φ)</td>
</tr>
<tr>
<td></td>
<td>2.2 (3φ)</td>
</tr>
<tr>
<td>Continuous Current (Arms)</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td>Peak Current (Arms) for 500 mSec</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>20</td>
</tr>
<tr>
<td>Peak Current (Arms) for 2 Sec</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>20</td>
</tr>
<tr>
<td>PWM Frequency (kHz)</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td>PWM Motor Current Ripple (kHz)</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td>Form Factor (rms/avg)</td>
<td>≤1.01</td>
</tr>
<tr>
<td>UnderVoltage Trip (nominal)</td>
<td>90 VDC</td>
</tr>
</tbody>
</table>

1 The drive will continue operating through a logic power drop-out of up to 14msec. If logic power is removed for 15msec or longer, the drive will behave as if power was cycled.
### Protective Functions

<table>
<thead>
<tr>
<th></th>
<th>Lx03</th>
<th>Lx06</th>
<th>Lx10</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnderVoltage Trip (nominal)</td>
<td>90 VDC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OverVoltage Trip (nominal)</td>
<td>430 VDC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OverTemperature Trip</td>
<td>176 °F</td>
<td>80 °C</td>
<td>132 °C</td>
</tr>
<tr>
<td>Internal heat dissipation (watts)</td>
<td>60</td>
<td>80</td>
<td>132</td>
</tr>
</tbody>
</table>

### Environment

<table>
<thead>
<tr>
<th></th>
<th>Lx03</th>
<th>Lx06</th>
<th>Lx10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation temperature</td>
<td>5°C (41°F) to 45°C (113°F)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage temperature</td>
<td>0°C (32°F) to 70°C (158°F)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Atmosphere</td>
<td>Without corrosive gasses or dust</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Altitude</td>
<td>Derate 5% per 1000 ft. (300m) above 3300 ft. (1000m)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vibration</td>
<td>0.5 g</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Altitude</td>
<td>Derate 5% per 1000 ft. (300m) above 3300 ft. (1000m)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.1.2 Regen Information

#### External Shunt Regulator

<table>
<thead>
<tr>
<th></th>
<th>Cx03</th>
<th>Cx06</th>
<th>Cx10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak current (amps)</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum resistance (ohms)</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Watts</td>
<td>200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Application Information

<table>
<thead>
<tr>
<th></th>
<th>Cx03</th>
<th>Cx06</th>
<th>Cx10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS Voltage (nominal) (VDC)</td>
<td>325</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V_{HYS} (Regen circuit turn-off) (VDC)</td>
<td>370</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V_{MAX} (Regen circuit turn-on) (VDC)</td>
<td>390</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### External Regen Kits

- ERH-26

### 5.2 600V Option

#### 5.2.1 Electrical Specification

<table>
<thead>
<tr>
<th></th>
<th>Lx06665</th>
<th>Lx10665</th>
</tr>
</thead>
<tbody>
<tr>
<td>Voltage (VAC_L-L) Nominal ±10%</td>
<td>380VAC Line-to-Line or 480VAC Line-to-Line</td>
<td>3φ only</td>
</tr>
<tr>
<td>Line Frequency (Hz)</td>
<td>47-63</td>
<td></td>
</tr>
<tr>
<td>KVA at 380VAC</td>
<td>4.6</td>
<td>7.5</td>
</tr>
<tr>
<td>KVA at 480VAC</td>
<td>5.8</td>
<td>9</td>
</tr>
<tr>
<td>Continuous Current (amps)</td>
<td>10 (3φ)</td>
<td>16 (3φ)</td>
</tr>
<tr>
<td>Line Fuses (FRN-R, LPN, or equivalent)</td>
<td>20Amp</td>
<td>25Amp</td>
</tr>
<tr>
<td>+24 VDC Ext. Logic Voltage (volts)</td>
<td>22 to 27</td>
<td></td>
</tr>
<tr>
<td>+24 VDC Ext. Logic Current (amps sink)</td>
<td>600mA</td>
<td></td>
</tr>
<tr>
<td>+24 VDC Ext. Logic Current (amps max surge)</td>
<td>2A for 5msec, and then 1.5A for 7msec</td>
<td></td>
</tr>
<tr>
<td>SoftStart Max. Surge Current (amps)</td>
<td>135 (For 380VAC); 170 (For 480VAC)</td>
<td></td>
</tr>
<tr>
<td>Max. Charge Time (sec)</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>Fault Contact Rating</td>
<td>1A</td>
<td></td>
</tr>
<tr>
<td>Rated Main Output (Ma, Mb, Mc)</td>
<td>Product Model</td>
<td>Lx06665</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
<td>---------</td>
</tr>
<tr>
<td>Fault Contact Rating</td>
<td></td>
<td>1A</td>
</tr>
<tr>
<td>Over Duty Closing Time (ms)</td>
<td>Close = 3 ms, Open = 2 ms</td>
<td></td>
</tr>
<tr>
<td>Continuous Power (KVA) at 380VAC</td>
<td>2.2</td>
<td>3.5</td>
</tr>
<tr>
<td>Line Input (45°C (113°F) Ambient)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous Power (KVA) at 480VAC</td>
<td>2.9</td>
<td>4.6</td>
</tr>
<tr>
<td>Line Input (45°C (113°F) Ambient)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Continuous Current (Arms)</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>Peak Current (Arms) 12 (for 2 Sec)</td>
<td>18 (for 500 mSec)</td>
<td></td>
</tr>
<tr>
<td>Peak Current (Arms) for 2 Sec</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>PWM Frequency (kHz)</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>PWM Motor Current Ripple (kHz)</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Protective Functions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>UnderVoltage Trip (nominal)</td>
<td>370 VDC</td>
</tr>
<tr>
<td>OverVoltage Trip</td>
<td>890 VDC</td>
</tr>
<tr>
<td>OverTemperature Trip (On heat sink)</td>
<td>80°C (176°F)</td>
</tr>
<tr>
<td>Internal heat dissipation (watts)</td>
<td>130</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Environment</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation temperature</td>
<td>5°C (41°F) to 45°C (113°F)</td>
</tr>
<tr>
<td>Storage temperature</td>
<td>0°C (32°F) to 70°C (158°F)</td>
</tr>
<tr>
<td>Ambient humidity</td>
<td>10% to 90%</td>
</tr>
<tr>
<td>Atmosphere</td>
<td>Without corrosive gasses or dust</td>
</tr>
<tr>
<td>Altitude</td>
<td>Derate 5% per 1000 ft. (300m) above 3300 ft. (1000m)</td>
</tr>
<tr>
<td>Vibration</td>
<td>0.5 g</td>
</tr>
</tbody>
</table>

5.2.2 Regen Information

<table>
<thead>
<tr>
<th>Application Information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak current (amps)</td>
<td>15</td>
</tr>
<tr>
<td>Minimum resistance (ohms)</td>
<td>70</td>
</tr>
<tr>
<td>Recommended: 80 - 90</td>
<td></td>
</tr>
<tr>
<td>Watts</td>
<td>200</td>
</tr>
<tr>
<td>Capacitance (Farads)</td>
<td>0.000470 (470µF)</td>
</tr>
<tr>
<td>BUS Voltage (nominal) (VDC)</td>
<td>540 (380VAC); 680 (480VAC)</td>
</tr>
<tr>
<td>V_HYS (Regen circuit turn-off) (VDC)</td>
<td>855</td>
</tr>
<tr>
<td>V_MAX (Regen circuit turn-on) (VDC)</td>
<td>875</td>
</tr>
</tbody>
</table>
6. Mounting

6.1 300V Model

6.1.1 Hardware Specifications

<table>
<thead>
<tr>
<th>Amplifier Model</th>
<th>Lx03</th>
<th>Lx06</th>
<th>Lx10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lbs./Kgs.</td>
<td>3.56/1.61</td>
<td>4.9/2.22</td>
<td>5.94/2.69</td>
</tr>
<tr>
<td>English (Metric)</td>
<td>10-32 (M4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applied Torque</td>
<td>20 lb-in. (2.26 Nm.)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Connection Hardware**

<table>
<thead>
<tr>
<th>Line Screw Size/Torque</th>
<th>M3.5/12 lb-in. (1.35 Nm.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUS Screw Size/Torque</td>
<td></td>
</tr>
<tr>
<td>Motor Screw Size/Torque</td>
<td></td>
</tr>
<tr>
<td>Ground Screw Size/Torque</td>
<td></td>
</tr>
</tbody>
</table>

**Wire Size (AWG#)**

<table>
<thead>
<tr>
<th>Control Logic (AWG/ mm²)</th>
<th>16 / 1.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor Line (AWG/ mm²)</td>
<td>14 / 2.5</td>
</tr>
<tr>
<td>Main Input (AWG/ mm²)</td>
<td>14 / 2.5</td>
</tr>
<tr>
<td></td>
<td>12 / 4</td>
</tr>
</tbody>
</table>
| Configurable I/O wire gauge | 22-18 AWG (0.3-0.75 mm²) Ferrules recommended:
|                           | 18 AWG Type H0 - 75/14 Weidmuller 4629.0 or equivalent
|                           | 20 AWG Type H0 - 5/14 Weidmuller 6907.0 or equivalent
|                           | 22 AWG Type H0 – 34/12 Weidmuller 902577 or equivalent
| Spade Terminals           | 16/14 AWG (1.5 mm²): Hollingsworth XSS0954S OR SS20947SF or equivalent
|                           | 12/10 AWG (4-6 mm²): Hollingsworth XSS20836 OR SS20832F or equivalent
| Clearances Distance       | Side-to-Side: 0.5 in (12.7mm)
|                           | Top/Bottom: 2.5 in (63.5mm)
| Mating Connector Hardware | CK100 Kit Includes: C1, C2, C4, C7 (plus 2 ft./0.69 m. of stranded bus ribbon), C8
|                           | C3 Kollmorgen #: A-93899-013 Vendor Info: Weidmuller BL3.5/13 Cat.No. 161574
|                           | C5 Kollmorgen #: A-81014-004 Vendor Info: PCD ELFP04110
|                           | Connector Screw Torque 2.25 lb-in. (0.25 m.)
|                           | 24 V Logic (optional) Kollmorgen #: A-81014-002 Vendor Info: PCD ELFP02210 (or equiv.)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com
6.1.2 Outline Dimensions

6.2 600V Model

6.2.1 Hardware Specifications

<table>
<thead>
<tr>
<th>Unit Weight</th>
<th>Lbs./Kgs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mounting Hardware</td>
<td>English (Metric)</td>
</tr>
<tr>
<td>Ground Screw Size/Torque</td>
<td>M3.5/12 lb-in. (1.35 Nm.)</td>
</tr>
<tr>
<td>Control Logic (AWG/ mm²)</td>
<td>16 / 1.5</td>
</tr>
<tr>
<td>Motor Line (AWG/ mm²)</td>
<td>14 / 2.5</td>
</tr>
<tr>
<td>Main Input (AWG/ mm²)</td>
<td>12 / 4</td>
</tr>
</tbody>
</table>
### Configurable I/O wire gauge

<table>
<thead>
<tr>
<th>Wire Size (AWG#)</th>
<th>Configurable I/O wire gauge</th>
<th>22-18 AWG (0.3-0.75 mm²) Ferrules recommended:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>18 AWG Type H0 - 75/14 Weidmuller 4629.0 or equivalent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>20 AWG Type H0 - 5/14 Weidmuller 6907.0 or equivalent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22 AWG Type H0 – 3/16 Weidmuller 902577 or equivalent</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clearance Distance</th>
<th>Side-to-Side</th>
<th>0.5in (12.7mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top/Bottom</td>
<td>2.5in (63.5mm)</td>
</tr>
</tbody>
</table>

#### CK100 Kit

- Includes: C1, C2, C4, C7 (plus 2 ft./0.69 m. of stranded bus ribbon), C8
- Kollmorgen #: A-93899-013
- Vendor Info: Weidmuller BL3.5/13
- Cat.No. 161574

#### Connector Screw Torque

- 2.25 lb-in. (0.25 m.)

#### Motor Power Connector

- Phoenix Contact # 1880038

#### Regen Connector

- Phoenix Contact # 1848892

#### 24 V Logic

- PCD ELFP02210

### 6.2.2 Outline Dimensions

All dimensions are in mm.

- Height (including fan): 255
- Width: 110
- Depth (excluding mating connectors): 185.50
6.2.2.1 Front View
6.2.2.2 Top View
6.2.2.3 Side View
7. Wiring Diagram

7.1 300V Model: 3A, 6A, 10A
7.2 600V Model and 20A 300V Model

---

**WARNING**
HIGH VOLTAGE MAY EXIST UP TO 5 MINUTES AFTER INPUT VOLTAGE IS REMOVED.

---

**WARNING**
B I A

---

**C8**
9 6 5 3 1

---

COMMON
RESERVED

---

REMOTE ENCODER INPUT
FEEDBACK (D25)

---

TO MOTOR
FEEDBACK
DEVICE

---

24V
RTN

---

C USER SUPPLIED
POWER SUPPLY

---

FAULT OUTPUT RELAY
(DRY CONTACTS)

---

ANALOG INPUT +/- 10V

---

REMOTE  ENABLE
HIGH
LOW
DC REFERENCE FOR ANOUT
SUPPLY
SUPPLY

---

(IN1)
(IN2)
(IN3)
(O1)

---

CONFIG. INPUT
CONFIG. INPUT
CONFIG. INPUT

---

25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

---

C2
C1
C3

---

6 7 8 9
1 2 3 4 5

---

6
7
8
9
1
2
3
4
5
6
7
8
9
10
11
12
13

---

XMP-SynqNet-xxx-RJ

---

Regen
Resistor

---

THERMOSTAT

---

E-STOP

---

EXTERNAL RESISTIVE
REGEN UNIT

---

MOMENTARY
PUSHBUTTON
TO ENERGIZE
MAIN POWER.

---

FUSE 1
FUSE 2
FUSE 3

---

CUSTOMER SUPPLIED
CONFIGUREABLE I/O INTERFACE

---

COMMUNICATIONS (D9S)

---

DIAGRAM REPRESENTS
TYPICAL CONNECTION

---

STAR
POINT

---

SHIELD         TO
STAR POINT

---

THE CD OPERATES WITH VOLTAGES
AS HIGH AS 480VAC AND 670VDC. USE CAUTION.
REFER INSTALLATION AND TROUBLESHOOTING
TO QUALIFIED PERSONNEL ONLY.

---

ARTISAN TECHNOLOGY GROUP - QUALITY INSTRUMENTATION... GUARANTEED | (888) 88-SOURCE | www.artisantg.com
7.3 SynqNet Cables
Use standard shielded CAT5e RJ45 cables. These can be purchased from MEI or MOLEX in lengths of 1.2m, 3m and 10m, or from most electronics and office equipment stores. Refer to http://www.synqnet.org/cabling.html

8. Operational Use

8.1 Powering Up
One of the characteristics of SynqNet drives is that at power up, the drive DSP is held in a RESET state by the SynqNet FPGA in the drive. The DSP reset is only released when a SynqNet RESET command is issued from the controller. Once this is done, the drive will come up.

The SynqNet RESET can be done in two ways:
- Using MotionConsole: Click on the RESET button in the Controller Summary window (on the ACTIONS tab)
- Using a DOS command:
  - Open a DOS PROMPT window in C:\MEI\XMP\Bin\ (assuming MotionConsole has been installed to C:\MEI)
  - Type RESET at the command prompt

8.2 Current Scaling
The drive multiplies currents by 0.8, and therefore full scale controller torque (32767) is equivalent to DIPEAK * SQRT(2) / 0.8, or

\[
\text{DIPEAK} \times \text{SQRT}(2) = 32767 \times 80\%.
\]

Where DIPEAK is the drive peak current rating in units of Arms.

8.3 Drive Configuration
The drive functionality is set using various drive parameters and instructions, which are communicated over the serial port or over SynqNet™. The parameters are stored in non-volatile memory in the drive, to be used on each power-up. A list of parameters appears in Section titled Table of CD SynqNet Parameters.

8.3.1 Current Loop Gain
The Current controller gain is proportional to \((G*I*L/V)\), where
- \(G\) is the adaptive gain
- \(I\) is DIPEAK
- \(L\) is MLMIN
- \(V\) is VBUS.
**VBUS, MLMIN**

These two parameters set the basic gain of the current loop. The current loop gain is directly proportional to MLMIN, but inversely proportional to VBUS. The current loop gain is further modified by MLGAINC and MLGAINP.

**MLGAINC, MLGAINP**

Set the adaptive gain of the current controller as a function of the motor current.

MLGAINC sets the adaptive gain at motor continuous current (MICONT).

MLGAINP sets the adaptive gain at motor peak current (MIPEAK).

Together, with unity gain at zero current, it creates two piece linear curve, where the drive calculates the adaptive gain for a given motor current.

The current-based adaptive gain algorithm is a gain calculation method that increases current loop stability by reducing the current loop gain as the motor current increases (there are motors that their magnetic flux decreases when the current increases, and so their gain increases). A value of 10 (unity gain) is a good starting point.

MLGAINC, MLGAINP units are gain*0.1 (10 means unity gain).

MLGAINC, MLGAINP range is 1 to 100.

**8.3.2 Back-EMF Compensation**

**MBEMFCOMP, MKT**

MBEMFCOMP sets the amount of BEMF compensation, while MKT is part of the Back-EMF compensation algorithm.

The Back-EMF is a feed-forward for the current loop. It takes the velocity, multiplies it by motor torque constant (set using MKT), and multiplies the result by the specified Back-EMF gain (MBEMFCOMP). Then it performs commutation (multiplies the above result by the same sin, sin+120, and sin+240, as it multiplies the current command). The results (one per phase) are added to the output of the corresponding current controller outputs, and the sums generate the PWM commands for each phase. One can consider the Back-EMF of the motor as a disturbance to the current loop. The drive has the capability to estimate the amount of Back-EMF, and to inject feed-forward correction.

Higher currents will be used when Back-EMF compensation is on, since the Back-EMF is a feed-forward to the current loop. The advantage of using Back-EMF compensation is that it bypass the current controller with its finite bandwidth (the back EMF comp is a gain only, and has "unlimited" bandwidth). As the motor speed increases, the commutation frequency increases, the current controller gets closer to its bandwidth, and the Back-EMF compensation effect is more emphasized. The disadvantage with using Back-EMF compensation is that it injects noise since it has "unlimited" bandwidth. As always, a balance needs to be found.

MBEMFCOMP units are percentage.

MBEMFCOMP range is 0 to 130.

MBEMFCOMP=0 means no BEMF compensation.

MBEMFCOMP=100 means that BEMF compensation equals to the estimated BEMF.

Typical values of MBEMFCOMP are 50 to 80.
8.3.3 Phase Angle Advance

**MVANGLH, MVANGLF**

Set the phase advance angle as a function of the motor velocity.

MVANGLH sets the phase advance angle at half of motor speed (MSPEED/2) in electrical degrees.

MVANGLF sets the phase advance angle at motor speed (MSPEED) in electrical degrees.

Together, with zero angle advance at zero speed, it creates two piece linear curve, where the drive calculates the phase advance for a given motor speed.

The phase advance helps to achieve higher torque for a given current and a given motor speed. These parameters are independent of the rotor magnets. They come to compensate for computing time and current loop phase lag. In general, they shouldn't be set to 0.

MVANGLH, MVANGLF units are electrical degrees.

MVANGLH, MVANGLF range is 0 to 90.

**MTANGLC, MTANGLP**

Set the phase advance angle as a function of the motor current.

MTANGLC sets the phase advance angle at motor continuous current (MICONT) in electrical degrees.

MTANGLP sets the phase advance angle at motor peak current (MIPEAK) in electrical degrees.

Together, with zero angle advance at zero current, it creates two piece linear curve, where the drive calculates the phase advance for a given motor current.

The phase advance helps to achieve higher torque for a given motor current. Usually, it is applicable for buried magnet rotor (as opposed to surface mounted magnet rotor). For a surface mounted magnet rotor the MTANGLx parameters should be set to 0. However, if MVANGLx parameters were not set optimally, non-zero MTANGLx parameters can help to get more torque.

MTANGLC, MTANGLP units are electrical degrees.

MTANGLC, MTANGLP range is 0 to 45.

8.4 Memory Descriptions

The drive contains a number of different memory types:

- Flash Memory: used to store the drive firmware
- RAM: used to store drive parameters during run-time
- DSP EEPROM: non-volatile memory used to store drive parameters even when the power is off

At power up, the drive will attempt to load parameter values from the EEPROM into the RAM. A checksum of these parameter values is kept, and this is verified when the EEPROM contents are loaded. If the checksum is invalid, default values for drive parameters are loaded into RAM. These default values are hard-coded, and are as such part of the firmware file.

When parameter values are set, these values are stored in RAM, and will be lost when power is removed form the drive. Once a working set of drive parameters has been found, the parameters can be stored in non-volatile EEPROM memory. This is done using either the serial SAVE command, or the SynqNet 0x1C Direct Command (see section Direct Commands).
Changes made to parameter values are stored in RAM. It is possible to revert to a saved configuration by explicitly loading the parameters from the EEPROM. This is done using either the serial LOAD command, or the SynqNet 0x1E Direct Command.

The default parameter values can be loaded into RAM by executing either the serial RSTVAR command, or the SynqNet 0x1D Direct Command.

The EEPROM may be cleared using either the serial CLREEPROM command, or the SynqNet 0x1F Direct Command.

The following diagram illustrates the relationship between the different type of memory.

![Diagram](image)

**Figure 8-1: Memory Operations**

## 8.5 Indicators

A 7-segment LED on the front panel is used to display the drive status. Four LEDs embedded into the SynqNet™ connectors display the SynqNet™ status.
### 8.5.1 Drive Status

Drive Status is indicated using the 7-segment LED that is located on the front panel. This display shows drive status and drive fault codes. In the case that more than one fault exists, fault codes are displayed on the 7-segment LED according to their priority and only one fault code will be displayed. Read the **Fault Status Word** for a complete fault summary.

Most faults (except for Over-Current) are resettable, and do not require power cycling. When a fault occurs, remove the source of the fault and then execute the Fault Clear instruction. See section on **Clearing Faults**.

The following table shows the display codes, the description and the fault priority.

<table>
<thead>
<tr>
<th>Description</th>
<th>Comments</th>
<th>Fault Display Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal point only</td>
<td>After logic power is applied, the LED will show a decimal point only. The drive is not operational at this point; A SynqNet RESET needs to be executed in order to bring the drive to an operational state.</td>
<td></td>
</tr>
<tr>
<td>Steady ‘2’ only</td>
<td>Torque Mode: the drive is configured and ready to be enabled</td>
<td></td>
</tr>
<tr>
<td>Steady ‘2’ with a decimal point</td>
<td>The decimal point is on when the drive is enabled</td>
<td></td>
</tr>
<tr>
<td>Flashing ‘2’</td>
<td>When using MENCTYPE 4 (WNS encoder initialization), this indicates that the drive is configured and ready to be enabled. The encoder initialization process will begin when the drive is enabled.</td>
<td></td>
</tr>
<tr>
<td>Steady ‘F’</td>
<td>Drive in foldback (current limiting)</td>
<td></td>
</tr>
<tr>
<td>Flashing ‘e’</td>
<td>Flash memory checksum failure (at power up). Need to re-configure the drive’s parameters and SAVE them in the flash memory.</td>
<td>1 (highest priority)</td>
</tr>
<tr>
<td>Flashing ‘P’</td>
<td>Over-current. Results from either a short circuit on the motor power, or by excessive current loop gain. This fault can only be cleared by cycling the power of the drive.</td>
<td>2</td>
</tr>
<tr>
<td>Flashing ‘o’</td>
<td>Over-voltage. Generally caused by regenerative voltage when decelerating the motor. Use a regen resistor to absorb the regen energy.</td>
<td>3</td>
</tr>
<tr>
<td>Flashing ‘t’</td>
<td>Drive over-temperature</td>
<td>4</td>
</tr>
<tr>
<td>Flashing ‘u’</td>
<td>Under-voltage. This fault will appear when the main AC power is not connected. It may also appear during high accelerations. If this is the case, consider programming UVMODE to ride through temporary voltage sags, and UVRECOVER to determine how the drive recovers from an under-voltage fault.</td>
<td>5</td>
</tr>
<tr>
<td>Alternating ‘1’ and minus sign (-)</td>
<td>No comp: The drive is not configured. Load a configuration file and execute the CONFIG instruction.</td>
<td>6</td>
</tr>
<tr>
<td><strong>E</strong></td>
<td>EEPROM fault. This is a hardware failure and the drive must be returned for repair</td>
<td>7</td>
</tr>
<tr>
<td>Alternating ‘A’ and ‘1’</td>
<td>Internal failure of positive analog supply voltage. This is a hardware failure and the drive must be returned for repair</td>
<td>8</td>
</tr>
<tr>
<td>Alternating ‘A’ and ‘2’</td>
<td>Internal failure of negative analog supply voltage. This is a hardware failure and the drive must be returned for repair</td>
<td>9</td>
</tr>
<tr>
<td>Description</td>
<td>Comments</td>
<td>Fault Display</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Alternating ‘c’ and ‘1’</td>
<td>SynqNet communications fault. Check that the SynqNet cables are in place.</td>
<td></td>
</tr>
<tr>
<td>Alternating ‘r’ and ‘4’</td>
<td>Encoder wire break</td>
<td></td>
</tr>
<tr>
<td>Alternating ‘r’ and ‘6’</td>
<td>Illegal Halls. A state of either ’000’ or ’111’ was detected on the Halls signals.</td>
<td></td>
</tr>
<tr>
<td>Alternating ‘r’ and ‘5’</td>
<td>Index line break</td>
<td></td>
</tr>
<tr>
<td>Alternating ‘4’ and minus sign (-)</td>
<td>The commutation initialization process has failed. Read the WNSERR value (parameter 0x2F) to see what caused the failure.</td>
<td></td>
</tr>
<tr>
<td>Alternating ‘r’ and ‘1’ and ‘0’</td>
<td>EnDat communications fault. Check that the EnDat encoder is connected, or check the MENCTYPE parameter to verify that it is correctly set.</td>
<td></td>
</tr>
<tr>
<td>Alternating ‘r’ and ‘8’</td>
<td>A/B out of range. For a sine encoder and a resolver, the drive checks that $\sin^2 + \cos^2 = 1$, within tolerance. This fault indicates that the signal amplitudes are out of tolerance. This fault is not relevant for Encoder feedback.</td>
<td></td>
</tr>
<tr>
<td>Flashing ‘H’</td>
<td>Motor over-temperature. This fault may be triggered if the motor does not contain a temperature-sensing device. If this is the case, set THERMODE to 1, which will tell the drive to ignore this fault.</td>
<td></td>
</tr>
<tr>
<td>Alternating ‘A’ and ‘3’</td>
<td>Internal failure of positive and negative analog supply voltages</td>
<td></td>
</tr>
<tr>
<td>Three horizontal bars</td>
<td>Watchdog: drive firmware failure</td>
<td></td>
</tr>
</tbody>
</table>

### 8.5.2 SynqNet™ Status

Each port has two LEDs. The function of the LEDs is as follows:

<table>
<thead>
<tr>
<th>Component reference</th>
<th>Port</th>
<th>Colour</th>
<th>Meaning when lit</th>
<th>Controlled by</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED1</td>
<td>IN</td>
<td>Green</td>
<td>Link active</td>
<td>PHY</td>
<td>Continuous</td>
</tr>
<tr>
<td>LED2</td>
<td>IN</td>
<td>Yellow</td>
<td>Node state</td>
<td>MAC</td>
<td>Blinking</td>
</tr>
<tr>
<td>LED3</td>
<td>OUT</td>
<td>Green</td>
<td>Link active</td>
<td>PHY</td>
<td>Continuous</td>
</tr>
<tr>
<td>LED4</td>
<td>OUT</td>
<td>Yellow</td>
<td>Repeater ON</td>
<td>MAC</td>
<td>Blinking</td>
</tr>
</tbody>
</table>

### 8.6 Firmware Update

Firmware on the drive is updated using a process called EMBER. Kollmorgen provides a program that downloads the firmware file to the drive over the RS232 serial communications link. At the prototype phase, firmware update is available over the serial port only. At a later stage, it will be available over SynqNet™ as well.
8.6.1 Terminology
- EMBER: Kollmorgen’s terms for the firmware upgrade process
- IGNITE: the name of the utility that you use to upgrade the firmware

8.6.2 Important Files
- Ember.a00: The name of the file that manages the firmware upgrade and programs the DSP
- cdSynq_vvv.i00: The firmware file. The file extension has no relevance to the firmware version. The version is indicated in the 3 letter/digit suffix of the file name.

8.6.3 Preparations

8.6.3.1 Retrieve Drive Parameters
It is recommended to retrieve and store the drive parameters before upgrading the firmware. New versions of firmware may have different sets of parameters; the drive verifies the checksum of the parameters and if a checksum error is found then the parameters are not loaded. Thus, if the new version has a different parameter set, the checksum will fail when the drive is powered up and the parameter settings will be lost.

- Using MotionLink
  1. Go the Drive Backup screen

![MotionLink Main Screen](image)
2. Click on the Receive button to get the parameters from the drive.
3. Click on the Save button to save the file to disk.

- **Using SynqNet Direct Commands**
  1. Open a DOS PROMPT window in `C:\MEI\XMP\Bin\`
  2. Use the `CDDriveConfig.exe` utility to retrieve the drive parameters to a file on your disk. For example:
     A. Open a DOS PROMPT window in `C:\MEI\XMP\Bin\`
     B. Execute the following command:
        `CDDriveConfig -upload -file MyParams.txt`

- **8.6.3.2 Clear the Drive Parameters**

  - **Using MotionLink**
    1. Execute a CLREEPROM instruction from the command prompt

  - **Using SynqNet Direct Commands**
    1. Open a DOS PROMPT window in `C:\MEI\XMP\Bin\`
    2. Use the `synqCmd.exe` utility to clear the drive non-volatile parameters memory. This is done by executing Direct Command 0x1F, as follows:
       `synqCmd -flags 0x003d0000 -blockNum 0 -addr 0x1F`
    3. Power off
8.6.3.3 Put the CD SynqNet Unit into Ember mode
When updating firmware from versions up to and including 0.1.2, the CD SynqNet has to be out into Ember mode using a **hardware** mechanism (jumpers). For versions from 0.1.3 and up, the hardware mechanism may be used, but a **software** mechanism is available that obviates the need to disassemble the drive in order to assemble the jumpers.

**Hardware Mechanism**

1. Close any program (such as MotionLink) that may be using the serial port of the PC.
2. Remove all power wiring.
3. Remove the cover. There are 3 flat-head Phillips screws on each of the top side and bottom side, and 2 screws on the back side of the CD.
4. Identify jumper JP101. The following diagram shows the layout of the bottom half of the PCB. JP101 is adjacent to the 13-pin front-panel I/O connector.

![Diagram of the PCB layout](image)

5. Connect a 2-pin jumper between pins 1 and 2, and a second 2-pin jumper between pins 3 and 4. The presence of these jumpers causes the drive to power up in Ember mode.
6. Apply 24V logic power to the drive. On power up the 7-segment LED will be blank, but the fan will work.
7. Execute a SynqNet reset command. This is necessary since the drive processor is held in a reset state by the SynqNet FPGA, and thus the processor is inactive until this reset is done.
   A. Open a DOS PROMPT window in C:\ MEI\XMP\Bin\
   B. Type RESET at the command prompt

---

**Figure 8-4: Location of Jumper for Ember**
An error message is expected, as shown in the picture below. The error message appears because the RESET command expects a response from the drive, but receives none since the drive is in Ember mode, and not in its operational mode.

![RESET Error Message](image)

**Figure 8-5: RESET Error Message**

### Software Mechanism
- No specific actions are required at this stage.

### 8.6.4 Update Drive Firmware
Use the CD SynqNet **ignite28_V305.exe** program to download the new firmware. This is a Windows program. Run the program; the following screen appears:

![Ignite28xx Main Screen](image)

**Figure 8-6: Ignite28xx Main Screen**
8.6.4.1 Communications Settings
1. Select the correct COM port
2. Select the correct baud rate.
   - For the Hardware Mechanism, The Ember process will work at up to 38400bps.
   - For the Software Mechanism, the baud rate must be set to 9600.

8.6.4.2 Select Files
1. Click on the Advanced button. A display similar to the following appears:

```
Advanced options

Files to download:

<table>
<thead>
<tr>
<th>File Name</th>
<th>File Path</th>
</tr>
</thead>
</table>
| csynq_012.i00 | C:\Documents and Settings\h...

Ember program path:
C:\Documents and Settings\hurwitz\My Documents\Briefcase\ENC

[Add] [Delete] [Browse] [OK] [Cancel]
```
2. Click on the `Browse` button to search for the `ember_ver5.a00` file.

3. Select the `ember_ver5.a00` file and click on the `Open` button. This file contains the code that programs the firmware and manages the download. The IGNITE program downloads this file to the DSP first.

4. Files to Download:
   A. Click on all files that are listed and DELETE them (one at a time).
   B. Click on the `Add` button, to select the file to be downloaded.
   C. Select the path to the `cdsynq_xyz.i00` file. This file contains the drive firmware.
   D. Click on Open to return to the Advanced Options screen.
   E. Click on OK to return to the main Ignite screen.

**8.6.4.3 Start Firmware Update**

1. Click on the Start button to start the firmware download. The 7-segment LED display will show an E.
2. When the process is complete, click on the Exit button. The 7-segment LED display will show an alternating

![E]

and reversed ‘E’

The firmware has been upgraded.

8.6.5 Resuming Operation

8.6.5.1 Return Drive to Operational State

If the HARDWARE mechanism was used to put the drive into the EMBER mode:

1. Power down the drive
2. Remove all power wiring
3. Remove the jumpers from JP101
4. Assemble the cover (for safety reasons)
5. Connect the wiring and power up
6. Execute a SynqNet RESET command. This is necessary since the drive processor is held in a reset state by the SynqNet FPGA, and thus the processor is inactive until this reset is done. This can be done through MotionConsole, by clicking on the RESET button.

If the SOFTWARE mechanism (EMBER instruction) was used:

1. Execute a SynqNet RESET command.

8.6.5.2 Restore Drive RESET Parameters

1. Read the drive firmware version (VER instruction) to verify that the new firmware has indeed been loaded.
2. Use either MotionLink or the SynqNet drive configuration utility to download the original drive parameters.
3. Set any parameters that may have been added to the new version.
4. Execute the 0x1C Direct Command to save the parameters to the non-volatile parameter memory.

   `synqCmd –flags 0x003d0000 –blockNum 0 –addr 0x1C`

8.7 Absolute Position

The drive calculates and manages absolute position in units of equivalent encoder counts. The value is stored in a 32-bit signed value called PFB. In sine encoder and resolver drives, this value is communicated from the drive to the SynqNet FPGA, and is reflected directly in the position value read by the motion controller. In encoder drives, the motion controller may receive the position in this way, or it may receive it directly from the SynqNet FPGA itself (since this device has a quadrature encoder interface).

The drive uses the value of PFB to manage commutation and to manage the Divide-by-N process.
8.8 Commutation Initialization without Halls

8.8.1 Overview
In brushless 3-phase motors the torque generated by the motor depends on the three phase currents and on the motor position. In order to properly control the motor and extract maximum torque out of the available motor and power stage, the controller must be provided with the current motor position. In encoder-based systems, the initial position for commutation initialization may be found using hall sensors switching, index finding, or an appropriate commutation-lock algorithm (or a combination of the three methods).

Encoder initialization without motion uses a commutation-lock algorithm.

Since, on power up, the absolute motor position is unknown, two methods may be implemented:

· Applying a forced commutation to move the motor to a predetermined position where the torque generated is zero and updating controller variables accordingly.
· Using a motion control algorithm that will "bring" the commutation angle of the motor from the initial, unknown position to the current motor position (instead of moving the motor).

The disadvantage of the first method is that it requires the motor to be moved, which produces a "jumpy" motion that may not be tolerable in some cases (for example - linear motor applications).

The second method is designed to solve this problem by implementing a closed loop commutation-lock algorithm that adjusts the commutation angle to the motor position rather than moving the motor to a predetermined place. The motor will move very slightly; motion of about ±4 electrical degrees is expected.

When an incremental encoder feedback device is used, commutation is initialized either through the use of hall effect devices, or through the use of the “Wake-No-Shake” (WNS) mechanism. WNS is a capability that allows commutation to be initialized without requiring hall effect sensors in the encoder.

8.8.2 The Process
1. Disable the drive
2. Select the encoder initialization process by setting MENCTYPE to 4. This will automatically set the INITMODE parameter to 2. The INITMODE parameter itself does need to be accessed.
3. Set the encoder initialization current using the IENCSTART instruction. Set this to the maximum allowed application current.
4. Set the gain for the process using the INITGAIN instruction. This gain is typically set to 1000.
5. Set the MJ parameter to the value of the motor’s inertia.
6. Enable the process. After power up and after a feedback loss fault, the process will be enabled automatically if MENCTYPE is set to 4. In addition, the process can be enabled at any time when the drive is disabled by entering the ENCSTART command (Direct Command 0x34).
7. Begin the process by enabling the drive.
8. Monitor the process by reading the status word WNSERR and the status of ACTIVE
9. If the process completes successfully, the drive will be enabled, and ACTIVE will be set to 1. If the process is not successful, bits in the WNSERR query give information:
10. If the process completes successfully, the drive will be enabled, and ACTIVE will be set to 1. If the process is not successful, bits in the WNSERR query give information:

<table>
<thead>
<tr>
<th>Bit Value</th>
<th>Error Description</th>
<th>Possible Corrective Action</th>
</tr>
</thead>
</table>

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com
<table>
<thead>
<tr>
<th>Bit Value</th>
<th>Error Description</th>
<th>Possible Corrective Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0001</td>
<td>WNS Stopped</td>
<td>Indicates whether the WNS process was interrupted due to drive disable (due to fault or disable command).</td>
</tr>
<tr>
<td>0x0002</td>
<td>Maximum velocity error</td>
<td>At the end of WNS process the motor should stand still. If the velocity at that time is above threshold, this bit is set.</td>
</tr>
<tr>
<td>0x0004</td>
<td>Too much motion</td>
<td>The motor moved distance which is above threshold during the WNS process.</td>
</tr>
<tr>
<td>0x0008</td>
<td>Motion Profile</td>
<td>If the settling time of the WNS process (step response) is above threshold this bit is set.</td>
</tr>
<tr>
<td>0x0010</td>
<td>Too little motion</td>
<td>Minimum movement of 4.5 electrical degrees is required. This prevents cases of locked motor or not enough current to move the motor.</td>
</tr>
<tr>
<td>0x0020</td>
<td>Encoder initialization failed</td>
<td>This bit is a summary bit of bits 0 to 5.</td>
</tr>
<tr>
<td>0x0040</td>
<td>Encoder initialization has been executed (but not necessarily succeeded)</td>
<td>This bit indicates that a request for WNS was issued. Before the WNS started, all error bits are 0 (indicate no fault).</td>
</tr>
</tbody>
</table>

In case bit 2, 3, or 4 were set, the user should modify the WNS gains (INITGAIN, IENCSTART)

## 8.9 Commutation Initialization with Halls

The following procedure allows the user to configure drive parameters such that commutation will be correct. The procedure is valid when the feedback type is Incremental Encoder with Halls (MENCTYPE=0). The drive parameters used in this procedure are MPHASE, MFBDIR and MHINVx.

In addition, the HALLS instruction is used to read the value of the halls sensor states. The HALLS instruction is best executed from within a MotionLink terminal. The drive returns a 3 digit value, with each digit being ‘1’ or ‘0’ and representing a hall state. The left-hand digit represents hall sensor C, the middle digit represents hall sensor B, and the right-hand digit represents hall sensor A. For example the halls sensor pattern

011

indicates that

- Hall C is 0
- Hall B is 1
- Hall A is 1
This section of the document makes use of a value of the halls sequence, with the value being a decimal representation of the binary halls states. In the above example, where the HALLS instruction returns the value ‘011’, the equivalent halls sequence value is 3. Similarly, a HALLS pattern of ‘110’ is equivalent to a halls sequence value of 6.

Assumption: wiring predefined. Once the motor and feedback have been wired to the drive, no change in wiring is allowed.

8.9.1 Reset Parameters
- Set MPHASE=0, MFBDIR=0, MHINVx=0.

8.9.2 Identify the direction of motor phases.
- Disconnect motor leads from drive.
- Use lab power supply with current limit capability.
- Apply current (the amount of current is the minimum that still locks the motor firmly) from motor phase C (connected to the positive terminal of the power supply) to motor phase B (connected to the negative terminal of the power supply).
- Make sure the motor is locked in position.
- Apply the same current from motor phase C to motor phase A. The motor should jump 60 electrical degrees.
- Watch the direction of the motor jump.
- Let define this direction as “positive motor phase direction”.

8.9.3 Set MFBDIR bit 2.
- Rotate the motor manually slowly to the positive motor phase direction, and monitor the hall state using HALLS command (from MotionLink Terminal).
- Positive hall sequence is {1,5,4,6,2,3}. Check only the sequence, not the start point.
- If the hall sequence is backwards, set MFBDIR bit 2 to 1.
- Please pay attention that setting this bit doesn’t change the readout of HALLS command.

8.9.4 Align the halls to motor phase (set MPHASE).
- Apply current from motor phase C to motor phase B (same way as in item 2).
- Read hall state.
- Try to manually move a bit (right and left) the motor from where it locked, while reading the hall state, to see if the motor is close to hall edge.
- When current is applied from motor phase C to motor phase B the motor should be locked between hall states 1 and 3.
- Calculate the locking location angle according to the following table:
<table>
<thead>
<tr>
<th>Hall edge</th>
<th>MFBDIR bit 2 = 0</th>
<th>MFBDIR bit 2 = 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 and 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1 and 5</td>
<td>60</td>
<td>300</td>
</tr>
<tr>
<td>5 and 4</td>
<td>120</td>
<td>240</td>
</tr>
<tr>
<td>4 and 6</td>
<td>180</td>
<td>180</td>
</tr>
<tr>
<td>6 and 2</td>
<td>240</td>
<td>120</td>
</tr>
<tr>
<td>2 and 3</td>
<td>300</td>
<td>60</td>
</tr>
</tbody>
</table>

- Set MPHASE = 360 – Angle.
- If machine positive direction is opposite to motor phase direction, add (or subtract) 180 from MPHASE.

8.9.5 Set MFBDIR bit 0.
- Rotate the motor manually one rev (or one pitch). The direction is not relevant.
- Rotate the motor manually slowly to the positive motor phase direction, and monitor PRD.
- If PRD counts down, set MFBDIR bit 0 to 1.

8.9.6 Set MFBDIR bit 1.
- Rotate the motor manually slowly to machine positive direction, and monitor PFB.
- If PFB counts down, set MFBDIR bit 1 to 1.

8.9.7 Operate the system.
- **Save Settings**
  At this point, everything should be working properly. Save the settings to the CD's EEPROM (using the serial SAVE command or the SynqNet 0x1C Direct Command) and to disk.

- **Confirm Proper Commutation at All Initialization Conditions**
  Disable the drive. Push the drive by hand until the HALL state is 001. Turn off power to the drive and wait for the LED display to go blank, and then wait 5 seconds more. Turn on power to the drive. Slowly increase positive DAC input to the drive until motion just begins in the positive direction. Slowly decrease negative DAC input to the drive until motion just begins in the negative direction. The positive and negative DAC values should be approximately the same (assuming the motor is level). Repeat this process by starting from each of the six Hall states.

- **Confirm Proper Operation in the Application Conditions**
  Tune the servo loop. Command aggressive moves and the highest acceleration used by the application. Observe the peak value of DAC input required for forward and reverse motion. These values should be approximately the same (within about 10%). If desired/required, adjust MPHASE up or down to give equal peak DAC output in both directions.
8.10 Considerations for Working with EnDat Sine Encoders

8.10.1 Setting the Encoder Type
The MENCTYPE parameter is used to tell the drive with which type of encoder it is working. Set MENCTPE to the value 9 when working with EnDat encoder.

8.10.2 Equivalent Counts per Revolution
The equivalent number of counts per revolution is calculated from

\[ \text{MENCRES} \times \text{MSININT} \times 4 \]

Where MENCRES is the encoder resolution (in lines per rev) and MSININT is the interpolation level.

8.10.3 Hardware Absolute Position
The absolute position is stored in the EnDat sine encoder. This value is read by the drive at power up and after a feedback loss fault, and is used, together with the analog information from the sine and cosine signals, to calculate the 32-bit absolute position. This value may be read at any time using the HWPOS Direct Command; reading the HWPOS does not affect the absolute position value (PFB) as managed by the drive.

8.10.4 Absolute Position Mode
The absolute position is read by the drive from the encoder at power up. This information can be read in either signed or unsigned format, and this affects the way the users sees this absolute position. The ABSPOSMOD drive parameter is used to determine whether the absolute position read on power up is signed or unsigned.

To illustrate this, let us assume that the encoder has a single turn, the encoder resolution is 2048 and the interpolation level is 256. Thus, the equivalent number of encoder counts per revolution is 2,097,152.

- If the ABSPOSMOD parameter is set to 0 (unsigned format), then the absolute position on power up will be in the range 0 through 2,097,151.
- If the ABSPOSMOD parameter is set to 1 (signed format), then the absolute position on power up will be in the range –1,048,576 through 1,048,575.

8.10.5 Position Feedback Offset
A position feedback offset can be set in order to change the value of the absolute position as seen by the user. This is done using the PFBOFF parameter. This parameter is a feedback offset that is added to the internal cumulative position counter to give the value of PFB. It can be used, for example, to set the absolute position read by the user to zero. In order to do this, follow the following steps:

- Set PFBOFF to zero
- Read PFB
- Set PFBOFF to the negative value of PFB

The value of PFBOFF is stored in the EnDat encoder EEPROM when the HSAVE Command is executed. At power up the value is read from the EnDat EEPROM, and PFB is automatically compensated.
8.10.6 Sine/Cosine Calibration

For maximum position accuracy, the amplitudes of the sine and cosine signals need to be well matched. The Heidenhain encoder specification shows that these signals may have amplitudes independently in the range of 0.8V to 1.2V. The CD SynqNet provides the ability to calibrate the sine and cosine signals. Refer to the section on Sine/Cosine Calibration.

8.11 Sine/Cosine Calibration

8.11.1 Overview

The software Sine Encoder and software Resolver algorithms are based on sampling the incoming sine and cosine signals. Although the process is transparent to the user, and therefore does not require additional commands, the accuracy of the process depends on the sampling accuracy of the sine and cosine values. In order to prevent accuracy degradation due to electronic component tolerances, the sine and cosine values must be gain- and offset-compensated.

The process of finding the gain and offset compensation parameters matches an amplifier to an encoder or resolver, and hence must be done after replacing a motor or an amplifier. After the process terminates the gain and offset values are stored in the non-volatile memory and are loaded each time the amplifier is powered on.

The process includes finding 128 maximum and minimum, Sine and Cosine peaks and calculating the average gain and offset values. Due to accuracy restrictions the motor must be rotated at a slow speed so that the Sine/Cosine waves generated will be at a frequency low enough for a valid result. The speed must be such that the frequency of the Sine/Cosine signals does not exceed 250Hz. For a rotary motor, the maximum motor speed in RPM can be calculated from

\[
60 \times 250 / (\text{MENCRES} \times \text{MSININT})
\]

For a linear motor, the maximum motor speed in mm/sec can be calculated form

\[
250 \times \text{MPITCH} / (\text{MSININT} \times \text{MENCRES})
\]

- \text{MENCRES} is the encoder resolution (SynqNet parameter 0x7).
- \text{MSININT} is the sine encoder interpolation level (SynqNet parameter 0x41).
- \text{MPITCH} is the linear motor pitch (SynqNet parameter 0x23).

8.11.2 The Process

During calibration the motor can be moved manually or under servo control (preferably under velocity control). The following steps should be taken:

- Initialize the process by entering the instruction SININIT. This is SynqNet Direct Command (0x37). When the process is initialized, the SININITST parameter (SynqNet parameter 0x3B) is set to 1, to indicate that the process is running.
- Move the motor in either direction.
- While moving the motor, query the status using the SININITST parameter.
- The process is complete when SININITST returns a value of 0 (procedure not running).

Once the process has been completed, the sine and cosine offset and gain values are stored automatically in the drive’s non-volatile memory (EEPROM). These values may be read, but they do not have physical units. Refer to the SINPARAMx parameters (SynqNet parameters 0x3C to 0x3F).
8.12 Drive Enable
The drive is enabled by a combination of 4 signals or states:

- Remote Enable. The state of this signal can be checked using the `REMOTE` parameter.
- SynqNet™ Enable, generated from the SynqNet FPGA.
- The Software Enable is generated by one of the SynqNet™ instructions/mechanisms.
- The drive can be enabled only when no faults exist.

By default, the Software Enable is disabled at power up.

8.13 Operating Modes
The CD SynqNet operates in torque mode only. This is indicated by a ‘2’ on the 7-segment LED display during normal operation.

8.14 Divide-by-N

8.14.1 Synchronizing the Position Feedback
Absolute position of the system is determined by the motion controller during the Homing process. Since the drive needs to generate the DBN pulses in relation to the absolute position, the drive needs to be told what the absolute position is.

Synchronize the drive to the system’s absolute position by writing the absolute position to the drive’s PFB (position feedback) parameter. PFB can only be written to when the motor is stopped. This operation needs to be done once only during continuous machine operation. It needs to be done after every Homing procedure.

8.14.2 DBN Setup
1. Set the number of counts for each pulse, using the DIVN parameter. When this parameter is set to zero, the DBN function is disabled; setting the parameter to a non-zero value enables the DBN function.
2. Set the position at which to start generating pulses using the DIVNSTART parameter.
3. Set the position at which to stop generating pulses using the DIVNSTOP parameter.
4. Set the direction of operation using the DIVNDIR parameter. The first pulse is always aligned with the DIVNSTART position, even if motion is in the negative direction.
5. Execute the DIVNCFG instruction to initialize the DBN algorithm. For greatest accuracy, the motor should be stopped when this instruction is executed.

8.14.3 Effect of Direction
There is no limitation on the combination of Direction, Start and Stop parameters. However, the first pulse is always aligned with the DIVNSTART position, even if motion is in the negative direction.
8.14.4 Calculating the DBN Position Error

The DBN update accuracy depends on the accuracy of the PFB at the time the DBN is configured. If the PFB that is used for DBN calculation does not equal to the actual PFB, an error is created. The error is actually the difference between the PFB that is used for DBN calculation and the actual PFB. A difference is these values is created when DBN in configured when the motor is not at rest.

Since the PFB is updated every 250 µS, the maximum DBN error is

\[
\text{Velocity} \times 250 \mu\text{S}
\]

The DBN function can be configured when the motor is not at rest. The allowed maximum velocity is set using the DIVNSPD parameter, which sets the maximum speed for which DIVNCFG will work.

8.15 Homing

Homing is controlled by the motion controller, and is based on the Home input. The homing process is controlled by the motion controller, and the position capture is done by the SynqNet FPGA.

8.16 Faults and Warnings

The CD SynqNet™ has a number of different fault codes. Faults and warnings are indicated in a 32-bit Fault Status word\(^1\), with each bit indicating a specific fault or warning. The drive is disabled when a fault occurs, and will be re-enabled when the fault condition is removed and the fault state is cleared.

8.16.1 Warning

The drive has one warning only, and this is the Foldback warning. The drive monitors the average current using an I\(^2\)t algorithm. When the rated RMS continuous current is exceeded a warning is issued; the current folds back but the drive will not be automatically disabled when foldback occurs.

Bit 14 of the Cyclic Status Flags indicates a warning. Since the Foldback is currently the only warning issued, bit 14 will be set to indicate a foldback condition.

8.16.2 Fault Status Word

When a fault occurs, a bit in the Fault Status word is latched and bit 15 in the SynqNet™ Cyclic Status Flags is set to indicate this state. The fault status word has 32 bits. Bit 14 in the lower 16 bits indicates that a feedback loss fault has occurred, and the upper 16 bits are used to identify the specific faults associated with a feedback loss.

The following table describes the bits in the Fault Status word. The fault status word is read using Direct Command 0x08.

<table>
<thead>
<tr>
<th>Bit mask of Faults Status word</th>
<th>Fault Description</th>
<th>Possible Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0001</td>
<td>EEPROM checksum fail</td>
<td>EEPROM checksum invalid on power up. Set all drive parameters and save them in the EEPROM.</td>
</tr>
</tbody>
</table>

\(^1\) Firmware version 0.1.1 has a single fault word; later versions may have multiple words
### Bit mask of Faults Status word

<table>
<thead>
<tr>
<th>Bit mask</th>
<th>Description</th>
<th>Possible Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0002</td>
<td>Over current</td>
<td>Power stage surge current. Can be caused by</td>
</tr>
<tr>
<td></td>
<td>The over current fault can only be cleared by either a SynqNet RESET or a power cycle.</td>
<td>• Short circuit of motor power leads</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Excessive current loop gain (try reducing MLMIN or increasing VBUS)</td>
</tr>
<tr>
<td>0x0004</td>
<td>Over voltage</td>
<td>Excessive deceleration rate, resulting in increased bus voltage due to regeneration</td>
</tr>
<tr>
<td>0x0008</td>
<td>Unused</td>
<td></td>
</tr>
<tr>
<td>0x0010</td>
<td>Drive over temperature</td>
<td>The temperature on the heat sink has exceeded 80°C.</td>
</tr>
<tr>
<td>0x0020</td>
<td>Under voltage</td>
<td>Bus voltage is too low. Check that AC power is still applied.</td>
</tr>
<tr>
<td>0x0040</td>
<td>Not configured</td>
<td>Invalid motor data or control loops not initialized</td>
</tr>
<tr>
<td>0x0080</td>
<td>Unused</td>
<td></td>
</tr>
<tr>
<td>0x0100</td>
<td>EEPROM fault</td>
<td></td>
</tr>
<tr>
<td>0x0200</td>
<td>Unused</td>
<td></td>
</tr>
<tr>
<td>0x0400</td>
<td>Positive analog supply fail</td>
<td>Failure in +12V supply</td>
</tr>
<tr>
<td>0x0800</td>
<td>Negative analog supply fail</td>
<td>Failure in -12V supply</td>
</tr>
<tr>
<td>0x1000</td>
<td>SynqNet communication fault</td>
<td>The SynqNet cable has been disconnected. This fault is latched upon receiving a dedicated bit (bit 9) in the downstream cyclic demand flags register. The drive doesn't filter this bit, nor doesn't it wait until the communication (between the SynqNet FPGA and the DSP) stabilizes. It means that even a single appearance of this bit can latch the fault.</td>
</tr>
<tr>
<td>0x2000</td>
<td>Unused</td>
<td></td>
</tr>
<tr>
<td>0x4000</td>
<td>Feedback loss</td>
<td>Some type of feedback loss has occurred. Read the Feedback Loss Status Word to see discover the cause.</td>
</tr>
<tr>
<td>0x8000</td>
<td>Unused</td>
<td></td>
</tr>
</tbody>
</table>

The upper 16 bits indicate faults associated with a feedback loss.

### Bit mask of Feedback Loss Status word

<table>
<thead>
<tr>
<th>Bit mask</th>
<th>Description</th>
<th>Possible Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00010000</td>
<td>A/B Line Break</td>
<td></td>
</tr>
<tr>
<td>0x00020000</td>
<td>Illegal halls</td>
<td>Illegal halls combination detected. The combinations 000 and 111 are invalid. Either the signals are not connected properly, or they have been inverted incorrectly. See <strong>MHINVx</strong> instructions.</td>
</tr>
<tr>
<td>0x00040000</td>
<td>Index break</td>
<td></td>
</tr>
<tr>
<td>0x00080000</td>
<td>Encoder not initialized</td>
<td></td>
</tr>
</tbody>
</table>
### Bit mask of Feedback Loss Status word

<table>
<thead>
<tr>
<th>Bit mask</th>
<th>Fault Description</th>
<th>Possible Cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00100000</td>
<td>EnDat fault</td>
<td></td>
</tr>
<tr>
<td>0x00200000</td>
<td>A/B Out of range</td>
<td></td>
</tr>
</tbody>
</table>
| 0x00400000 | Motor over-temperature fault       | • Motor thermistor leads are not connected (should be connected between pins 13 and 25 of connector C2)  
• Motor has overheated  
• There is no thermistor in the motor. Set THERMODE to 1  
• The thermistor type is not set correctly. Set THERMTYPE to 0 for a PTC device, or 1 for an NTC device |
| 0x00800000 | Sine-quad mismatch                 | This fault is set toggled when the digital counter quad and analog quad mismatch at least 2 consequent sample times and the motor is moving more than 3 counts in digital counter. |
| 0x01000000 | Unused                             |                                                                               |
| 0x02000000 | Unused                             |                                                                               |
| 0x04000000 | Unused                             |                                                                               |
| 0x08000000 | Unused                             |                                                                               |
| 0x10000000 | Unused                             |                                                                               |
| 0x20000000 | Unused                             |                                                                               |
| 0x40000000 | Unused                             |                                                                               |
| 0x80000000 | Unused                             |                                                                               |

### 8.16.3 Reading Faults Over SynqNet

The Fault Status word can be read using Direct Command 0x08 (see section Direct Commands for details on how to use Direct Commands). When using the `synqCmd` utility, the following instruction will cause the Fault Status word to be read:

```
    synqCmd --flags 0x003d0000 --blockNum <node> -addr 0x08
```

where `<node>` is the SynqNet™ node address.

### 8.16.4 Reading Warnings Over SynqNet

The Fault Status word can be read using Direct Command 0x0A. The drive have only one warning, and this is the Foldback warning.

---

1 Fault 0x00800000 appears from firmware version 1.2.1
8.16.5 Clearing Faults
Faults cleared using Direct Command 0x09 (see section Direct Commands for details on how to use Direct Commands). Faults will only be cleared if the fault condition no longer exists. When using the \texttt{synqCmd} utility, the following instruction will cause the faults to be cleared:
\begin{verbatim}
synqCmd --flags 0x003d0000 --blockNum <node> -addr 0x09
\end{verbatim}
where \(<node>\) is the SynqNet™ node address.

\textbf{Notes:}
\begin{itemize}
  \item The Over Current and Watchdog faults can only be cleared by a power cycle.
  \item The Current Foldback bit is a warning only; the drive is not disabled when foldback occurs, and this bit is not latched. It is cleared when the drive is no longer in the foldback state.
\end{itemize}
When working over the serial port, the faults are cleared by toggling the Enable signal. The drive will clear the fault when it detects a transition from Low to High on the Enable signal (not-active to active transition).

8.16.6 Fault History
The drive stores the last 10 faults in a cyclic buffer. Each fault has a time stamp, indicating the time at which the fault occurred. The timer is reset to zero at each power up. The fault history log is accessible only from the serial port, and is read using the FTLHIST instruction.

8.17 Analog Inputs
The CD SynqNet™ drive has 2 external analog inputs, having a range of \(\pm 10\text{V}\). The inputs can be read either via by reading the relevant drive parameter (firmware version 0.1.2 or higher), or by using Monitored Data in the cyclic channel (see \textit{Real Time Monitoring} section below). A third method is by using the \textit{Direct Commands} mechanism.

\textbf{Note:} The preferred method of reading analog inputs is via the drive parameters, and not via Direct Commands.

8.17.1 Drive Parameters
The drive parameters \texttt{ANIN1} and \texttt{ANIN2} are used to read the analog inputs. The values are returned in units of milli-volts. An internal offset can be applied to an analog signal, and this is typically used to zero an analog signal.

8.17.2 Direct Commands
When analog inputs are read via the Direct Command mechanism, the value returned is that read by the drive processor, and is \textbf{not} scaled to reflect the actual input voltage. The value read in this way is actually in the range 0-3V, this being the range of measurement of the drive’s Analog-to-Digital Converter.

8.18 Direct Commands
Direct Commands are used to execute instructions and to access certain drive parameters.

8.18.1 Direct Command Syntax
When using the \texttt{synqCmd} utility, the command syntax is as follows:
synqCmd -flags 0xww3z0000 -blockNum <node> -addr <command code> [-data <data value>]

where
- **ww** is the axis number. For the CD SynqNet, this must always be set to 00
- **z** = F for write or D for read
- **<node>** is the SynqNet™ node; generally 0 for the first drive, 1 for second and so on.
- **<command code>** is the identifier of the Direct Command
- **<data value>** is the data to be written, when accessing a Direct Command that takes data.

### 8.18.2 Table of Direct Command Codes

The direct commands are summarized in the following table. Commands appearing in **Grey** are not implemented.

<table>
<thead>
<tr>
<th>Command Code</th>
<th>Definition</th>
<th>R/W</th>
<th>Pipelining applicable?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>NOP</td>
<td>-</td>
<td>N</td>
<td>Null command</td>
</tr>
<tr>
<td>0x01</td>
<td>Get_Synq_Period / Set_Synq_Period</td>
<td>R/W</td>
<td>N</td>
<td>in units of 40ns</td>
</tr>
<tr>
<td>0x02</td>
<td>Get_Drive_Update_Period / Set_Drive_Update_Period</td>
<td>R/W</td>
<td>N</td>
<td>in units of 40ns</td>
</tr>
<tr>
<td>0x03</td>
<td>Download_Page_Start</td>
<td>W</td>
<td>N</td>
<td>data field selects the download page</td>
</tr>
<tr>
<td>0x04</td>
<td>Download_Data</td>
<td>W</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>0x05</td>
<td>Download_Page_Write</td>
<td>W</td>
<td>N</td>
<td>activates writing to non-volatile memory at the drive</td>
</tr>
<tr>
<td>0x06</td>
<td>Upload_Page_Start</td>
<td>W</td>
<td>N</td>
<td>data field selects the upload page</td>
</tr>
<tr>
<td>0x07</td>
<td>Upload_Data</td>
<td>R</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>0x08</td>
<td>Fault_Read</td>
<td>R</td>
<td>N</td>
<td>Reads code of the existing Fault(s). Refer to section Fault Status Word</td>
</tr>
<tr>
<td>0x09</td>
<td>Fault_Clear</td>
<td>W</td>
<td>N</td>
<td>clears all existing Faults</td>
</tr>
<tr>
<td>0x0A</td>
<td>Warning_Read</td>
<td>R</td>
<td>N</td>
<td>Reads code of the existing Warning(s)</td>
</tr>
<tr>
<td>0x0B</td>
<td>Warning_Clear</td>
<td>W</td>
<td>N</td>
<td>clears all existing Warnings</td>
</tr>
<tr>
<td>0x0C</td>
<td>Warning_Count</td>
<td>R</td>
<td>N</td>
<td>returns how many Warning now exist</td>
</tr>
<tr>
<td>0x0D</td>
<td>Turn_Count_Read</td>
<td>R</td>
<td>N</td>
<td>Reads the number of turns of the absolute encoder</td>
</tr>
<tr>
<td>0x0E</td>
<td>Turn_Count_Clear</td>
<td>W</td>
<td>N</td>
<td>Sets the number of turns of the absolute encoder to zero</td>
</tr>
<tr>
<td>0x0F</td>
<td>Get_Monitor_A_Table / Set_Monitor_A_Table</td>
<td>R/W</td>
<td>N</td>
<td>Using the data passed, Pointer_A is set to one of the tabulated values in tabulated values</td>
</tr>
<tr>
<td>0x10</td>
<td>Get_Monitor_A_Memory/ Set_Monitor_A_Memory</td>
<td>R/W</td>
<td>N</td>
<td>Using the data passed, Pointer_A is set to point to a memory location in the data memory space</td>
</tr>
<tr>
<td>0x11</td>
<td>Get_Monitor_B_Table / Set_Monitor_B_Table</td>
<td>R/W</td>
<td>N</td>
<td>Using the data passed, Pointer_B is set to one of the tabulated values.</td>
</tr>
<tr>
<td>Command Code</td>
<td>Definition</td>
<td>R/W</td>
<td>Pipelining applicable?</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------</td>
<td>-----</td>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x12</td>
<td>Get_Monitor_B_Memory/ Set_Monitor_B_Memory</td>
<td>R/W</td>
<td>N</td>
<td>Using the data passed, Pointer_B is set to point to a memory location in the data memory space</td>
</tr>
<tr>
<td>0x13</td>
<td>Get_Monitor_C_Table / Set_Monitor_C_Table</td>
<td>R/W</td>
<td>N</td>
<td>Using the data passed, Pointer_C is set to one of the tabulated values in tabulated values</td>
</tr>
<tr>
<td>0x14</td>
<td>Get_Monitor_C_Memory/ Set_Monitor_C_Memory</td>
<td>R/W</td>
<td>N</td>
<td>Using the data passed, Pointer_C is set to point to a memory location in the data memory space</td>
</tr>
<tr>
<td>0x15</td>
<td>Get_Char / Put_Char</td>
<td>R/W</td>
<td>N</td>
<td>Gets a character from the virtual serial port buffer or puts a character into the virtual serial port buffer</td>
</tr>
<tr>
<td>0x19</td>
<td>Get_Parameter_Index / Set_Parameter_Index</td>
<td>R/W</td>
<td>N</td>
<td>Returns/Sets-up the parameter pointer to point to the motor’s Nth parameter</td>
</tr>
<tr>
<td>0x1A</td>
<td>Get_Parameter / Set_Parameter</td>
<td>R/W</td>
<td>N</td>
<td>Accesses the value of the parameter pointed to by the parameter pointer</td>
</tr>
<tr>
<td>0x1C</td>
<td>Store_Parameters</td>
<td>W</td>
<td>N</td>
<td>Copies the motor’s parameter table from the Drive Processor’s RAM to its local EEPROM or other local non-volatile memory. Equivalent to the serial communications SAVE instruction.</td>
</tr>
<tr>
<td>0x1D</td>
<td>Restore_Factory_Defaults</td>
<td>W</td>
<td>N</td>
<td>Loads the motor’s parameter table in the Drive Processor’s RAM with a set of factory default parameters. Equivalent to the serial communications RSTVAR instruction.</td>
</tr>
<tr>
<td>0x1E</td>
<td>Reload_Parameters</td>
<td>W</td>
<td>N</td>
<td>Copies the motor’s parameter table in the drive Processor’s local EEPROM to the Drive Processor’s RAM. Equivalent to the serial communications LOAD instruction.</td>
</tr>
<tr>
<td>0x1F</td>
<td>Clear_Parameters</td>
<td>W</td>
<td>N</td>
<td>Clears the motor’s parameter table from the Drive Processor’s local EEPROM or other local non-volatile memory. Equivalent to the serial communications CLREEEPROM instruction.</td>
</tr>
<tr>
<td>0x20</td>
<td>Config_From_Parameters</td>
<td>W</td>
<td>N</td>
<td>Causes the Drive Processor to re-compute the set of internal variables that are derived from the motor’s parameter list that is now in RAM. Equivalent to the serial communications CONFIG instruction.</td>
</tr>
<tr>
<td>0x30</td>
<td>Get_ADC</td>
<td>R</td>
<td>N</td>
<td>Gets the value of an ADC channel implemented at the Drive Processor. Refer to section Accessing Analog Inputs Using Direct Commands</td>
</tr>
</tbody>
</table>

1 The parameter functions provide a general way of accessing drive quantities that are not otherwise accessible by direct commands, for example gains.

2 A local serial EEPROM attached to the DP is typical but the exact implementation of this will vary. Note that this Parameter EEPROM is distinct from the Identification EEPROM.
### 8.18.3 Accessing Analog Inputs Using Direct Commands

Analog inputs can be read using the `Get_ADC` Direct Command. The specific analog input being accessed is specified as shown in the table below.

<table>
<thead>
<tr>
<th>ADC_Channel</th>
<th>Analog Input definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Analog Input 0</td>
</tr>
<tr>
<td>1</td>
<td>Analog Input 1</td>
</tr>
</tbody>
</table>

Note: Accessing an analog input in this way returns the value that is read by the drive processor. The value is not scaled to reflect the actual input voltage.

### 8.18.4 Examples

The following are some examples of commonly used Direct Commands, showing the syntax of the `synqCmd` utility.

**Read Faults:**
```
synqCmd –flags 0x003d0000 –blockNum 0 –addr 0x08
```

**Clear Faults:**
```
synqCmd –flags 0x003f0000 –blockNum 0 –addr 0x09
```

**Store parameters in non-volatile memory:**
```
synqCmd –flags 0x003d0000 –blockNum 0 –addr 0x1C
```
Read value from analog input 1:
   synqCmd –flags 0x003d0000 –blockNum 0 –addr 0x30 –data 0

Read value from analog input 2:
   synqCmd –flags 0x003d0000 –blockNum 0 –addr 0x30 –data 1

Configure Divide-by-N:
   synqCmd –flags 0x003d0000 –blockNum 0 –addr 0x31

Read drive parameter:
   This done in two stages: First the parameter index is set using Direct Command 0x19. For example, in order to read the state of the Remote Enable input (parameter 0x25), the following is done:
      synqCmd –flags 0x003f0000 –blockNum 0 –addr 0x19 –data 0x25
      Next, the actual parameter is read by using Direct Command 0x1A
      synqCmd –flags 0x003d0000 –blockNum 0 –addr 0x1A
   Note how the flags parameter contains “3f” when writing a value and “3d” when reading.

Write Drive Parameter:
   This done in two stages: First the parameter index is set using Direct Command 0x19. For example, in order to set the state of the MPOLES parameter (parameter 0x10) to the value 6, say, the following is done:
      synqCmd –flags 0x003f0000 –blockNum 0 –addr 0x19 –data 0x10
      Next, the actual parameter value is set by using Direct Command 0x1A
      synqCmd –flags 0x003f0000 –blockNum 0 –addr 0x1A –data 6

8.19 Real Time Monitoring
The CD SynqNet supports real-time data monitoring. Monitoring is set up using Direct Commands, and thereafter the data can be either gathered and analyzed by the application, or graphed using MotionScope. Up to three data values can be monitored simultaneously; this data is communicated by the drive to the controller in the Upstream Message. Monitoring can be performed on analog inputs and on any memory location.

8.19.1 Analog Value Monitoring
The following table shows the analog inputs that can be monitored and the index value that needs to be specified.

<table>
<thead>
<tr>
<th>Monitor Index</th>
<th>Data (16-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>U phase current</td>
</tr>
<tr>
<td>1</td>
<td>V phase current</td>
</tr>
<tr>
<td>2</td>
<td>W phase current</td>
</tr>
<tr>
<td>10</td>
<td>Actual torque</td>
</tr>
<tr>
<td>30</td>
<td>Analogue Input 1</td>
</tr>
<tr>
<td>31</td>
<td>Analogue Input 2</td>
</tr>
</tbody>
</table>
8.19.2 Setting up Analog Input Monitoring

Analog input monitoring is set up using Direct Commands 0x0F, 0x11 and 0x13 for monitor channels A, B, and C respectively. The following are examples of how to set up analog input monitoring, showing the syntax of the `synqCmd` utility.

**Analog input 1 on channel A:**
```bash
synqCmd -flags 0x003f0000 -blockNum 0 -addr 0x0F -data 30
```

**Analog input 2 on channel B:**
```bash
synqCmd -flags 0x003f0000 -blockNum 0 -addr 0x11 -data 31
```

**Actual Torque on channel C:**
```bash
synqCmd -flags 0x003f0000 -blockNum 0 -addr 0x13 -data 3
```

8.19.3 Viewing Monitored Data on MotionScope

In order to view monitored data on MotionScope, one needs to know the internal address at which the data appears. Once the address known, a trace can be created in MotionScope.

8.19.3.1 Finding the Monitored Data Address

The address is found using the VM3 utility. Open VM3, and CTRL-PGDN until the `MEIXmpSynqNetData` screen appears. Below is an example.
In this screen the Monitored data appear in two 32-bit words, Monitor[0] and Monitor[1]. Monitor A and Monitor B are in the lower and upper 16 bits respectively of Monitor[0], and Monitor C is in the lower 16 bits of Monitor[1].

8.19.3.2 Defining New Traces in MotionScope

In MotionScope, click on the Traces button. The Select Traces screen appears.
Click on the **New** button to get to a dialog screen in which we will define the monitored data traces. Assuming that the monitored data has been set up such that analog input 1 is being monitored on channel A and analog input 2 is being monitored on channel B, the new traces will be defined as shown in the following screen captures:
Since the monitor data is in a 32-bit word and the data itself is 16-bits wide, the address must be masked. The data for channel A is masked by applying the mask value 0xffff0000. This does 2 things:

- Masks out the upper 16 bits (which belong to monitor channel B)
- Masks out the lower 4 bits of the data (since the analog input has 12-bit resolution).

The data for channel B is masked by applying the mask value 0xfff0. This does 2 things:

- Masks out the lower 16 bits (which belong to monitor channel A)
- Masks out the lower 4 bits (since the analog input has 12-bit resolution).

The analog inputs will now appear as traces on MotionScope.

9. Instruction Set

9.1 Table of CD SynqNet Parameters

The information below describes the parameters of the drive and their conditions of use. Drive parameters may be stored in non-volatile memory, in order to be preserved through power cycles.

9.1.1 NOP

<table>
<thead>
<tr>
<th>Description</th>
<th>No Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Index</td>
<td>0x0</td>
</tr>
<tr>
<td>Firmware Version</td>
<td>0.0.1</td>
</tr>
<tr>
<td>Data Access</td>
<td>N/A</td>
</tr>
<tr>
<td>Data Type</td>
<td>N/A</td>
</tr>
<tr>
<td>Units</td>
<td>N/A</td>
</tr>
<tr>
<td>Range</td>
<td>N/A</td>
</tr>
</tbody>
</table>

9.1.2 MBEMFCOMP

<table>
<thead>
<tr>
<th>Description</th>
<th>Sets the amount of BEMF compensation. This variable affects the amount of back EMF compensation that is applied to the motor command. You can consider the BEMF of the motor as a disturbance to the current loop. The drive has the capability to estimate the amount of BEMF, and to inject feed-forward correction. MBEMFCOMP=0 means no BEMF compensation. MBEMFCOMP=100 means that BEMF compensation equals to the estimated BEMF. Typical values of MBEMFCOMP are 50 to 80.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Index</td>
<td>0x01</td>
</tr>
<tr>
<td>Firmware Version</td>
<td>0.0.1</td>
</tr>
<tr>
<td>Data Access</td>
<td>Read/Write</td>
</tr>
<tr>
<td>Data Type</td>
<td>Integer</td>
</tr>
<tr>
<td>Units</td>
<td>Percent</td>
</tr>
<tr>
<td>Range</td>
<td>0 to 130</td>
</tr>
<tr>
<td>Default</td>
<td>50</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Yes</td>
</tr>
</tbody>
</table>
9.1.3 DICONT

**Description**
Defines the continuous rated current for the drive (sinusoidal RMS). This is a hardware-defined read-only variable that is detected automatically by the drive. In a given application, the drive may be configured to a lower rating than DICONT by setting the value of ICONT to the desired rating.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x02</td>
<td>0.0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Access</th>
<th>Data Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read only</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Units</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Amps RMS * 0.1</td>
<td></td>
</tr>
</tbody>
</table>

9.1.4 DIPEAK

**Description**
Defines the peak rated current of the drive (sinusoidal RMS). This is a hardware-defined read-only variable that will be set to a value of DICONT * 3 on the 3- or 6- Amp units and to the value of DICONT *2 on the 10-Amp unit. DIPEAK sets the 100% reference for many other current variables.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x03</td>
<td>0.0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Access</th>
<th>Data Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read only</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Units</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Amps RMS * 0.1</td>
<td></td>
</tr>
</tbody>
</table>

9.1.5 ICONT

**Description**
Sets the system continuous current. This variable is used in the foldback algorithm. The default value of this variable is the minimum of DICONT (Drive Continuous Current) and MICONT (Motor Continuous Current), unless that value exceeds IMAX, in which case ICONT is set equal to IMAX. This variable is reset to its default whenever DICONT or MICONT is changed. The user can override the default.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x04</td>
<td>0.0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Access</th>
<th>Data Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read/Write</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Units</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>% of DIPEAK * 0.1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>min of DICONT and MICONT</td>
<td>EEPROM</td>
</tr>
</tbody>
</table>

9.1.6 ILIM

**Description**
Sets the application current limit, allowing the user to limit the drive’s peak current. This variable limits the current command issued by the control loops. This variable is an independent variable that is not calculated from hardware parameters and is not tied to any other variables.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x05</td>
<td>0.0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Access</th>
<th>Data Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read/Write</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Units</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>% of DIPEAK * 0.1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Default</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAX</td>
<td>EEPROM</td>
</tr>
</tbody>
</table>
9.1.7 IMAX

Description
Displays the system current maximum for a drive and motor combination. This variable is actually the minimum of the drive Peak Current (DIPEAK) and the Motor Peak Current (MIPEAK).

Parameter Index
0x06

Firmware Version
0.0.1

Data Access
Read only

Data Type
Integer

Units
% of DIPEAK * 0.1

Range
0 to 1000

Default
min of DIPEAK and MIPEAK

9.1.8 MENCRES

Description
Sets the resolution of the motor encoder in number of lines per revolution of the motor (in the case of a rotary motor) or number of lines per motor pitch (in the case of linear motors).

For an incremental encoder, the number of encoder counts per revolution or per pitch is obtained by multiplying MENCRES by 4.

For a sine encoder, the number of encoder counts per revolution is obtained by multiplying MENCRES by MSININT and by 4. The equivalent number of counts per revolution is limited by

MSININT * MENCRES <= 2^{30}

Changing this value puts the drive into a no-comp state, and requires execution of the CONFIG command to release the drive from this state.

Parameter Index
0x07

Firmware Version
0.0.1

Data Access
Read/Write

Data Type
Long Integer

Units
Rotary: Lines per motor revolution
Linear: Lines per motor pitch

Range
100 to 10,000,000

Default
0

EEPROM
Yes
9.1.9 MENCOFF

Description
Sets the encoder index position, and is relevant only for systems that use encoder feedback, and that use the Index mark. This variable is expressed in units of encoder counts after quadrature. It can be set automatically using ENCINIT.

First issue the ENCINIT command. The drive enters find index mode. ENCINITST command will return 1. You have to move the motor manually. When the index is crossed ENCINITST command will return 2. At this point MENCOFF is updated to the index location.

Parameter Index 0x08
Data Access Read/Write
Units Encoder counts/mechanical motor rev
Default 0

Firmware Version 0.0.1
Data Type Long Integer
Range 0 to (4* MENCRES) – 1
EEPROM Yes

9.1.10 MICONT

Description
Sets the motor’s continuous rated current. When this variable is changed, the drive enters a no-comp state, requiring a CONFIG command

Parameter Index 0x09
Data Access Read/Write
Units Amperes RMS * 0.1
Default 0

Firmware Version 0.0.1
Data Type Integer
Range 10 to 1,1750
EEPROM Yes

9.1.11 MIPEAK

Description
Sets the motor’s rated peak current. When this variable is changed, the drive enters a no-comp state, requiring a CONFIG command

Parameter Index 0x0A
Data Access Read/Write
Units Amperes RMS * 0.1
Default 0

Firmware Version 0.0.1
Data Type Integer
Range 10 to 3,500
EEPROM Yes
### 9.1.12 MKT

**Description**
Sets the motor’s torque constant. MKT is part of the Back EMF compensation algorithm.

Motor data sheets often specify the back-EMF constant (MBEMF). The conversion from back-EMF constant to torque constant is done as follows:

**Rotary:**
\[
\text{MKT} = \text{MBEMF} \times 16.55
\]
where MBEMF is in units of V/1000RPM

**Linear:**
\[
\text{MKT} = \text{MBEMF} \times \text{MPITCH} \times 16.55 / (60 \times \sqrt{2})
\]
where MBEMF is in units of V/m/s

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Access</th>
<th>Data Type</th>
<th>Units</th>
<th>Range</th>
<th>Default</th>
<th>EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0B</td>
<td>0.0.1</td>
<td>Read/Write</td>
<td>Integer</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 9.1.13 MLGAINC

**Description**
Sets the current loop adaptive gain value at continuous motor current.

MLGAINC sets the adaptive gain at motor continuous current (MICONT).

MLGAINP sets the adaptive gain at motor peak current (MIPEAK).

Together, with unity gain at zero current, it creates two piece linear curve, where the drive calculates the adaptive gain for a given motor current.

The current-based adaptive gain algorithm is a gain calculation method that increases current loop stability by reducing the current loop gain as the motor current increases (there are motors that their magnetic flux decreases when the current increases, and so their gain increases). A value of 10 (unity gain) is a good starting point.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Access</th>
<th>Data Type</th>
<th>Units</th>
<th>Range</th>
<th>Default</th>
<th>EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0C</td>
<td>0.0.1</td>
<td>Read/Write</td>
<td>Integer</td>
<td>% * 10</td>
<td>1 to 100</td>
<td>8</td>
<td>Yes</td>
</tr>
</tbody>
</table>
9.1.14 MLGAINP

Description
Sets the current loop adaptive gain value at peak motor current.

MLGAINC sets the adaptive gain at motor continuous current (MICONT).
MLGAINP sets the adaptive gain at motor peak current (MIPEAK).
Together, with unity gain at zero current, it creates two piece linear curve, where the drive calculates the adaptive gain for a given motor current.
The current-based adaptive gain algorithm is a gain calculation method that increases current loop stability by reducing the current loop gain as the motor current increases (there are motors that their magnetic flux decreases when the current increases, and so their gain increases). A value of 10 (unity gain) is a good starting point.

Parameter Index: 0x0D
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Integer
Units: % * 10
Range: 1 to 100
Default: 4
EEPROM: Yes

9.1.15 MLMIN

Description
Sets the motor’s minimum line- to- line inductance. This variable is used for current loop controller design and as an input to the Torque Angle Control algorithms. The current loop gain is directly proportional to the value of MLMIN. When this variable is changed, the drive enters a no- comp state, requiring a CONFIG command.

Parameter Index: 0x0E
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Integer
Units: millihenries *10^-2
Range: 1 to 32,767
Default: 0
EEPROM: Yes

9.1.16 MPHASE

Description
Defines the encoder phase relative to the “standard” commutation table.

Parameter Index: 0x0F
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Integer
Units: Electrical degrees
Range: 0 to 359
Default: 0
EEPROM: Yes

For EnDat systems, the MPHASE is saved in the EEPROM of the encoder,
9.1.17 MPOLES

Description
Sets the number of motor poles. This variable is used for commutation control and represents the number of individual magnetic poles of the motor (not pole pairs). When this variable is changed, the drive enters a state, requiring a CONFIG command. When MOTORTYPE= 2, this variable is forced no-comp to a value of 2.

Parameter Index: 0x10
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Integer
Units: Poles
Range: 2 to 80 (even values)
Default: 0
EEPROM: Yes

9.1.18 MSPEED

Description
Defines the maximum recommended velocity of the Motor. When this variable is changed, the drive enters a no-comp state, requiring a CONFIG command.

Parameter Index: 0x11
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Integer
Units: Rotary: RPM
Linear: mm/sec
Range: 6 to 17,464
Default: 0
EEPROM: Yes

9.1.19 MTANGLC

Description
Sets the value of the torque-related commutation angle advance at the motor’s continuous current rating.

MTANGLC sets the phase advance angle at motor continuous current (MICONT) in electrical degrees.

MTANGLP sets the phase advance angle at motor peak current (MIPEAK) in electrical degrees.

Together, with zero angle advance at zero current, it creates two piece linear curve, where the drive calculates the phase advance for a given motor current.

The phase advance helps to achieve higher torque for a given motor current. Usually, it is applicable for buried magnet rotor (as opposed to surface mounted magnet rotor). For a surface mounted magnet rotor the MTANGLx parameters should be set to 0. However, if MVANGLx parameters were not set optimally, non-zero MTANGLx parameters can help to get more torque.

Parameter Index: 0x12
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Integer
Units: Electrical degrees
Range: 0 to 45
Default: 10
EEPROM: Yes
9.1.20 MTANGLP

Description
Sets the value of the torque-related commutation angle advance at the motor’s peak current.

MTANGLC sets the phase advance angle at motor continuous current (MICONT) in electrical degrees.
MTANGLP sets the phase advance angle at motor peak current (MIPEAK) in electrical degrees.

Together, with zero angle advance at zero current, it creates two piece linear curve, where the drive calculates the phase advance for a given motor current.

The phase advance helps to achieve higher torque for a given motor current. Usually, it is applicable for buried magnet rotor (as opposed to surface mounted magnet rotor). For a surface mounted magnet rotor the MTANGLx parameters should be set to 0. However, if MVANGLx parameters were not set optimally, non-zero MTANGLx parameters can help to get more torque.

Parameter Index
0x13

Firmware Version
0.0.1

Data Access
Read/Write

Data Type
Integer

Units
Electrical degrees

Range
0 to 45

Default
23

EEPROM
Yes

9.1.21 MVANGLF

Description
Sets the value of the velocity-rated commutation angle advance to be used when the motor is operating at motor max speed.

MVANGLH sets the phase advance angle at half of motor speed (MSPEED/2) in electrical degrees.
MVANGLF sets the phase advance angle at motor speed (MSPEED) in electrical degrees.

Together, with zero angle advance at zero speed, it creates two piece linear curve, where the drive calculates the phase advance for a given motor speed.

The phase advance helps to achieve higher torque for a given current and a given motor speed. These parameters are independent of the rotor magnets. They come to compensate for computing time and current loop phase lag. In general, they shouldn’t be set to 0.

Parameter Index
0x14

Firmware Version
0.0.1

Data Access
Read/Write

Data Type
Integer

Units
Electrical degrees

Range
0 to 90

Default
0

EEPROM
Yes
9.1.22 MVANGLH

Description
Sets the value of the velocity-rated commutation angle advance to be used when the motor is operating at half of the motor max speed.

MVANGLH sets the phase advance angle at half of motor speed (MSPEED/2) in electrical degrees.

MVANGLF sets the phase advance angle at motor speed (MSPEED) in electrical degrees.

Together, with zero angle advance at zero speed, it creates two piece linear curve, where the drive calculates the phase advance for a given motor speed.

The phase advance helps to achieve higher torque for a given current and a given motor speed. These parameters are independent of the rotor magnets. They come to compensate for computing time and current loop phase lag. In general, they shouldn’t be set to 0.

Parameter Index 0x15
Firmware Version 0.0.1
Data Access Read/Write
Data Type Integer
Units Electrical degrees
Range 0 to 90
Default 0
EEPROM Yes

9.1.23 PWMFRQ

Description
PWM frequency. This value generally set by the drive according to the power stage being used

Parameter Index 0x16
Firmware Version 0.0.1
Data Access Read only
Data Type Integer
Units KHz
Range 8, 16
Default 16 kHz for 3A and 6A drives; 8kHz for 10A drive

9.1.24 VBUS

Description
Sets the drive bus voltage. This variable is used for current controller design. The current loop gain is inversely proportional to the value of VBUS. VBUS also affects the value of VMAX (see VMAX). When this variable is changed, the drive will enter a no-comp state, requiring a CONFIG command.

Parameter Index 0x17
Firmware Version 0.0.1
Data Access Read/Write
Data Type Integer
Units Volts
Range 10 to 850
Default 24
EEPROM Yes
9.1.25 VER

**Description**  
Indicates the version of the drive firmware. The drive firmware consists of 3 digits, for example 1.0.2. Over SynqNet, a 16 bit word is returned, with each digit of the version number in one nibble. For version 1.2.3, the value returned over SynqNet will be 0x0123. The most significant nibble will always be 0 (zero).

**Parameter Index**  
0x18

**Data Access**  
Read only

**Units**  
N/A

**Data Type**  
Long Integer

**Range**  
N/A

---

9.1.26 PFB

**Description**  
Read absolute position from drive, or write value for position initialization.

**Parameter Index**  
0x19

**Data Access**  
Read/Write

**Units**  
Encoder counts

**Data Type**  
Long Integer

**Range**  
-2147483648 to 2147483647

---

9.1.27 DIVN

**Description**  
Divide by N period.

If DIVN equals to 0, the divide by N feature is disabled.

**Parameter Index**  
0x1A

**Data Access**  
Read/Write

**Units**  
Encoder counts

**Data Type**  
Unsigned Integer

**Range**  
0 to 65535

**Default**  
0

EEPROM  
Yes

---

9.1.28 DIVNDIR

**Description**  
Divide by N direction

**Parameter Index**  
0x1B

**Data Access**  
Read/Write

**Units**  
N/A

**Data Type**  
Integer

**Range**  
0 (negative)  
1 (positive)  
2 (Bi-directional pulses)

**Default**  
EEPROM  
Yes

---

9.1.29 DIVNSTART

**Description**  
Divide by N start position

**Parameter Index**  
0x1C

**Data Access**  
Read/Write

**Units**  
Encoder counts

**Data Type**  
Long Integer

**Range**  
-2147483648 to 2147483647

**Default**  
0

EEPROM  
Yes
9.1.30 DIVNSTOP

Description: Divide by N stop position

Parameter Index: 0x1D
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Long Integer
Units: Encoder counts
Range: -2147483648 to 2147483647
Default: 0
EEPROM: Yes

9.1.31 MHINVA

Description: MHINVA is a variable that applies to encoder-based systems that use hall switches to commutate. This variable inverts the hall sensor A feedback, causing the system to read the ‘A’ hall channel as inverted data.

Parameter Index: 0x1E
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Boolean
Units: N/A
Range: 0 (not invert), 1 (invert)
Default: 0
EEPROM: Yes

9.1.32 MHINVB

Description: MHINVB is a variable that applies to encoder-based systems that use hall switches to commutate. This variable inverts the hall sensor B feedback, causing the system to read the ‘B’ hall channel as inverted data.

Parameter Index: 0x1F
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Boolean
Units: N/A
Range: 0 (not invert), 1 (invert)
Default: 0
EEPROM: Yes

9.1.33 MHINVC

Description: MHINVC is a variable that applies to encoder-based systems that use hall switches to commutate. This variable inverts the hall sensor C feedback, causing the system to read the ‘C’ hall channel as inverted data.

Parameter Index: 0x20
Firmware Version: 0.0.1
Data Access: Read/Write
Data Type: Boolean
Units: N/A
Range: 0 (not invert), 1 (invert)
Default: 0
EEPROM: Yes
9.1.34 IN
Description
Returns state of the 3 front-panel digital inputs. The state is represented as a 3-digit number, where each digit represents a different input.
Bit 0 – CW limit input status
Bit 1 – CCW limit input status
Bit 2 – home input status.
For example: The value “100” indicates that the CW limit is engaged, but neither the CCW limit nor the Home switch are engaged.

Parameter Index 0x21
Data Access Read only
Units N/A
Default

9.1.35 NOP
Description
Reserved

Parameter Index 0x22
Data Access Read only
Units N/A
Default

9.1.36 MPITCH
Description
MPITCH is a variable for use with linear motors (MOTORTYPE = 2). It defines the pole pitch (length in millimeters of one electrical cycle - 360 electrical degrees) of the motor and allows the drive to calculate other variables (such as velocity). The drive assumes a 'no-comp' state after an entry of this parameter and requires the CONFIG command.

Parameter Index 0x23
Data Access Read/Write
Units Millimeters per 360 electrical degrees
Default
### 9.1.37 MENCTYPE

**Description**: Sets the motor encoder type. When this variable is changed on an encoder-based system, the drive enters a no-comp state, requiring a CONFIG command.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x24</td>
<td>0.1.0</td>
<td>Integer</td>
<td>0 (A/B/I with Halls)</td>
</tr>
</tbody>
</table>

**Units**: N/A

**Range**: 4 (A/B only; WNS encoder initialization)

**Range**: 6 (A/B with Halls)

**Range**: 9 (EnDat)

**Default**: 0

**EEPROM**: Yes

### 9.1.38 REMOTE

**Description**: Indicates the state of the external hardware enable input line.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x25</td>
<td>0.0.1</td>
<td>Boolean</td>
<td>0 (remote enable input off)</td>
</tr>
</tbody>
</table>

**Units**: N/A

**Range**: 1 (remote enable input on)

### 9.1.39 DIVNSPD

**Description**: Sets the maximum speed in which DIVCFG direct command will work.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x26</td>
<td>0.0.1</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

**Units**: Counts/250 microseconds

**Range**: 2

**Default**: 0

**EEPROM**: Yes

### 9.1.40 MOTORTYPE

**Description**: Sets the drive control algorithms to different motor types. When working with Linear motors, the motor pitch (MPITCH) must also be set.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x27</td>
<td>0.0.1</td>
<td>Integer</td>
<td>0 – rotary</td>
</tr>
</tbody>
</table>

**Units**: N/A

**Range**: 2 – linear

**Default**: 0

**EEPROM**: Yes
9.1.41 VLIM

**Description**
Sets an application maximum velocity. Used in calculated angle advance

**Parameter Index**
0x28

**Data Access**
Read/Write

**Units**
Rotary: RPM
Linear: mm/sec

**Firmware Version**
0.1.0

**Data Type**
Integer

**Range**
10 to VMAX

**Default**
10

**EEPROM**
Yes

9.1.42 MFBDIR

**Description**
Sets the motor feedback direction. MFBDIR is a bit-wise value, with bits 0, 1 and 2 being significant.

**Bit 0** – controls the direction of PRD.
- 0 – Normal, follows the increments of A/B encoder counts.
- 1 – Reversed, follows the negated increments of A/B encoder counts.

**Bit 1** – controls the direction of PFB.
- 0 – Normal, follows the increments of A/B encoder counts. Positive velocity and positive PFB increments for positive increments of A/B encoder counts.
- 1 – Reversed, follows the negated increments of A/B encoder counts. Negative velocity and negative PFB increments for positive increments of A/B encoder counts.

**Bit 2** – controls the inversion of the initial commutation angle, according to the halls.
- 0 – Normal, initial commutation angle according hall state.
- 1 – Reversed, initial commutation angle equals 360 – angle according hall state.

**Parameter Index**
0x29

**Data Access**
Read/Write

**Units**
N/A

**Range**
0, 1, 2, 3, 4, 5, 6, 7

**Default**
0

**EEPROM**
Yes

9.1.43 ANOFF1

**Description**
Sets the analog offset which is added to analog input #1. This is used to compensate for the analog input signal (ANIN1) offset or drift.

**Parameter Index**
0x2A

**Data Access**
Read/Write

**Units**
Milli-volts

**Firmware Version**
0.1.1

**Data Type**
Integer

**Range**
SynqNet: 0 – FFF0
Serial: ±10,000

**Default**
N/A

**EEPROM**
Yes
### 9.1.44 ANOFF2

**Description**
Sets the analog offset which is added to analog input #2. This is used to compensate for the analog input signal (ANIN2) offset or drift.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>0x2B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware Version</td>
<td>0.1.1</td>
</tr>
<tr>
<td>Data Access</td>
<td>Read/Write</td>
</tr>
<tr>
<td>Units</td>
<td>Milli-volts</td>
</tr>
<tr>
<td>Range</td>
<td>SynqNet: 0 – FFF0 Serial: ±10,000</td>
</tr>
<tr>
<td>Default</td>
<td>N/A</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### 9.1.45 ANIN1

**Description**
Displays the analog input value after being filtered by ANOFF1. The value is returned in hex format.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>0x2C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware Version</td>
<td>0.1.1</td>
</tr>
<tr>
<td>Data Access</td>
<td>Read only</td>
</tr>
<tr>
<td>Units</td>
<td>Milli-volts</td>
</tr>
<tr>
<td>Range</td>
<td>±10,000</td>
</tr>
<tr>
<td>Default</td>
<td>N/A</td>
</tr>
</tbody>
</table>

### 9.1.46 ANIN2

**Description**
Displays the analog input value after being filtered by ANOFF2. The value is returned in hex format.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>0x2D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware Version</td>
<td>0.1.1</td>
</tr>
<tr>
<td>Data Access</td>
<td>Read only</td>
</tr>
<tr>
<td>Units</td>
<td>Milli-volts</td>
</tr>
<tr>
<td>Range</td>
<td>±10,000</td>
</tr>
<tr>
<td>Default</td>
<td>N/A</td>
</tr>
</tbody>
</table>
9.1.47 VMAX

Description
Displays the system velocity maximum for a drive and motor combination. This variable is based on drive and motor hardware parameters and is set equal to the MINIMUM of the following values.

For a Rotary motor:
1) MSPEED
2) \((VBUS * 0.707 / MBEMF) * 1000\)
3) 24,000
4) 180,000,000 / MENCRES

where MSPEED is in units of RPM and MENCRES in units of lines per rev

For a Linear motor:
1) MSPEED * 60 / MPITCH
2) \((VBUS * 0.707 / MBEMF) * 1000\)
3) 24,000
4) 180,000,000 / MENCRES * MPITCH / 60

where MSPEED is in units of mm/sec and MENCRES in units of lines per pitch

Parameter Index 0x2E
Firmware Version 0.1.1
Data Access Read only
Data Type Integer
Units
Rotary: RPM
Range 10 to 24,000
Linear: mm/sec
Default 10
EEPROM Yes

9.1.48 WNSERR

Description
This query returns a bit-wise code showing the status of the WNS commutation initialization process. The bits are as follows. Refer to section Commutation Initialization Without Halls above for a complete explanation.

<table>
<thead>
<tr>
<th>Bit Value</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0001</td>
<td>WNS Stopped</td>
</tr>
<tr>
<td>0x0002</td>
<td>Maximum velocity error</td>
</tr>
<tr>
<td>0x0004</td>
<td>Too much motion</td>
</tr>
<tr>
<td>0x0008</td>
<td>Motion Profile</td>
</tr>
<tr>
<td>0x0010</td>
<td>Too little motion</td>
</tr>
<tr>
<td>0x0020</td>
<td>Encoder initialization failed</td>
</tr>
<tr>
<td>0x0040</td>
<td>Encoder initialization has been executed (but not necessarily succeeded)</td>
</tr>
</tbody>
</table>

Parameter Index 0x2F
Firmware Version 0.1.1
Data Access Read only
Data Type Integer
Units N/A
Range See table above
Default 0
9.1.49  INITGAIN
Description Sets the gain for the encoder initialization process controller. Generally, it is set to 1000. Set it to a lower value if too much motion is experienced.

Parameter Index 0x30
Data Access Read/Write
Units N/A
Default 1000

9.1.50  IENCSTART
Description Sets the maximum current for the WNS commutation initialization process.

Parameter Index 0x31
Data Access Read/Write
Units % of MICONT
Default 25

9.1.51  MJ
Description Sets the motor’s rotor inertia (rotary motors) or motor coil mass (linear motors, MOTORTYPE= 2). This parameter is necessary when “Wake-No-Shake” encoder commutation initialization is used.

Parameter Index 0x32
Data Access Read/Write
Units rotary: Kg * m^2 * 10^-6
linear: grams
Default 0

9.1.52  UVMODE
Description Defines how the drive will respond to an under-voltage (UV) fault:
- 0 = latch fault immediately, display flashing “u”.
- 1 = display steady “u”. Warning only, with no fault latch.
- 2 = display steady “u”. After UVTIME elapses, latch fault relay.

If UVMODE= 1 or 2, and the drive is disabled, the UV fault is ignored.

See also UVRECOVER.

Parameter Index 0x33
Data Access Read/Write
Units N/A
Default 0
9.1.53 UVTIME
Description Sets the amount of time an under-voltage warning is displayed ("u")
before it is latched when UVMODE = 2.
Parameter Index 0x34 Firmware Version 0.1.2
Data Access Read/Write Data Type Integer
Units Seconds Range 1 to 300
Default 30

9.1.54 UVRECOVER
Description Defines how the drive will recover from an under-voltage (UV)
fault:
- 0 = recover by executing Clear Faults procedure after the UV
  condition clears
- 1 = automatically recover when the UV condition clears
See also UVMODE.
Parameter Index 0x35 Firmware Version 0.1.2
Data Access Read/Write Data Type Integer
Units N/A Range 0, 1
Default 0

9.1.55 THERM
Description Indicates the state of the motor thermostat input.
- 0 = thermostat input closed (normal).
- 1 = thermostat input open (overheat condition)
Parameter Index 0x36 Firmware Version 0.1.1
Data Access Read only Data Type Integer
Units N/A Range 0, 1
Default Hardware defined

9.1.56 THERMODE
Description Determines the operation of the drive when the Motor Thermostat
Input (THERM) opens.
- 0 = disable drive and open fault relay immediately
- 1 = ignore thermostat input
Parameter Index 0x37 Firmware Version 0.1.1
Data Access Read/Write Data Type Integer
Units N/A Range 0, 1
Default 0
9.1.57 THERMTYPE

Description
Sets the motor temperature sensor type:
- \( 0 = \text{PTC (Positive Temperature Coefficient)} \)
- \( 1 = \text{NTC (Negative Temperature Coefficient)} \)

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x38</td>
<td>0.1.1</td>
</tr>
</tbody>
</table>

Data Access: Read/Write
Data Type: Integer
Units: N/A
Range: 0, 1
Default: 0

9.1.58 IZERO

Description
Sets the C-B phase current for ZERO Mode (A= 0). See ZERO instruction.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x39</td>
<td>0.1.1</td>
</tr>
</tbody>
</table>

Data Access: Read/Write
Data Type: Integer
Units: % of MICONT
Range: 1 to 100
Default: 25

9.1.59 ZERO

Description
Enables and disables feedback Zeroing Mode. If Zeroing Mode is enabled, the drive rotates the motor to an electrical null by placing IZERO current from the motor C terminal to the B terminal.
- \( 0 = \text{zeroing mode disabled} \)
- \( 1 = \text{zeroing mode enabled} \)

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3A</td>
<td>0.1.1</td>
</tr>
</tbody>
</table>

Data Access: Read/Write
Data Type: Integer
Units: N/A
Range: 0, 1
Default: 0
EEPROM: No

9.1.60 SININITST

Description
Queries the status of the sine calibration process. The following values may be returned by the query:
- \( 0 = \text{no request} \)
- \( 1 = \text{process running} \)

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3B</td>
<td>1.0.0</td>
</tr>
</tbody>
</table>

Data Access: Read Only
Data Type: Boolean
Units: N/A
Range: 0, 1
Default: 0
EEPROM: No
### 9.1.61 SINPARAM1

**Description**
Queries the sine signal offset, as read by the ADC.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3C</td>
<td>1.0.0</td>
</tr>
</tbody>
</table>

**Data Access**
Read Only

**Units**
N/A

**Range**
0 to 0xFFF0. The least significant 4 bits are always zero.

**Default**
0

**EEPROM**
Yes

### 9.1.62 SINPARAM2

**Description**
Queries the cosine signal offset, as read by the ADC.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3D</td>
<td>1.0.0</td>
</tr>
</tbody>
</table>

**Data Access**
Read Only

**Units**
N/A

**Range**
0 to 0xFFF0. The least significant 4 bits are always zero.

**Default**
0

**EEPROM**
Yes

### 9.1.63 SINPARAM3

**Description**
Queries the sine to cosine matching gain.

The algorithm requires that the sine and cosine signals should have the same amplitude. The factor used to match them is calculated from \((\text{SINPARAM3} / 2^{\text{SINPARAM4}})\), and represents the amplitude difference of the sine and cosine signals. It should be close to 1. The firmware multiplies the Sine signal samples by this value to get the same amplitude for the Sine and the Cosine signals.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x3E</td>
<td>1.0.0</td>
</tr>
</tbody>
</table>

**Data Access**
Read Only

**Units**
N/A

**Range**
1 to 32767

**Default**
0x4000

**EEPROM**
Yes
9.1.64 SINPARAM4

Description
Queries the sine to cosine matching scale. The algorithm requires that the sine and cosine signals should have the same amplitude. The factor used to match them is calculated from (SINPARAM3 / 2 ^ SINPARAM4), and represents the amplitude difference of the sine and cosine signals. It should be close to 1. The firmware multiplies the Sine signal samples by this value to get the same amplitude for the Sine and the Cosine signals.

Parameter Index 0x3F  
Firmware Version 1.0.0  
Data Access Read Only  
Data Type Integer  
Units N/A  
Range 1 to 15  
Default 14  
EEPROM Yes

9.1.65 ABSPOSMOD

Description
This parameter defines whether the absolute position read at power up from the EnDat encoder will be interpreted as a signed or an unsigned value.

Parameter Index 0x40  
Firmware Version 1.0.0  
Data Access Read/Write  
Data Type Boolean  
Units N/A  
Range 0, 1  
Default 0  
EEPROM Yes

9.1.66 MSININT

Description
This parameter is used with the sine encoder option and sets the interpolation level of the drive. The equivalent number of counts per revolution is calculated from

\[ \text{MSININT} \times \text{MENCRES} \times 4 \]

The equivalent number of counts per revolution is limited by

\[ \text{MSININT} \times \text{MENCRES} \leq 2^{30} \]

NOTE: Although the value can be set up to 1024, the practical upper limit is probably 256.

Parameter Index 0x41  
Firmware Version 1.0.0  
Data Access Read/Write  
Data Type Integer  
Units N/A  
Range 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024  
Default 256  
EEPROM Yes
9.1.67 PFBOFF

Description
This parameter is a feedback offset that is added to the internal cumulative position counter to give the value of PFB. It can be used, for example, to set the absolute position read by the user to zero. In order to do this, follow the following steps:

- Set PFBOFF to zero
- Read PFB
- Set PFBOFF to the negative value of PFB

The value of PFBOFF is stored in the EnDat encoder EEPROM when the HSAVE command is executed. At power up the value is read from the EnDat EEPROM, and PFB is automatically compensated.

Parameter Index 0x42
Data Access Read/Write
Units N/A
Default 0

For EnDat sine encoder systems, the value is initialized at power up by the absolute position read from the encoder.

9.1.68 Feedback Type

Description
This query indicates the type of feedback device that the drive is set up to support.

Parameter Index 0x43
Data Access Read Only
Units N/A
Default Defined by hardware

9.1.69 Halls

Description
Returns the hall switch values (encoder feedback option only). The switch values are displayed as a three-bit code in the sequence C-B-A.

Parameter Index 0x44
Data Access Read Only
Units N/A
Default N/A
9.1.70 ANZERO1

Description
Sets the offset on analog input 1 (ANOFF1) such that reading the analog input would return the value zero. This is an “action”-type instruction; it does not read or write a parameter, but causes a specific action to be taken. Use this parameter as if it were a write-only parameter with a data value of zero.

Parameter Index 0x45
Firmware Version 1.1.0
Data Access Action
Data Type Integer
Units N/A
Range 0
Default N/A
EEPROM No

9.1.71 ANZERO2

Description
Sets the offset on analog input 2 (ANOFF2) such that reading the analog input would return the value zero. This is an “action”-type instruction; it does not read or write a parameter, but causes a specific action to be taken. Use this parameter as if it were a write-only parameter with a data value of zero.

Parameter Index 0x46
Firmware Version 1.1.0
Data Access Action
Data Type Integer
Units N/A
Range 0
Default N/A
EEPROM No

9.1.72 HSAVE

Description
Save the MPHASE and PFOFF parameters to the EnDat EEPROM. This is an “action”-type instruction; it does not read or write a parameter, but causes a specific action to be taken. Use this parameter as if it were a write-only parameter with a data value of zero.

Parameter Index 0x47
Firmware Version 1.1.0
Data Access Action
Data Type Integer
Units N/A
Range 0
Default N/A
EEPROM No
9.1.73 SININIT

**Description**
Initialize the sine/cosine calibration routine. Refer to section Sine/Cosine Calibration for details.

This is an “action”-type instruction; it does not read or write a parameter, but causes a specific action to be taken. Use this parameter as if it were a write-only parameter with a data value of zero.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Access</th>
<th>Data Type</th>
<th>Units</th>
<th>Range</th>
<th>Default</th>
<th>EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x48</td>
<td>1.1.0</td>
<td>Action</td>
<td>Integer</td>
<td>N/A</td>
<td>0</td>
<td>N/A</td>
<td>No</td>
</tr>
</tbody>
</table>

9.1.74 ENCSTART

**Description**
Explicitly put the drive into its Encoder Initialization state. This can be used when MENCTYPE is set to 4, for encoder initialization without Halls.

This is an “action”-type instruction; it does not read or write a parameter, but causes a specific action to be taken. Use this parameter as if it were a write-only parameter with a data value of zero.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Access</th>
<th>Data Type</th>
<th>Units</th>
<th>Range</th>
<th>Default</th>
<th>EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x49</td>
<td>1.1.0</td>
<td>Action</td>
<td>Integer</td>
<td>N/A</td>
<td>0</td>
<td>N/A</td>
<td>No</td>
</tr>
</tbody>
</table>

9.1.75 HWPOS

**Description**
Read encoder absolute position directly from the EnDat encoder. This operation is relevant only for EnDat encoders.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Access</th>
<th>Data Type</th>
<th>Units</th>
<th>Range</th>
<th>Default</th>
<th>EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x4A</td>
<td>1.1.0</td>
<td>Read-only</td>
<td>Integer</td>
<td>N/A</td>
<td>depend on EnDat type</td>
<td>N/A</td>
<td>No</td>
</tr>
</tbody>
</table>

9.1.76 RDRES

**Description**
Sets the resolution of the resolver feedback in resolver systems. The value is in bits, and indicates how many equivalent encoder counts there are per mechanical revolution.

<table>
<thead>
<tr>
<th>Parameter Index</th>
<th>Firmware Version</th>
<th>Data Access</th>
<th>Data Type</th>
<th>Units</th>
<th>Range</th>
<th>Default</th>
<th>EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x4B</td>
<td>1.1.0</td>
<td>Read/Write</td>
<td>Integer</td>
<td>N/A</td>
<td>12 (4096 counts per rev)</td>
<td>14 (16384 counts per rev)</td>
<td>16 (65536 counts per rev)</td>
</tr>
</tbody>
</table>

Default 14
9.1.77 RESBW

Description: Sets the bandwidth of software resolver mechanism. As a general rule of thumb, set RESBW to 4 or 5 times the velocity loop bandwidth.

Parameter Index: 0x4c
Firmware Version: 1.1.1
Data Access: Read/Write
Data Type: Integer
Units: N/A
Range: 200 to 800
Default: 300
EEPROM: Yes

9.1.78 PRD

Description: Sets the absolute position feedback of the hardware feedback device (for both resolver and encoder based systems). PRD will increment from 0 to 65,535 throughout the course of one mechanical motor shaft revolution (360 degrees). The range of PRD will not change. Its resolution for resolver feedback systems is dependent upon the value of RDRES:
- RDRES = 12, resolution of PRD = 16.
- RDRES = 14, resolution of PRD = 4.
- RDRES = 16, resolution of PRD = 1.

For encoder-based systems, until the encoder has been initialized, PRD will be uninitialized and its value will not be useful or meaningful.

Parameter Index: 0x4d
Firmware Version: 1.1.1
Data Access: Read
Data Type: Integer
Units: N/A
Range: 0 to 65,535
Default: N/A
EEPROM: No

9.2 Order of Parameter Download

For some parameters, the order in which they are written is important. This is due to the various interactions between these parameters. The following list shows the parameters for which download order is important, and the order in which they must appear:

- MPITCH
- MOTORTYPE
- MIPEAK
- MICONT
- MSPEED
- MKT
- MENCRES
MENCTYPE
MENCOFF
MLMIN
MPHASE
MPOLES
MBEMFCOMP
MLGAINC
MLGAINP
MTANGLC
MTANGLP
MVANGLF
MVANGLH
MHINVA
MHINVB
MHINVC
VBUS
ILIM
ICONT
VLIM

9.3 Error Codes

The following is a list error codes that may be returned in response to accessing drive parameters.

<table>
<thead>
<tr>
<th>Returned Error value</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x17</td>
<td>Drive is Active</td>
<td>User want to set parameter when drive is enabled and this parameter can be set only when drive is disable.</td>
</tr>
<tr>
<td>0x19</td>
<td>Value out of range</td>
<td>When user want to set the value of parameter that out of it range.</td>
</tr>
<tr>
<td>0x2E</td>
<td>Argument not binary</td>
<td>Usable in MPOLES command when user wants to set odd value.</td>
</tr>
<tr>
<td>0x33</td>
<td>Not Available</td>
<td>Usable in MENCOFF command, when user want to change it and ENCINIT in progress.</td>
</tr>
<tr>
<td>0x2A</td>
<td>EEPROM invalid</td>
<td>When master wants to execute one of the direct commands: Store_Parameters / Reload_Parameters / Clear_Parameters and EEPROM is invalid</td>
</tr>
<tr>
<td>0x2D</td>
<td>EEPROM empty</td>
<td>When master wants to execute Reload_Parameters direct command and EEPROM is empty</td>
</tr>
</tbody>
</table>
### 9.4 Additional Error Codes

Here are additional Error Codes that are returned to the master via Read Data LSW word. The controller can ignore these error codes, because it has Failed Status flag, but it is another option for checking errors.

<table>
<thead>
<tr>
<th>Error Code Number</th>
<th>Description</th>
<th>Possible Cause</th>
<th>Service Command that caused this error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x80</td>
<td>This is not monitor table value</td>
<td>Monitor Table Entry that user want to set to one of Monitored words is not one of Monitored Data tabulated entries (see “Monitored Data – using tabulated values for CD SYNQNET drive” table)</td>
<td>Set_Monitor_A_Table or Set_Monitor_B_Table or Set_Monitor_C_Table</td>
</tr>
<tr>
<td>0x81</td>
<td>Drive Parameter Index out of range</td>
<td>Via “Set Parameter Index” direct command Parameter Index is out of range</td>
<td>Set_Parameter_Index</td>
</tr>
<tr>
<td>0x82</td>
<td>Data or Program Address is out of range</td>
<td>Usable when via memory command given address is out of range</td>
<td>Memory_Space_Select = 00 (DATA) and Address is out of range</td>
</tr>
<tr>
<td>0x83</td>
<td>Synq. period not previously set</td>
<td>Synq period not previously set, so “Get Synq Period” direct command failed</td>
<td>Get_Synq_Period</td>
</tr>
<tr>
<td>0x84</td>
<td>The value of direct command is out of range.</td>
<td>In &quot;Set Drive Update Period&quot; direct command the period specified is not valid for this drive</td>
<td>Set_Drive_Period</td>
</tr>
<tr>
<td>0x85</td>
<td>This parameter is read only</td>
<td>Was attempting to set Parameter Value to parameter that read only.</td>
<td>Set_Parameter</td>
</tr>
</tbody>
</table>

**Note:** All errors except of error 0x82 belong to Direct Commands and error 0x82 belongs to Memory Command.
9.5 SynqNet Cyclic Status Bits

The status flags are set by the drive processor as follows:

<table>
<thead>
<tr>
<th>Bit#</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>2</td>
<td>Reserved</td>
</tr>
<tr>
<td>3</td>
<td>HALL A</td>
</tr>
<tr>
<td>4</td>
<td>HALL B</td>
</tr>
<tr>
<td>5</td>
<td>HALL C</td>
</tr>
<tr>
<td>6</td>
<td>Reserved</td>
</tr>
<tr>
<td>7</td>
<td>Amp Ready</td>
</tr>
<tr>
<td>8</td>
<td>WD toggle</td>
</tr>
<tr>
<td>9</td>
<td>Reserved</td>
</tr>
<tr>
<td>10</td>
<td>Reserved</td>
</tr>
<tr>
<td>11</td>
<td>Amp Powered (Enable)</td>
</tr>
<tr>
<td>12</td>
<td>Drive Ready</td>
</tr>
<tr>
<td>13</td>
<td>Capture</td>
</tr>
<tr>
<td>14</td>
<td>Warning</td>
</tr>
<tr>
<td>15</td>
<td>Fault</td>
</tr>
</tbody>
</table>

The bit interpretation is as follows:

- **Amp Powered**  Drive enabled
- **WD toggle**    Watchdog bit toggled by controller
- **Drive Ready**  Drive has completed synchronization with controller and is able to exchange data cyclically
- **Capture**      Capture event has occurred.
- **Warning**      Warning exists (Foldback is the only warning)
- **Fault**        Drive in fault mode
- **Amp Ready**    Drive ready to be enabled. When AMP_EN is set, the drive will be enabled (ACTIVE).
Artisan Technology Group is your source for quality new and certified-used/pre-owned equipment

• FAST SHIPPING AND DELIVERY
• TENS OF THOUSANDS OF IN-STOCK ITEMS
• EQUIPMENT DEMOS
• HUNDREDS OF MANUFACTURERS SUPPORTED
• LEASING/MONTHLY RENTALS
• ITAR CERTIFIED SECURE ASSET SOLUTIONS

SERVICE CENTER REPAIRS
Experienced engineers and technicians on staff at our full-service, in-house repair center

WE BUY USED EQUIPMENT
Sell your excess, underutilized, and idle used equipment
We also offer credit for buy-backs and trade-ins
www.artisantg.com/WeBuyEquipment

InstraView™ REMOTE INSPECTION
Remotely inspect equipment before purchasing with our interactive website at www.Instraview.com

LOOKING FOR MORE INFORMATION?
Visit us on the web at www.artisantg.com for more information on price quotations, drivers, technical specifications, manuals, and documentation

Contact us: (888) 88-SOURCE | sales@artisantg.com | www.artisantg.com