# Parameter Dictionary

## Installation & Service Instructions

Rev. 02 | PN16-01091 | 3/2019

# CONTENTS

# 1. ABOUT THIS MANUAL

## 1.1 Overview and Scope

This manual provides cross-referenced definitions of the parameters used to program and operate Copley Controls drives.

## 1.2 Related Documentation

CANopen-related documents:

- *CANopen Programmer's Manual*
- *CML Reference Manual*
- *Copley Motion Objects Programmer's Guide*

DeviceNet-related:

- *Copley DeviceNet Programmer's Guide*

Also of related interest:

- *CME User Guide*
- *Copley Indexer 2 Program User Guide (describes use of Indexer 2 Program to create motion control sequences)*
- *Copley ASCII Interface Programmer's Guide (describes how to send ASCII format commands over an RS232 serial bus to control one or more drives)*
- *Copley Camming User Guide (describes the use of the Copley Controls Camming feature, and its setup through CME)*
- Extending Plus Module I/O *application note.*
- *Trigger Outputs at Position application note.*

All these publications, along with hardware manuals and data sheets, can be found on www.copleycontrols.com

## 1.3 Comments

Copley Controls welcomes your comments on this manual. See www.copleycontrols.com for contact information.

## 1.4 Copyrights

## 1.5 Document Validity

We reserve the right to modify our products. The information in this document is subject to change without notice and does not represent a commitment by Copley Controls. Copley Controls assumes no responsibility for any errors that may appear in this document.

## 1.6 Product Warnings

Observe all relevant state, regional, and local safety regulations when installing and using Copley Controls drives. For safety and to assure compliance with documented system data, only Copley Controls should perform repairs to drives.

**DANGER**

**Hazardous voltages.**

Exercise caution when installing and adjusting Copley drives.

**Risk of electric shock.**

On some Copley Controls drives, high-voltage circuits are connected to mains power. Refer to hardware documentation.

**Risk of unexpected motion with non-latched faults.**

After the cause of a non-latched fault is corrected, the drive re-enables the PWM output stage without operator intervention. In this case, motion may re-start unexpectedly. Configure faults as latched unless a specific situation calls for non-latched behavior. When using non-latched faults, be sure to safeguard against unexpected motion.

**Latching an output does not eliminate the risk of unexpected motion with non-latched faults.**

Associating a fault with a latched, custom-configured output does not latch the fault itself. After the cause of a non-latched fault is corrected, the drive re-enables without operator intervention. In this case, motion may re-start unexpectedly.

For more information, see Fault Mask (0xA7).

When operating the drive as a CAN or DeviceNet node, the use of CME or ASCII serial commands may affect operations in progress. Using such commands to initiate motion may cause network operations to suspend.

Operation may restart unexpectedly when the commanded motion is stopped.

**Use equipment as described.**

Operate drives within the specifications provided in the relevant hardware manual or data sheet.

**FAILURE TO HEED THESE WARNINGS CAN CAUSE EQUIPMENT DAMAGE, INJURY, OR DEATH.**

## 1.7 Revision History

| Revision | Date | Comments |
|---|---|---|
| 00 | December 2013 | Added new parameters and fixed existing content. |
| 01 | September 2014 | Fixed units for parameter 0x5e |
| 02 | March 2019 | Added new parameters and fixed the existing content |

# 2 INTRODUCTION

## 2.1 Scope and Purpose of this Document

This document provides a listing and definitions of the parameters used to program and operate Copley Controls drives. These parameters can be accessed using any of several communication interfaces, each with its own protocol and set of IDs for the parameters.

There are many CANopen and EtherCAT objects for which there are no direct correlations to Copley drive parameters. Refer to the *CANopen Programmer's Manual* for a complete list of supported objects.

## 2.2 Organization of the Parameter Listings

The parameters are listed in tables consisting of the following columns:

The **ASCII** (American Standard Code for Information Exchange) column contains the parameter's Copley ASCII Interface parameter ID. This ID would also be used with Copley Controls Indexer 2 Program. The ID is listed in hex format.

The **CAN/ECAT IDX:SUB** column contains the CANopen and EtherCAT object index and sub-index of a parameter. The index is in hex format and the sub-index is in decimal format.

Note that the CANopen and EtherCAT object libraries are identical.

The **MACRO** column contains the parameter's MACRO I-variable ID. The MACRO I-variable ID of a parameter is offset from the ASCII Interface parameter ID by decimal 1024 (hex 0x400).

The **Mem** column indicates whether the parameter is stored in drive RAM (R), drive flash memory (F), or both (RF).
An asterisk next to R in this column indicates that the parameter is read-only. Parameters without an asterisk can be read and written.

The **Type** column indicates the parameter's data type. Types include:

- String: 20 words
- Integer (8, 16, 32, or 64-bit): INT8, INT16, INT32, INT64
- Unsigned (8, 16, 32, or 64-bit): U8, U16, U32, U64)

Cross references for each parameter include, where applicable, the equivalent MACRO I-variable ID, and CANopen (and EtherCAT) object index and sub-index.

The **DvcNet** column has been removed from this revision of the parameter dictionary. DeviceNet ID can be derived from the ASCII ID by adding 1 to it.

Example: ASCII 0x00 = DvcNet 0x01 or ASCII 0x0F = DvcNet 0x10

## 2.3 Important Notes

**CME Refresh Behavior**

When parameters are changed using one of the interfaces described in this manual, the changes will not necessarily be recognized by an active CME session.

**Input/Output Numbering**

Inputs and Outputs on Copley drives are numbered starting from zero for all the communication interfaces listed in this document. If a drive has 12 inputs, they are numbered 0 through 11. CME software starts numbering at 1 (input 0 is called IN1 in CME software).

# 3 PARAMETERS

*** Indicates read only parameters. All others are read and write parameters.**

## 3.1 Parameters Sorted by ASCII Interface Parameter ID

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0x00 | 0x400 | 0x2380:1 | RF | U16 | Current Loop Proportional Gain (Cp). |
| 0x01 | 0x401 | 0x2380:2 | RF | U16 | Current Loop Integral Gain (Ci). |
| 0x02 | 0x402 | 0x2340 | RF | INT16 | Current loop programmed value. Units: 0.01 A.<br>This current will be used to command drive when Desired State (0x24) is set to 1. |
| 0x03 | 0x403 | 0x2203 | R* | INT16 | Winding A Current. Units: 0.01 A.<br>Actual current measured at winding A. |
| 0x04 | 0x404 | 0x2204 | R* | INT16 | Winding B Current. Units: 0.01 A.<br>Actual current measured at winding B. |
| 0x05 | 0x405 | 0x2210 | R* | INT16 | Current Offset A. Units: 0.01 A.<br>Offset value applied to winding A current reading. This offset is calculated by drive at startup. |
| 0x06 | 0x406 | 0x2211 | R* | INT16 | Current Offset B. Units: 0.01 A.<br>Offset value applied to winding B current reading. This offset is calculated by drive at startup. |
| 0x07 | 0x407 | 0x2212 | R* | INT16 | X Axis of calculated stator current vector.<br>Units: 0.01 A. |
| 0x08 | 0x408 | 0x2213 | R* | INT16 | Y Axis of calculated stator current vector.<br>Units: 0.01 A. |
| 0x09 | 0x409 | 0x221A | R* | INT16 | Stator Voltage, X axis. Units: 0.1 V |
| 0x0A | 0x40A | 0x221B | R* | INT16 | Stator Voltage, Y axis. Units: 0.1 V |
| 0x0B | 0x40B | 0x2214 | R* | INT16 | Actual Current, D axis of rotor space. Units: 0.01 A. |
| 0x0C | 0x40C | 0x2215 | R* | INT16 | Actual Current, Q axis of rotor space. Units: 0.01 A. |
| 0x0D | 0x40D | 0x2216 | R* | INT16 | Commanded current, D axis of rotor space. Part of internal current loop calculation. Units: 0.01 A. |
| 0x0E | 0x40E | 0x2217 | R* | INT16 | Commanded Current, Q axis of rotor space. Part of internal current loop calculation. Units: 0.01 A. |
| 0x0F | 0x40F | None | R* | INT16 | Current Error, D axis of rotor space. Units: 0.01 A. |
| 0x10 | 0x410 | None | R* | INT16 | Current Error, Q axis of rotor space. Units: 0.01 A. |
| 0x11 | 0x411 | None | R* | INT16 | Current Integral Value, D axis of rotor space. |
| 0x12 | 0x412 | None | R* | INT16 | Current Integral Value, Q axis of rotor space. |
| 0x13 | 0x413 | 0x2218 | R* | INT16 | Current Loop Output, D axis of rotor space.<br>Units: 0.1 V |
| 0x14 | 0x414 | 0x2219 | R* | INT16 | Current Loop Output, Q axis of rotor space.<br>Units: 0.1 V |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0x15 | 0x415 | 0x221D | R* | INT16 | Commanded Motor Current. Units: 0.01 A.<br>This is value presently being sent to current loop.<br>It may come from programmed value, analog reference, velocity loop, etc. depending on drive's desired state. |
| 0x17 | 0x417 | 0x6064 | R | INT32 | Actual Position. Units: Counts.<br>Used to close position loop in drive every servo cycle. For single feedback systems, this value is same as Actual Motor Position (0x32). For dual feedback systems, this value is same as Load Encoder Position (0x112).<br>CANopen objects 0x6064 and 0x6063 hold same value. |
| 0x18 | 0x418 | 0x6069 | R* | INT32 | Actual Velocity. Units: 0.1 encoder counts/s.<br>For estimated velocity. Units: 0.01 RPM.<br>For stepper mode: Units: 0.1 microsteps/s. |
| 0x19 | 0x419 | 0x2310 | RF | INT32 | Analog reference scaling factor.<br>This value is used to scale analog reference input voltage to command that will be used to drive current, velocity or position loop (depending on drive state).<br>When in current mode (Desired State (0x24) = 2), value programmed specifies commanded current when 10 V is applied to analog input. Units: 0.01 A.<br>For example, to command 12 A at 10 V, scaling factor would be 1200.<br>When in velocity mode (Desired State (0x24) = 12), value programmed specifies commanded velocity when 10 V is applied to analog input.<br>Units: 0.1 encoder counts/s.<br>For estimated velocity. Units: 0.01 RPM.<br>For stepper mode. Units: 0.1 microsteps/s.<br>When in position mode (Desired State (0x24) = (22 or 32), value programmed specifies commanded position (in encoder counts) when 10 V is applied to analog input. |
| 0x1A | 0x41A | 0x2311 | RF | INT16 | Offset Value applied to Analog Input or Analog Reference Input. Units: mV. |
| 0x1B | 0x41B | 0x2205 | R* | INT16 | Analog Encoder Sine Input Voltage. Units: 0.1 mV. Also known as Sine Feedback Voltage. |
| 0x1C | 0x41C | 0x2206 | R* | INT16 | Analog Encoder Cosine Input Voltage. Units: 0.1 mV.<br>Also known as Cosine Feedback Voltage. |
| 0x1D | 0x41D | 0x2200 | R* | INT16 | Analog Input. Units: mV.<br>Also known as Analog Reference Input Voltage. |
| 0x1E | 0x41E | 0x2201 | R* | INT16 | High Voltage A/D Reading. Units: 100 mV.<br>Bus Voltage present on internal high-voltage bus. |
| 0x20 | 0x420 | 0x2202 | R* | INT16 | Drive Temperature A/D Reading.<br>Units: degrees C. |
| 0x21 | 0x421 | 0x2110 | RF | INT16 | Peak Current Limit. Units: 0.01 A.<br>Used by $I^2T$ algorithm to protect motor.<br>Also known as Boost current on stepper drives.<br>This value cannot exceed Drive Peak Current. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| 0x22 | 0x422 | 0x2111 | RF | INT16 | Continuous Current Limit. Units: 0.01 A.<br>Used by $I^2T$ algorithm to protect motor. Also known as Run Current on stepper drives. This value should cannot exceed User Peak Current Limit. | |
| 0x23 | 0x423 | 0x2112 | RF | U16 | Time at Peak Current Limit. Units: ms.<br>Used by $I^2T$ algorithm to protect motor. Also known as Time at Boost Current for stepper drives. | |
| 0x24 | 0x424 | 0x2300 | RF | U16 | Desired State: | |
| | | | | | Value | Description |
| | | | | | 0 | Drive disabled. |
| | | | | | 1 | Programmed current value drives current loop |
| | | | | | 2 | Analog reference drives current loop |
| | | | | | 3 | PWM input drives current loop |
| | | | | | 4 | Function generator drives current loop |
| | | | | | 5 | UV current mode |
| | | | | | 6 | Reserved for future use |
| | | | | | 7 | Current command slaved to other axis |
| | | | | | 8-10 | Reserved for future use |
| | | | | | 11 | Programmed velocity value drives velocity loop |
| | | | | | 12 | Analog reference drives velocity loop |
| | | | | | 13 | PWM input drives velocity loop |
| | | | | | 14 | Function generator drives velocity loop |
| | | | | | 15-16 | Reserved for future use |
| | | | | | 17 | Velocity command saved to other axis |
| | | | | | 18-20 | Reserved for future use |
| | | | | | 21 | Trajectory generator drives position loop |
| | | | | | 22 | Analog reference drives position loop |
| | | | | | 23 | Digital input lines drive position loop (Pulse & direction, master encoder, etc.) |
| | | | | | 24 | Function generator drives position loop |
| | | | | | 25 | Cam tables drive position loop |
| | | | | | 26 | Analog reference commands velocity to position loop |
| | | | | | 27 | Position command slaved to another axis |
| | | | | | 28-29 | Reserved for future use |
| | | | | | 30 | CANopen interface controls drive |
| | | | | | 31 | Trajectory generator drives microstepper |
| | | | | | 32 | Analog reference drives micostepper position |
| | | | | | 33 | Digital input lines drive microstepper |
| | | | | | 34 | Function generator drives microstepper |
| | | | | | 35 | Cam tables drive microstepper |
| | | | | | 36 | Analog reference drives microstepper velocity |
| | | | | | 37-39 | Reserved for future use |
| | | | | | 40 | CANopen interface controls microstepper |
| | | | | | 41 | Reserved for future use |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 42 | Simple micro-stepping mode.<br>For diagnostic use only |
| 0x25 | 0x425 | 0x221E | R* | INT16 | Limited Motor Current Command. Units: 0.01 A. | |
| 0x26 | 0x426 | 0x2313 | RF | INT16 | Analog Reference Input Deadband. Units: mV.<br>Deadband window value applied to analog command input. | |
| 0x27 | 0x427 | 0x2381:1 | RF | U16 | Velocity Loop Proportional Gain (Vp). | |
| 0x28 | 0x428 | 0x2381:2 | RF | U16 | Velocity Loop Integral Gain (Vi). | |
| 0x29 | 0x429 | 0x2230 | R* | INT32 | Velocity Loop Limited Velocity. This is commanded velocity after it passes through acceleration and velocity limits and velocity command filter. Velocity error used by loop is difference between actual velocity and this value.<br>Units: 0.1 encoder counts/s.<br>For estimated velocity. Units: 0.01 RPM.<br>For stepper mode. Units: 0.1 microsteps/s. | |
| 0x2A | 0x42A | None | R* | INT32 | Velocity Loop Error. | |
| 0x2B | 0x42B | None | R* | INT32 | Velocity Loop Integral Sum. | |
| 0x2C | 0x42C | 0x606B | R* | INT32 | Commanded Velocity.<br>Units: 0.1 encoder counts/s.<br>For estimated velocity. Units: 0.01 RPM.<br>For stepper mode. Units: 0.1 microsteps/s. | |
| 0x2D | 0x42D | 0x6062 | R* | INT32 | Commanded Position. Units: counts.<br>Also known as Limited Position in CME. | |
| 0x2E | 0x42E | 0x2381:3 | RF | U16 | Velocity Loop Acceleration Feed Forward (Aff).<br>Acceleration command from trajectory generator is multiplied by this value and result is added to velocity loop output. | |
| 0x2F | 0x42F | 0x2341 | RF | IN0T32 | Programmed Velocity Command.<br>Only used in Programmed Velocity Mode (Desired State (0x24) = 11)<br>Units: 0.1 encoder counts/s.<br>For estimated velocity. Units: 0.01 RPM.<br>For stepper mode. Units: 0.1 microsteps/s. | |
| 0x30 | 0x430 | 0x2382:1 | RF | U16 | Position Loop Proportional Gain (Pp). | |
| 0x31 | 0x431 | 0x2381:4 | RF | INT16 | Velocity Loop Shift Value. After velocity loop is calculated, result is right shifted this (value) many times to arrive at commanded current value. This allows velocity loop gains to have reasonable values for high resolution encoders. | |
| 0x32 | 0x432 | 0x2240 | R* | INT32 | Actual Motor Position. Units: counts.<br>Gives feedback position of motor. For single feedback systems, this is same as Actual Position (0x17). | |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0x33 | 0x433 | 0x2382:2 | RF | U16 | Position Loop Velocity Feed Forward (Vff).<br>Vff value is multiplied by Instantaneous Commanded Velocity (0x3B) generated by trajectory generator. Product is added to output of position loop.<br>This gain is scaled by 1/16384. Therefore, setting this gain to 0x4000 (16384) would cause input velocity to be multiplied by 1.0, and result added to output of position loop. |
| 0x34 | 0x434 | 0x2382:3 | RF | U16 | Position Loop Acceleration Feed Forward (Aff).<br>Aff value is multiplied by Instantaneous Commanded Velocity (0x3B) generated by trajectory generator. Product is added to output of position loop. |
| 0x35 | 0x435 | 0x60F4 | R* | INT32 | Position Loop Error. Units: counts.<br>Difference between Actual Position (0x17) and Commanded Position (0x2D). |
| 0x36 | 0x436 | 0x2100 | RF | U32 | Velocity Loop Acceleration Limit.<br>Units: 1000 counts/$s^2$.<br>Used by velocity loop limiter. Not used when velocity loop is controlled by position loop. |
| 0x37 | 0x437 | 0x2101 | RF | U32 | Velocity Loop Deceleration Limit.<br>Units: 1000 counts/$s^2$.<br>Used by velocity loop limiter. Not used when velocity loop is controlled by position loop. |
| 0x38 | 0x438 | 0x221C | R* | INT16 | Actual Motor Current. Units: 0.01 A.<br>This current is calculated based on both D and Q axis currents. |
| 0x39 | 0x439 | 0x2102 | RF | U32 | Velocity Loop Emergency Stop Deceleration Rate.<br>Units: 1000 counts/$s^2$. |
| 0x3A | 0x43A | 0x2103 | RF | INT32 | Velocity Loop Velocity Limit. Units 0.1 counts/s.<br>This value is limit on commanded velocity used by velocity loop. Note that this limit is always in effect. |
| 0x3B | 0x43B | 0x2250 | R* | INT32 | Instantaneous Commanded Velocity.<br>Units: 0.1 encoder counts/s.<br>This velocity is output of trajectory generator and is value by which position loop's velocity feed forward is multiplied. |
| 0x3C | 0x43C | 0x2251 | R* | U32 | Instantaneous Commanded Acceleration.<br>Units: 10 encoder counts/$s^2$.<br>This acceleration is output of trajectory generator and is value by which position loop's acceleration feed forward is multiplied. |
| 0x3D | 0x43D | 0x2122 | R* | INT32 | Trajectory Destination Position.<br>Units: encoder counts.<br>This is position that trajectory generator is using as its destination. |
| 0x3E | 0x43E | 0x2104 | RF | INT32 | Velocity Window. Units: 0.1 counts/s.<br>If absolute value of velocity loop error exceeds this, then |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| | | | | | velocity window bit in Event Status Register (0xA0) will be set. |
| 0x3F | 0x43F | 0x2105 | RF | U16 | Velocity Window Time. Units: ms.<br>Velocity window bit in Event Status Register (0xA0) will be cleared when absolute velocity error is less than velocity window for this amount of time. |
| 0x40 | 0x440 | 0x2383:1 | F | U16 | Motor Type. Type of motor connected to drive.<br>Bit-mapped as follows: |
| | | | | | Bits: Description |
| | | | | | 0: Set for linear, clear for rotary. |
| | | | | | 1-3: Reserved. |
| | | | | | 4-5: Motor architecture: |
| | | | | | 0: Not specified. |
| | | | | | 1: Brushed servo. |
| | | | | | 2: Microstepper. |
| | | | | | 3: Brushless servo. |
| | | | | | 6-15: Reserved. |
| 0x41 | 0x441 | 0x6404 | F | String | Motor Manufacturer. |
| 0x42 | 0x442 | 0x6403 | F | String | Motor Model. |
| 0x43 | 0x443 | 0x2383:27 | F | INT16 | Motor Units. This is only used by CME for display. (0=metric, 1=English). |
| 0x44 | 0x444 | 0x2383:9 | F | INT32 | Motor Inertia (Mass).<br>Units: Rotary = 0.000001 Kg/cm$^2$<br>Units: Linear = 0.0001 Kg. |
| 0x45 | 0x445 | 0x2383:2 | F | INT16 | Motor Poll Pairs (used only for rotary motors). Number of motor pole pairs (electrical phases) per rotation. For stepper motors, Poll Pairs = (360 deg / Motor deg/step) / 4. |
| 0x46 | 0x446 | 0x2383:16 | F | U16 | Motor Brake Type. 0=present, 1=none. |
| 0x47 | 0x447 | 0x2383:15 | F | U16 | Motor Temperature Sensor Type.<br>0=none, 1=present. |
| 0x48 | 0x448 | 0x2383:12 | F | INT32 | Motor Torque Constant. Units: 0.00001 Nm/A. |
| 0x49 | 0x449 | 0x2383:7 | F | INT16 | Motor Resistance. Units: 10 mΩ. |
| 0x4A | 0x44A | 0x2383:8 | F | INT16 | Motor Inductance. Units: 10 µH. |
| 0x4B | 0x44B | 0x2383:13 | F | INT32 | Motor Peak Torque. Units: 0.00001 Nm units. |
| 0x4C | 0x44C | 0x2383:14 | F | INT32 | Motor Continuous Torque. Units: 0.00001 Nm units. |
| 0x4D | 0x44D | 0x2383:11 | F | INT32 | Motor Max Velocity. Units: 0.1 encoder counts/s. |
| 0x4E | 0x44E | 0x2383:3 | F | U16 | Motor Wiring. 0=standard, 1= drive's U and V outputs are swapped. |
| 0x4F | 0x44F | 0x2383:6 | RF | INT16 | Motor Hall Offset. Units: degrees.<br>Offset angle to be applied to Hall Effect sensors. |
| 0x50 | 0x450 | 0x2383:4 | F | INT16 | Motor Hall Type.<br>Type of Hall Effect sensors attached to motor: |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| | | | | | Value: Description<br>0: No Hall Effect sensors available.<br>1: Digital Hall Effect sensors.<br>2: Analog Hall Effect sensors. |
| 0X51 | 0x451 | 0x2383:10 | F | U32 | Motor back EMF constant (**obsolete**, variable 0x56 is now used which accesses same data but with extended range)<br>Units: rotary 0.01 V/krpm, linear 0.01 V/mps |
| 0x52 | 0x452 | 0x2383:5 | F | INT16 | Motor Hall Effect Wiring. Bit-mapped as follows:<br>NOTE: When analog Halls are used, only bit 8 is relevant.<br>Bits: Description<br>0-2: The Hall wiring code (see below).<br>Value: Hall Ordering<br>0: U V W<br>1: U W V<br>2: V U W<br>3: V W U<br>4: W V U<br>5: W U V<br>6, 7: Reserved<br>3: Reserved.<br>4: Invert W Hall input if set. Inversion occurs after Halls wiring is changed by bits 0-2.<br>5: Invert V Hall input if set. Inversion occurs after Halls wiring is changed by bits 0-2.<br>6: Invert U Hall input if set. Inversion occurs after Halls wiring is changed by bits 0-2.<br>7: Reserved.<br>8: If set, reverse analog Halls.<br>9-15: Reserved. |
| 0x53 | 0x453 | 0x2383:17 | F | U16 | Motor Brake Activation Time. Units: ms. |
| 0x54 | 0x454 | 0x2383:18 | F | U16 | Motor Brake Delay Time. Units: ms.<br>After brake output is activated, drive will stay enabled for this amount of time to allow brake to engage. |
| 0x55 | 0x455 | 0x2383:19 | F | INT32 | Motor Brake Activation Velocity. Units: 0.1 counts/s.<br>Also known as Motor Brake Velocity (CANopen). During Motor Brake Activation Time (0x53), if motor's actual velocity falls below this value brake output is activated immediately. |
| 0x56 | 0x456 | 0x2383:10 | F | U32 | Motor Back EMF Constant.<br>Units: rotary 0.01 V/krpm.<br>Units: linear 0.01 V/mps.<br>Back EMF velocity estimation can be disabled by setting to zero. |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="3">Description</th></tr>
<tr><td>0x57</td><td>0x457</td><td>0x2383:29</td><td>F</td><td>U32</td><td colspan="3">Microsteps/Motor Rev. Units: microsteps.<br>This parameter is used in true microstepping mode.</td></tr>
<tr><td>0x58</td><td>0x458</td><td>0x2383:33</td><td>F</td><td>INT32</td><td colspan="3">Motor Gear Ratio.<br>This parameter may be used to store gear ratio information for dual encoder systems where gearbox sits between two encoders. This parameter is not used by firmware and is supported as convenience to CME program.<br>Gear ratio is ratio of two 16-bit values. First word gives number of motor turns and is numerator. Second word gives number of position turns and is denominator.</td></tr>
<tr><td>0x59</td><td>0x459</td><td>0x2107</td><td>RF</td><td>INT16</td><td colspan="3">Hall Velocity Mode Shift Value. This parameter is only used in Hall velocity mode. It specifies left shift value for position and velocity information calculated in that mode.</td></tr>
<tr><td rowspan="15">0x5A</td><td rowspan="15">0x45A</td><td rowspan="15">0x2241</td><td rowspan="15">RF</td><td rowspan="15">INT16</td><td colspan="3">Encoder Output Configuration.<br>This parameter determines source of buffered encoder output on drives which support it.<br>Bit-mapped as follows:</td></tr>
<tr><td>Bits</td><td colspan="2">Description</td></tr>
<tr><td>0-1</td><td colspan="2">Mode of operation for encoder output lines.</td></tr>
<tr><td></td><td>0</td><td>Output buffered primary encoder (hardware buffering).</td></tr>
<tr><td></td><td>1</td><td>Configure pins as secondary encoder input.</td></tr>
<tr><td></td><td>2</td><td>Output simulated encoder outputs tracking motor encoder.</td></tr>
<tr><td></td><td>3</td><td>Output simulated encoder outputs tracking load encoder.</td></tr>
<tr><td>4</td><td colspan="2">If set, force X and S channels to be inputs no matter what mode bits 0-1 specify. This is useful for some special modes that take commands on these lines while outputting encoder data on the A and B lines.</td></tr>
<tr><td>8-11</td><td colspan="2">For simulated encoder outputs, these bits configure scaling value that adjusts number of encoder output counts for each encoder count on the input.</td></tr>
<tr><td></td><td>0</td><td>No adjustment, 1 count on the encoder is 1 output count.</td></tr>
<tr><td></td><td>1</td><td>Multiply encoder counts by 2.</td></tr>
<tr><td></td><td>2</td><td>Divide encoder counts by 2.</td></tr>
<tr><td></td><td>3</td><td>Divide encoder counts by 4.</td></tr>
<tr><td></td><td>4</td><td>Divide encoder counts by 8.</td></tr>
<tr><td></td><td>5</td><td>Divide encoder counts by 16.</td></tr>
<tr><td></td><td>6</td><td>Divide encoder counts by 32.</td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| | | | | | | 7 | Divide encoder counts by 64. |
| | | | | | | 8 | Divide encoder counts by 128. |
| | | | | | | 9 | Divide encoder counts by 256. |
| | | | | | 12-13 | For simulated encoder outputs, these bits allow the max output frequency to be reduced | |
| | | | | | | 0 | No change to max output frequency |
| | | | | | | 1 | Divide max output frequency by 2 |
| | | | | | | 2 | Divide max output frequency by 4 |
| | | | | | | 3 | Divide max output frequency by 8 |
| 0x5B | 0x45B | 0x2383:32 | F | INT32 | Load Encoder Resolution. Units: Encoder unit/count.<br>Used for linear motors only. Number of Motor Encoder Units (0x61) per encoder count. | | |
| 0x5C | 0x45C | 0x2383:31 | F | INT16 | Load Encoder Direction. 0=normal, 1=reverse.<br>Note: Change in direction will affect motor phasing. | | |
| 0x5D | 0x45D | 0x2383:30 | F | U16 | Load Encoder Type.<br>This parameter identifies type of encoder used on load when running in dual loop mode. Encoding of this parameter has changed over time to support more encoder types than were originally envisioned when parameter was first defined. Bit 12 is used to identify which encoding is active. | | |
| | | | | | Bits | Meaning | |
| | | | | | 0-3 | Encoder hardware to use: | |
| | | | | | | 0 | No load encoder present |
| | | | | | | 1 | Primary (differential) quad encoder |
| | | | | | | 2 | Analog encoder |
| | | | | | | 3 | Secondary quad encoder from input lines |
| | | | | | | 4 | Servo tube/analog halls |
| | | | | | | 5 | Resolver |
| | | | | | | 11 | EnDat absolute encoder |
| | | | | | | 12 | SSI serial encoder |
| | | | | | | 13 | BiSS absolute encoder |
| | | | | | | 14 | Various absolute encoders made by Sanyo Denki, Panasonic and Harmonic Drives |
| | | | | | | 15 | Harmonic Drives custom encoder |
| | | | | | 4 | If set, linear encoder.<br>If clear, rotary encoder. | |
| | | | | | 5 | If set, do not use this encoder for position feedback. | |
| | | | | | 6-15 | Must be zero | |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="3">Description</th></tr>
<tr><td rowspan="12"></td><td rowspan="12"></td><td rowspan="12"></td><td rowspan="12"></td><td rowspan="12"></td><td colspan="3">New encoding supported by 8367 firmware starting with version 2.10:</td></tr>
<tr><td rowspan="8">0-11</td><td colspan="2">Encoder hardware to use:</td></tr>
<tr><td>0-15</td><td>Same encoder types as listed above</td></tr>
<tr><td>16</td><td>Simple analog potentiometer for feedback</td></tr>
<tr><td>17</td><td>Gurley virtual absolute encoder</td></tr>
<tr><td>18</td><td>Custom encoder K</td></tr>
<tr><td>19</td><td>S2 custom encoder</td></tr>
<tr><td>20</td><td>Hiperface</td></tr>
<tr><td>22</td><td>Sankyo absolute encoder</td></tr>
<tr><td>12</td><td colspan="2">Always set to identify new encoding</td></tr>
<tr><td>13</td><td colspan="2">If set, linear encoder.<br>If clear, rotary encoder.</td></tr>
<tr><td>14</td><td colspan="2">If set, do not use this encoder for position feedback.</td></tr>
<tr><td>15</td><td colspan="2">Reserved</td></tr>
<tr><td>0x5E</td><td>0x45E</td><td>0x2231</td><td>R*</td><td>INT32</td><td colspan="3">Load Encoder Velocity. Units: 0.1 encoder counts/s</td></tr>
<tr><td>0x5F</td><td>0x45F</td><td>0x2106</td><td>RF</td><td>9 or 14</td><td colspan="3">Velocity Loop Output Filter. Bi-quad filter which acts on output of velocity loop. 9- or 14-word parameters, see Filter Coefficients.</td></tr>
<tr><td rowspan="17">0x60</td><td rowspan="17">0x460</td><td rowspan="17">0x2383:20</td><td rowspan="17">F</td><td rowspan="17">U16</td><td colspan="3">Motor Encoder Type:</td></tr>
<tr><td>Value</td><td colspan="2">Meaning</td></tr>
<tr><td>0</td><td colspan="2">Primary (differential) quad encoder.</td></tr>
<tr><td>1</td><td colspan="2">No encoder (use motor back EMF for velocity estimation).</td></tr>
<tr><td>2</td><td colspan="2">Analog encoder.</td></tr>
<tr><td>3</td><td colspan="2">Secondary quad encoder from input lines.</td></tr>
<tr><td>4</td><td colspan="2">Low frequency analog encoder.</td></tr>
<tr><td>5</td><td colspan="2">Resolver.</td></tr>
<tr><td>6</td><td colspan="2">Use digital hall signals for position & velocity estimates.</td></tr>
<tr><td>7</td><td colspan="2">Analog encoder updated at current loop rate.</td></tr>
<tr><td>8</td><td colspan="2">Custom encoder</td></tr>
<tr><td>9</td><td colspan="2">Panasonic</td></tr>
<tr><td>10</td><td colspan="2">SPI command (reserved for custom firmware use).</td></tr>
<tr><td>11</td><td colspan="2">EnDat</td></tr>
<tr><td>12</td><td colspan="2">SSI</td></tr>
<tr><td>13</td><td colspan="2">BiSS</td></tr>
<tr><td>14</td><td colspan="2">Serial encoders from Sanyo Denki, Tamagawa, Panasonic and HD systems.</td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 15 | Custom encoders from HD systems. |
| | | | | | 16 | Simple analog potentiometer feedback. |
| | | | | | 17 | Gurley virtual absolute encoder |
| | | | | | 18 | Custom Encoder K |
| | | | | | 19 | S2 custom encoder |
| | | | | | 20 | Hiperface |
| | | | | | 21 | Wire saving incremental encoder which outputs hall signals on encoder lines at power-up. |
| | | | | | 22 | Sankyo absolute encoder |
| | | | | | 23 | Custom HG absolute encoder |
| | | | | | 24 | Digital inputs used as tertiary encoder inputs. Inputs configured as single ended or differential by using Digital Input Command Configuration (0xA8). Not used in Desired State (0x24) modes 3, 13 and 23 (PWM or Digital Input Command Modes) |
| 0x61 | 0x461 | 0x2383:21 | F | INT16 | Motor Encoder Units. Value defines units used to describe linear motor encoders. Not used with rotary motors. | |
| | | | | | Value | Description |
| | | | | | 0 | Microns. |
| | | | | | 1 | Nanometers. |
| | | | | | 2 | Millimeters. |
| 0x62 | 0x462 | 0x2383:23 | F | INT32 | Motor Encoder Counts/Rev. Units: Counts/rev.<br>Used for rotary motors only. When resolver is used as motor feedback, sets resolution of interpolated position. | |
| 0x63 | 0x463 | 0x2383:24 | F | INT16 | Motor Encoder Resolution. Linear motor only.<br>Units: encoder units/count. | |
| 0x64 | 0x464 | 0x2383:25 | F | INT32 | Motor Encoder Electrical Distance. Linear motor only.<br>Units: encoder units/electrical cycle. | |
| 0x65 | 0x465 | 0x2383:22 | F | U16 | Motor Encoder Direction. 0=normal, 1=reverse.<br>Note: Change in direction will affect motor phasing. | |
| 0x66 | 0x466 | 0x2383:26 | F | U32 | Encoder Index Marker Pulse Distance.<br>Units: rotary, counts; linear, encoder units.<br>Reserved for future use. | |
| 0x67 | 0x467 | 0x2383:28 | F | INT16 | Analog Encoder Shift Amount.<br>This value gives number of bits of interpolation to be applied to an analog encoder. Encoder resolution with no interpolation (shift value of 0) is 4 encoder counts/encoder line. Setting this parameter to value of n would give total of 2^(n+2) counts/line. | |
| 0x68 | 0x468 | 0x2402 | R* | INT32 | Captured Index Position. Units: counts.<br>Provides position that axis was in when an index pulse was captured. Configured by setting bits in Position Capture Control Register (0x6C), and status of captured data can be checked in Position Capture Status Register (0x6D).<br>Reading this variable resets *bits 0 & 3* of Position Capture Status Register (0x6D). | |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="2">Description</th></tr>
<tr><td>0x69</td><td>0x469</td><td>0x2232</td><td>R*</td><td>INT32</td><td colspan="2">Unfiltered Motor Encoder Velocity.<br>Units 0.1 counts/s.</td></tr>
<tr><td>0x6A</td><td>0x46A</td><td>0x2113</td><td>RF</td><td>INT32</td><td colspan="2">Commanded Current Ramp Limit. Units: mA/s.<br>Used when running in Current (Torque) mode. Setting this to zero disables slope limiting.</td></tr>
<tr><td>0x6B</td><td>0x46B</td><td>0x2108</td><td>RF</td><td>9 or 14</td><td colspan="2">Velocity Loop Command Filter Coefficients.<br>Biquad filter structure that acts on command input of velocity loop just after velocity & acceleration limiting. 9- or 14-word parameters, see Filter Coefficients.</td></tr>
<tr><td rowspan="14">0x6C</td><td rowspan="14">0x46C</td><td rowspan="14">0x2400</td><td rowspan="14">RF</td><td rowspan="14">INT16</td><td colspan="2">Position Capture Control Register. Sets up position capture based on index or home input.<br>Bit-mapped as follows:</td></tr>
<tr><td>Bits</td><td>Description</td></tr>
<tr><td>0</td><td>If set, Captured Index Position (0x68) is captured on rising edge of index input.</td></tr>
<tr><td>1</td><td>If set, Captured Index Position (0x68) is captured on falling edge of index input.</td></tr>
<tr><td>2</td><td>If set, Captured Index Position (0x68) value will not be overwritten by new position until it has been read.<br>If clear, new positions will overwrite old positions.</td></tr>
<tr><td>3, 4</td><td>Reserved.</td></tr>
<tr><td>5</td><td>If set, Captured Home Position (0x10A) will be captured on active to inactive edge of home input switch.<br>If clear, home position will be captured on inactive to active edge.</td></tr>
<tr><td>6</td><td>If set, Captured Home Position (0x10A) will not be overwritten by new position until it has been read.<br>If clear, new positions will overwrite old positions.</td></tr>
<tr><td>7</td><td>Reserved.</td></tr>
<tr><td>8</td><td>If set, enable high speed input position capture, Captured Position for High Speed Position Capture (0x111).</td></tr>
<tr><td>9</td><td>If set, don't overwrite high speed input capture positions.</td></tr>
<tr><td>10</td><td>If set, latch high speed position capture.</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
<tr><td>12</td><td>Clear Actual Position (0x17) on every encoder index pulse.</td></tr>
<tr><td>0x6D</td><td>0x46D</td><td>0x2401</td><td>R*</td><td>INT16</td><td colspan="2">Position Capture Status Register. This register shows status of index/home capture mechanism.<br>Bit-mapped as follows:</td></tr>
</table>

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="2">Description</th></tr>
<tr><td rowspan="12"></td><td rowspan="12"></td><td rowspan="12"></td><td rowspan="12"></td><td rowspan="12"></td><td>Bits</td><td>Description</td></tr>
<tr><td>0</td><td>If set, index position has been captured.<br>Cleared when captured position is read.</td></tr>
<tr><td>1-2</td><td>Reserved.</td></tr>
<tr><td>3</td><td>If set, new index transition occurred when captured position was already stored.<br>Depending on mode, new position may have been discarded or overwritten previously stored position.</td></tr>
<tr><td>4</td><td>If set, home position has been captured.<br>Cleared when captured position is read.</td></tr>
<tr><td>5-6</td><td>Reserved.</td></tr>
<tr><td>7</td><td>If set, new home pin transition occurred when captured position was already stored.<br>Depending on mode, new position may have been discarded or overwritten previously stored position.</td></tr>
<tr><td>8</td><td>If set, new high-speed capture data.</td></tr>
<tr><td>9-10</td><td>Reserved.</td></tr>
<tr><td>11</td><td>If set, high-speed capture overflow.</td></tr>
<tr><td>0x6E</td><td>0x46E</td><td>0x2383:34</td><td>F</td><td>INT16</td><td colspan="2">Number of Resolver Cycles/Motor Rev.<br>Used only with resolver feedback devices.</td></tr>
<tr><td rowspan="13">0x6F</td><td rowspan="13">0x46F</td><td rowspan="13">0x2140</td><td rowspan="13">RF</td><td rowspan="13">INT16</td><td colspan="2">PWM Mode and Status. Bit-mapped as follows:</td></tr>
<tr><td>Bits</td><td>Description</td></tr>
<tr><td>0</td><td>If set, force bus clamping.<br>If clear, disable bus clamping.<br>If bit 1 set, this bit is ignored.</td></tr>
<tr><td>1</td><td>If set, automatic bus clamping.<br>Setting this bit causes bus clamping mode to be automatically selected based on output voltage.<br>Bit 0 ignored if this bit is set.</td></tr>
<tr><td>2</td><td>Reserved</td></tr>
<tr><td>3</td><td>Factory reserved.<br>If set, short motor outputs when disabled.</td></tr>
<tr><td>4</td><td>If set, use hexagonal voltage limiting.<br>If clear, use circular voltage limiting.<br>This setting only used with brushless motors.</td></tr>
<tr><td>5</td><td>Reserved</td></tr>
<tr><td>6</td><td>If set, double PWM frequency.</td></tr>
<tr><td>7</td><td>Reserved</td></tr>
<tr><td>8</td><td>Status bit set when bus clamping is active.</td></tr>
<tr><td>0x70</td><td>0x470</td><td>0x2193:1</td><td>RF</td><td>3 -5</td><td colspan="2">Output 0 Configuration.<br>For notes on Output numbering see Input/Output Numbering.</td></tr>
</table>

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="3">Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="3">Data type is dependent on configuration and uses 1- to 5-words.<br><br>First word is bit-mapped configuration value. Remaining words give additional parameter data used by output pin. Typically, second and third words are used as 32-bit mask to identify which bit(s) in status register output should follow.<br>If any selected bits in status register are set, then output will go active. If no selected bits are set, then output will be inactive.<br><br>Output 0 may be programmed as sync output for use in synchronizing multiple drives. In this configuration, first word of this variable should be set to 0x0200 (i.e., only bit 9 is set) and remaining words should be set to zero. Note that only Output 0 has this feature. Attempting to program any other output pin as sync output will have no effect.<br><br>The first word is bit-mapped as follows:</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Bits</td><td colspan="2">Configuration</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td rowspan="11">0-4</td><td colspan="2">Define which internal register drives output. Acceptable values for these bits are as follows:</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Value</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>0</td><td>Track bits in Event Status Register (0xA0).</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>1</td><td>Track bits in Latched Event Status Register (0xA1).</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>2</td><td>Track bits in Manual Output Control Register. See Output States and Program Control (0xAB).</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>3</td><td>Track bits in Trajectory Status Register (0xC9).</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>4</td><td>Go active if position is between the two positions specified in words 2, 3 (low) and 4, 5 (high). If bit 14 is set, commanded position is used. If bit 14 is clear, actual position is used.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>5</td><td>Go active on low to high crossing of position specified by words 2, 3. Stay high for number of ms specified by words 4, 5. If bit 14 is set, commanded position is used. If bit 14 is clear, actual position is used.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>6</td><td>Same as 5, but for high to low crossings.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>7</td><td>Same as 5 but for any crossing.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>8</td><td>Go active if motor phase angle (plus an offset) is between 0 and 180 degrees. Offset is set using</td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| | | | | | | | first word of extra data in units of degrees. |
| | | | | | | 9 | Pulse output each time a position is crossed from an array of positions stored in trace memory. |
| | | | | | | 10 | Use output to trigger an external regen resister. |
| | | | | | | 11 | For EtherCAT drives, pulse on SYNC0 signal. |
| | | | | | | 16 | Track Hardware Position Compare function on drives supporting it. |
| | | | | | | 17 | Logical OR of function 0 and 2. Output will track both a set of selected Event Status Register (0xA0) bits and Output States and Program Control (0xAB). Bits 14 and 15 of configuration also effect operation. If any of selected Event Status bits are set then output is active (if bit 14 is clear) or inactive (if bit 14 is set). If selected Event Status bits aren't active, then if Output States and Program Control (0xAB) bit is set then output is either active (bit 15 is clear) or inactive (bit 15 is set). If neither of those conditions is true, then output is either active (if bit 15 is set) or inactive (if bit 15 is clear). |
| | | | | | | 18 | Firmware 2.98 and later.<br>BEL drives only support this special mode in which output is configured as a brake which goes active for programmable time after which it starts to PWM with programmable on and off times. Word 2 of output configuration gives PWM on time in us. Word three gives PWM period in us. Word four is reserved, word 5 gives delay before PWM starts in ms. |
| | | | | | | 19 | Output is active if drive is being disabled by STO input. |
| | | | | | 5-7 | Reserved. | |
| | | | | | 8 | If set, invert normal active state of output. E.g. outputs that are normally active low become active high. For programmed control, see Output States and Program Control (0xAB). | |
| | | | | | 9 | If set, program output as sync output. This bit is reserved for all output pins except pin 0. | |
| | | | | | 10-11 | Reserved. | |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="3">Description</th></tr>
<tr><td rowspan="5"></td><td rowspan="5"></td><td rowspan="5"></td><td rowspan="5"></td><td rowspan="5"></td><td>12-13</td><td colspan="2">Axis number for multi-axis drives.</td></tr>
<tr><td>14-15</td><td colspan="2">Usage depends on output function selected.</td></tr>
<tr><td colspan="3">Remaining words give additional parameter data used by output pin. Typically, second and third words are used as 32-bit bitmask to identify which bit(s) in Event Status Register (0xA0) output should follow. If any selected bits in Event Status Register (0xA0) are set, then output will go active.<br>If no selected bits in Event Status Register (0xA0) are set then output will be inactive.<br>Version 4.80 firmware added several advanced output pin configurations which required more parameter data. At that point support for optional<br>5-word configuration was added to firmware.<br>For these output pin configurations, words 2 and 3 define one 32-bit parameter and words 4 and 5<br>define second 32-bit parameter.<br>Output 0 may be programmed as sync output for use in synchronizing multiple drives. In this configuration, first word of this variable should be set to 0x0200 (i.e. only bit 9 is set) and remaining words should be set to zero.<br>Note that only Output 0 has this feature. Attempting to program any other output as sync output will have no effect.</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3"></td></tr>
<tr><td>0x71</td><td>0x471</td><td>0x2193:2</td><td>RF</td><td>See text</td><td colspan="3">Output 1 Configuration. See Output 0 Configuration (0x70).</td></tr>
<tr><td>0x72</td><td>0x472</td><td>0x 2193:3</td><td>RF</td><td>See text</td><td colspan="3">Output 2 Configuration. See Output 0 Configuration (0x70).</td></tr>
<tr><td>0x73</td><td>0x473</td><td>0x 2193:4</td><td>RF</td><td>See text</td><td colspan="3">Output 3 Configuration. See Output 0 Configuration (0x70).</td></tr>
<tr><td>0x74</td><td>0x474</td><td>0x 2193:5</td><td>RF</td><td>See text</td><td colspan="3">Output 4 Configuration. See Output 0 Configuration (0x70).</td></tr>
<tr><td>0x75</td><td>0x475</td><td>0x 2193:6</td><td>RF</td><td>See text</td><td colspan="3">Output 5 Configuration. See Output 0 Configuration (0x70).</td></tr>
<tr><td>0x76</td><td>0x476</td><td>0x 2193:7</td><td>RF</td><td>See text</td><td colspan="3">Output 6 Configuration. See Output 0 Configuration (0x70).</td></tr>
<tr><td>0x77</td><td>0x477</td><td>0x 2193:8</td><td>RF</td><td>See text</td><td colspan="3">Output 7 Configuration. See Output 0 Configuration (0x70).</td></tr>
<tr><td rowspan="8">0x78</td><td rowspan="8">0x478</td><td rowspan="8">0x 2192:1</td><td rowspan="8">RF</td><td rowspan="8">U16</td><td colspan="3">Input 0 Configuration. Assigns function to input pin. All values not listed below are reserved for future use.<br>For notes on Input numbering see Input/Output Numbering.<br>Sync Input function is only valid for high-speed input pins. In addition, input pins 2 & 3 of Accelus and Junus drives do not support this feature.</td></tr>
<tr><td>Bits</td><td colspan="2">Configuration</td></tr>
<tr><td rowspan="6">0-7</td><td>Value</td><td>Meaning</td></tr>
<tr><td>0</td><td>No function</td></tr>
<tr><td>1</td><td>Reserved for future use (no function).</td></tr>
<tr><td>2</td><td>Reset drive on rising edge of input.</td></tr>
<tr><td>3</td><td>Reset drive on falling edge of input.</td></tr>
<tr><td colspan="3"></td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| | | | | | | 4* | Positive limit switch. Active high. |
| | | | | | | 5* | Positive limit switch. Active low. |
| | | | | | | 6* | Negative limit switch. Active high. |
| | | | | | | 7* | Negative limit switch. Active low. |
| | | | | | | 8* | Motor temperature switch. Active high. |
| | | | | | | 9* | Motor temperature switch. Active low. |
| | | | | | | 10* | Clear faults on rising edge, disable drive while high. |
| | | | | | | 11* | Clear faults on falling edge, disable drive while low. |
| | | | | | | 12* | Reset on rising edge, disable drive while high. |
| | | | | | | 13* | Reset on falling edge, disable drive while low. |
| | | | | | | 14* | Home switch. Active high. |
| | | | | | | 15* | Home switch. Active low. |
| | | | | | | 16* | Drive disable. Active high |
| | | | | | | 17* | Drive disable. Active low. |
| | | | | | | 18 | Sync input on rising edge. If bit 8 is set, pin debounce time is used as sync offset in 0.1 us units. |
| | | | | | | 19 | Sync input on falling edge. If bit 8 is set, pin debounce time is used as sync offset in 0.1 us units. |
| | | | | | | 20* | Halt motor. Active high. |
| | | | | | | 21* | Halt motor. Active low. |
| | | | | | | 22 | Scale analog input. Active high. |
| | | | | | | 23 | Scale analog input. Active low. |
| | | | | | | 24* | High speed position capture on rising edge. Only for high speed inputs. |
| | | | | | | 25* | High speed position capture on falling edge. Only for high speed inputs. |
| | | | | | | 26 | Count rising edges of input to indexer register. Register number identified by bits 8-11. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| | | | | | | 27 | Count falling edges of input to indexer register. Register number identified by bits 8-11. |
| | | | | | | 28* | Encoder fault input. Active high. |
| | | | | | | 29* | Encoder fault input. Active low. |
| | | | | | | 30-35 | Reserved. |
| | | | | | | 36 | Abort move on rising edge if greater than *n* counts from destination position. Number of counts *n* is stored in an index register identified by bits 8-11. |
| | | | | | | 37 | Abort move on falling edge if greater than *n* counts from destination position. Number of counts *n* is stored in an index register identified by bits 8-11. |
| | | | | | | 38* | Mark HV loss on rising edge, disable while high. |
| | | | | | | 39* | Mark HV loss on falling edge, disable while low. |
| | | | | | | 40* | Update trajectory on rising edge. |
| | | | | | | 41* | Update trajectory on falling edge. |
| | | | | | | 42* | Clear faults & event latch on rising edge. |
| | | | | | | 43* | Clear faults & event latch on falling edge. |
| | | | | | | 44* | Disable simulated encoder output when low. Burst current position on encoder output on rising edge. |
| | | | | | | 45* | Disable simulated encoder output when high. Burst current position on encoder output on falling edge. |
| | | | | | | 46 | Disable drive and act like safety input is active when high.<br>Additionally, bits 8-11 of configuration word are set in Safety Status Register (0x139) bits 0-3. This input type is intended for custom hardware that implements a STO circuit external to drive. |
| | | | | | | 47 | Like input type 46, but active low. |
| | | | | | | *These functions use bit 8 to indicate that input function should apply to all axes. This feature is only available in Plus products. | |
| | | | | | 8-11 | Used to pass parameters to input pin functions. | |
| | | | | | 12-13 | Used to select axis on multi-axis drives. | |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0x79 | 0x479 | 0x2192:2 | RF | U16 | Input 1 Configuration. See Input 0 Configuration (0x78). |
| 0x7A | 0x47A | 0x2192:3 | RF | U16 | Input 2 Configuration. See Input 0 Configuration (0x78). |
| 0x7B | 0x47B | 0x2192:4 | RF | U16 | Input 3 Configuration. See Input 0 Configuration (0x78). |
| 0x7C | 0x47C | 0x2192:5 | RF | U16 | Input 4 Configuration. See Input 0 Configuration (0x78). |
| 0x7D | 0x47D | 0x2192:6 | RF | U16 | Input 5 Configuration. See Input 0 Configuration (0x78). |
| 0x7E | 0x47E | 0x2192:7 | RF | U16 | Input 6 Configuration. See Input 0 Configuration (0x78). |
| 0x7F | 0x47F | 0x2192:8 | RF | U16 | Input 7 Configuration. See Input 0 Configuration (0x78). |
| 0x80 | 0x480 | 0x6503 | F* | String | Drive Model Number. |
| 0x81 | 0x481 | 0x2384:1<br>or<br>0x1018:4 | F* | U32 | Drive Serial Number. |
| 0x82 | 0x482 | 0x2384:3 | F* | INT16 | Drive's rated Peak Current. Unit: 0.01 A. |
| 0x83 | 0x483 | 0x2384:4 | F* | INT16 | Drive's rated Continuous Current. Unit: 0.01 A |
| 0x84 | 0x484 | 0x2384:14 | F* | INT16 | Current Corresponding to Max A/D Reading.<br>Units: 0.01 A. |
| 0x85 | 0x485 | 0x2384:11 | F* | U16 | PWM Period. Units: 10 ns. |
| 0x86 | 0x486 | 0x2384:12 | F* | U16 | Drive Servo Period (PWM periods). Servo loop update period as a multiple of the current loop period. |
| 0x87 | 0x487 | None | F* | U16 | Product Family. Identifies the drive product family. For specific drive hardware type, see Drive Hardware Type (0xAD). |
| 0x88 | 0x488 | 0x2384: 5 | F* | INT16 | Drive's rated Time at Peak Current. Units: ms.<br>Maximum time for which drive is rated to output peak current. |
| 0x89 | 0x489 | 0x2384:6 | F* | INT16 | Drive's rated Maximum Voltage. Units: 0.1 V.<br>Maximum bus voltage rating. |
| 0x8A | 0x48A | 0x2384:15 | F* | INT16 | Drive's rated Voltage Corresponding to Max A/D Reading.<br>Units: 0.1 V. |
| 0x8B | 0x48B | 0x2384:7 | F* | INT16 | Drive's rated Minimum Voltage. Units: 0.1 V.<br>Minimum bus voltage rating. |
| 0x8C | 0x48C | 0x2384:9 | F* | INT16 | Drive's rated Maximum Temperature.<br>Units: degrees C. |
| 0x8D | 0x48D | 0x2384:2 | F* | String | Manufacturing info (date code) of drive |
| 0x8E | 0x48E | 0x2384:16 | F* | INT16 | Analog Input Reference Scaling Factor. This is voltage applied to analog input which causes max A/D value on drive. |
| 0x90 | 0x490 | None | R | U32 | Serial Port Baud Rate. Units: bits/s.<br>*Defaults to 9600 at power up or reset.* |
| 0x91 | 0x491 | None | R* | INT16 | Maximum number of data words allowed per binary command over serial interface. |
| 0x92 | 0x492 | 0x21A0 | F | String | Axis label string |
| 0x93 | None | None | F | U32 | Reserved |
| 0x94 | 0x494 | 0x2384:24 | R* | INT16 | Firmware Version Number. Version number consists of major and minor version number.<br>Minor number passed in bits 0-7; major number passed in |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>bits 8-15.<br>Example: version 1.12 would be encoded 0x010C.</td></tr>
<tr><td>0x95</td><td>0x495</td><td>0x2421</td><td>F</td><td>String</td><td>Host Configuration State.<br>Reserved for use by CME software.</td></tr>
<tr><td>0x96</td><td>0x496</td><td>0x2312</td><td>RF</td><td>INT16</td><td>Calibration Offset for Analog Input or Analog Reference. This voltage is added to analog reference input and is calibrated at factory to give zero reading for zero input voltage.</td></tr>
<tr><td>0x97</td><td>0x497</td><td>0x2384:10</td><td>F*</td><td>INT16</td><td>Hysteresis value for drive over temperature cut-out. Units: degrees C.</td></tr>
<tr><td>0x98</td><td>0x498</td><td>0x2330</td><td>RF</td><td>INT16</td><td>Function Generator Configuration. Configures drive’s internal function generator, which can drive current, velocity or position loop. Bit-mapped as follows:
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0-2</td><td>Function code (type of waveform to generate):
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>None (disabled).</td></tr>
<tr><td>1</td><td>Square wave output</td></tr>
<tr><td>2</td><td>Sine wave output</td></tr>
<tr><td>3</td><td>White noise (Plus & AFS products)</td></tr>
<tr><td>4</td><td>Triangular waveform (Plus & AFS products)</td></tr>
</table></td></tr>
<tr><td>3</td><td>Reserved.</td></tr>
<tr><td>4-5</td><td>Function generator injection into running loop. Allows output of function generator to be injected into input of either current or velocity loop while drive is operating in some mode of operation other than function generator mode. This feature is only available on plus product drives starting with firmware 3.34. This can be useful for testing system response in presence of a disturbance.
<table>
<tr><th>Mode</th><th>Description</th></tr>
<tr><td>0</td><td>No function generator injection</td></tr>
<tr><td>1</td><td>Inject function generator output into input of current loop</td></tr>
<tr><td>2</td><td>Inject function generator output into input of velocity loop</td></tr>
<tr><td>3</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>6-7</td><td>Reserved.</td></tr>
<tr><td>8</td><td>If set, use high resolution mode. In this mode Function Generator Frequency (0x99) is in units of 0.01 Hz. (Plus & AFS products).</td></tr>
<tr><td>9-11</td><td>Reserved</td></tr>
<tr><td>12</td><td>If set, one shot mode.</td></tr>
</table></td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 13 | If set, invert every other period. |
| | | | | | 14-15 | Reserved. |
| | | | | | Note that drive is placed under control of function generator by setting Desired State (0x24) to one of the following values:<br>4 (function generator drives current loop);<br>14 (function generator drives velocity loop);<br>24 (function generator drives position loop in servo mode);<br>34 (function generator drives position loop in stepper mode).<br>Note that if one-shot mode is selected, then after one period (two if invert is selected), function type will reset to zero. | |
| 0x99 | 0x499 | 0x2331 | RF | U16 | Function Generator Frequency. Units: Hz.<br>Plus & AFS products support high-resolution mode.<br>Units: 0.01 Hz. See bit 8 of Function Generator Configuration (0x98). | |
| 0x9A | 0x49A | 0x2332 | RF | INT32 | Function Generator Amplitude. Amplitude of signal generated by internal function generator.<br>Units depend on operating mode: | |
| | | | | | Mode | Units |
| | | | | | Current | 0.01 A. |
| | | | | | Velocity | 0.1 encoder counts/s. |
| | | | | | Position | Encoder counts. |
| 0x9B | 0x49B | 0x2333 | RF | U16 | Function Generator Duty Cycle (square wave only). Units: 0.1% (for instance, 1000 for 100%). | |
| 0x9C | 0x49C | 0x2384:8 | F* | U16 | Hysteresis for Maximum Bus Voltage Cut-Out.<br>Units: 0.1 V. | |
| 0x9D | 0x49D | 0x2384:18 | F* | U16 | PWM Dead Time at Continuous Current Limit.<br>Units: CPU cycles.<br>This parameter gives PWM dead time used at or above continuous current limit. Dead time below continuous current limit is defined by linear function of this parameter and PWM Dead Time at Zero Current (0x9F). | |
| 0x9E | 0x49E | 0x2384:17 | F* | U16 | Drive Minimum PWM Off Time. Units: 10 ns.<br>This parameter gives minimum amount of time for which all PWM outputs must be disabled for each current loop cycle. | |
| 0x9F | 0x49F | 0x2384:19 | F* | U16 | PWM Dead Time at Zero Current. Units: CPU cycles. This parameter gives PWM dead time at zero current. Dead time above zero current is defined by linear function of this parameter and PWM Dead Time at Continuous Current Limit (0x9D). | |
| 0xA0 | 0x4A0 | 0x1002 | R* | U32 | Event Status Register.<br>Bit-mapped as follows: | |
| | | | | | Bits | Description |
| | | | | | 0 | Short circuit detected |
| | | | | | 1 | Drive over temperature |
| | | | | | 2 | Over voltage |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 3 | Under voltage |
| | | | | | 4 | Motor temperature sensor active |
| | | | | | 5 | Encoder power error |
| | | | | | 6 | Motor phasing error |
| | | | | | 7 | Current output limited |
| | | | | | 8 | Voltage output limited |
| | | | | | 9 | Positive limit switch active |
| | | | | | 10 | Negative limit switch active |
| | | | | | 11 | Enable input not active |
| | | | | | 12 | Drive is disabled by software |
| | | | | | 13 | Trying to stop motor |
| | | | | | 14 | Motor brake activated |
| | | | | | 15 | PWM outputs disabled |
| | | | | | 16 | Positive software limit condition |
| | | | | | 17 | Negative software limit condition |
| | | | | | 18 | Following Error Fault. A following error has occurred, and drive is in following error mode. |
| | | | | | 19 | Following Error Warning. Indicates position error is greater than position following warning. |
| | | | | | 20 | Drive is currently in reset condition |
| | | | | | 21 | Position has wrapped. Position variable cannot increase indefinitely. After reaching a certain value the variable rolls back. This type of counting is called position wrapping or modulo count |
| | | | | | 22 | Drive fault. Fault configured as latching in Fault Mask (0xA7) has occurred. Latched faults may be cleared using Latching Fault Status Register (0xA4). |
| | | | | | 23 | Velocity limit (0x3A) has been reached |
| | | | | | 24 | Acceleration limit (0x36) has been reached |
| | | | | | 25 | Position Tracking. Position Loop Error (0x35) is outside of Following Error Fault Limit (0xBA). |
| | | | | | 26 | Home switch is active |
| | | | | | 27 | In motion. Set if trajectory generator is running profile or Following Error Fault Limit (0xBA) is outside tracking window.<br>Clear when drive is settled in position. |
| | | | | | 28 | Velocity window. Set when velocity error is larger than programmed velocity window |
| | | | | | 29 | Phase not yet initialized. If drive is phasing with no Halls, this bit is set until drive has initialized its phase. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 30 | Command fault. CANopen or EtherCAT master not sending commands or PWM command not present.<br>Note: If *Allow 100% Output* option is enabled by setting Bit 3 of Digital Input Command Configuration (0xA8) this fault will not detect missing PWM command. |
| | | | | | 31 | Reserved. |
| 0xA1 | 0x4A1 | 0x2181 | R | U32 | Latched Event Status Register. This is latched version of Event Status Register (0xA0). Bits are set by drive when events occur. Bits are only cleared by writing to this parameter as explained below: When writing to Latched Event Status Register, any bit set will cause corresponding bit in register to be cleared. For example, to clear over voltage bit, write decimal 4 or 0x4 to register. To clear all bits, write 0xffffffff to register | |
| 0xA2 | 0x4A2 | 0x2261 | R* | INT16 | Hall Input State. Lower three bits of returned value give present state of Hall input pins. Hall state is value of Hall lines AFTER ordering and inversions specified in Hall Wiring Configuration (0x52) have been applied. | |
| 0XA3 | 0x4A3 | None | R | U32 | Drive test parameter. This parameter is reserved for use by Copley during drive test. | |
| 0xA4 | 0x4A4 | 0x2183 | R | U32 | Latching Fault Status Register. Bit-mapped to show which latching faults have occurred in drive. When latching fault has occurred, the *fault bit (bit 22)* of Event Status Register (0xA0) is set.<br>Cause of fault can be read from this register.<br>To clear fault condition, write a 1 to associated bit in this register. Events that cause drive to latch fault are programmable.<br>See Fault Mask (0xA7) for details. | |
| | | | | | Latched Faults | |
| | | | | | Bits | Fault Description |
| | | | | | 0 | Data flash CRC failure. This fault is considered fatal and cannot be cleared. This bit is read-only and will always be set. If drive detects corrupted flash data values on startup it will remain disabled and indicate fault condition. |
| | | | | | 1 | Drive internal error. This bit is read-only and will always be set. If drive fails its power-on self-test, it will remain disabled and indicate fault condition. |
| | | | | | 2 | Short circuit. If set: programs drive to latch a fault when short circuit is detected on motor outputs.<br>If clear: programs drive to disable outputs for 100 ms after short circuit and then re-enable. |
| | | | | | 3 | Drive over temperature. If set: programs drive to latch a fault when drive over temperature event happens. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | | If clear: programs drive to re-enable as soon as it cools sufficiently from over temperature event. |
| | | | | | 4 | Motor over temperature. If set: programs drive to latch a fault when motor temperature sensor input activates.<br>If clear: programs drive to re-enable as soon as over temperature input becomes inactive. |
| | | | | | 5 | Over voltage. If set: programs drive to latch a fault when excessive bus voltage is detected.<br>If clear: programs drive to re-enable as soon as bus voltage is within normal range. |
| | | | | | 6 | Under voltage. If set: programs drive to latch a fault condition when inadequate bus voltage is detected.<br>If clear: programs drive to re-enable as soon as bus voltage is within normal range. |
| | | | | | 7 | Feedback fault. If set: programs drive to latch a fault when feedback faults occur. Feedback faults occur if too much current is drawn from 5 V source on drive, resolver or analog encoder is disconnected, or resolver or analog encoder has levels out of tolerance. |
| | | | | | 8 | Phasing error. If set: programs drive to latch a fault when phasing errors occur.<br>If clear: programs drive to re-enable when phasing error is removed. |
| | | | | | 9 | Following error. If set: programs the drive to latch a fault and disable drive when following error occurs.<br>If clear: programs drive to abort current move and remain enabled when following error occurs. |
| | | | | | 10 | If set: programs drive to latch a fault when output current is limited by $I^2T$ algorithm. |
| | | | | | 11 | FPGA failure. This bit is read-only. |
| | | | | | 12 | Command input lost fault. If set: programs drive to latch a fault and disable when command input is lost. |
| | | | | | 13 | Unable to initialize internal drive hardware. This bit is read-only. |
| | | | | | 14 | If set, programs drive to latch a fault when there is safety circuit consistency check failure. |
| | | | | | 15 | If set, programs drive to latch a fault when drive is unable to control motor current. |
| | | | | | 16 | If set, programs drive to latch a fault when motor wiring is disconnected, see Open Motor Wiring Check Current (0x19D). |
| | | | | | 17 | Reserved. |
| | | | | | 18 | Safe torque off active |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="2">Description</th></tr>
<tr><td>0xA5</td><td>0x4A5</td><td>0x2191</td><td>RF</td><td>U16</td><td colspan="2">Input Pin Configuration Register. Some drives have one or more pull-up resistors associated with their general-purpose input pins. On these drives, state<br>of pull-ups can be controlled by writing to this register.<br>This register has one bit for each pull-up resistor available on drive. Setting bit causes resistor to pull any inputs connected to it up to high state when they are not connected. Bit 0 controls first pullup resistor on drive, bit 1 controls second pullup resistor, etc. Please refer to drive datasheet to determine how many pullup resistors are available for particular drive.<br>On drives that allow groups of inputs to be configured as either single ended or differential, bit 8 controls this feature. Set bit 8 to 0 for single ended, 1 for differential.</td></tr>
<tr><td rowspan="18">0xA6</td><td rowspan="18">0x4A6</td><td rowspan="18">0x2190</td><td rowspan="18">R*</td><td rowspan="18">U16</td><td colspan="2">Input Pin States. The 16-bit value returned by this command gives current state (high/low) of drive’s input pins after debounce. Each bit represents one input as shown below. See also Input Pin States,<br>32-Bit (0x15C) for newer drives which support more than 16 input pins.</td></tr>
<tr><td>Bits</td><td>Description</td></tr>
<tr><td>0</td><td>Programmable input pin 0 (In1)</td></tr>
<tr><td>1</td><td>Programmable input pin 1 (In2)</td></tr>
<tr><td>2</td><td>Programmable input pin 2 (In3)</td></tr>
<tr><td>3</td><td>Programmable input pin 3 (In4)</td></tr>
<tr><td>4</td><td>Programmable input pin 4 (In5)</td></tr>
<tr><td>5</td><td>Programmable input pin 5 (In6)</td></tr>
<tr><td>6</td><td>Programmable input pin 6 (In7)</td></tr>
<tr><td>7</td><td>Programmable input pin 7 (In8)</td></tr>
<tr><td>8</td><td>Programmable input pin 8 (In9)</td></tr>
<tr><td>9</td><td>Programmable input pin 9 (In10)</td></tr>
<tr><td>10</td><td>Programmable input pin 10 (In11)</td></tr>
<tr><td>11</td><td>Programmable input pin 11 (In12)</td></tr>
<tr><td>12</td><td>Programmable input pin 12 (In13)</td></tr>
<tr><td>13</td><td>Programmable input pin 13 (In14)</td></tr>
<tr><td>14</td><td>Programmable input pin 14 (In15)</td></tr>
<tr><td>15</td><td>Programmable input pin 15 (In16)</td></tr>
<tr><td>0xA7</td><td>0x4A7</td><td>0x2182</td><td>RF</td><td>U32</td><td colspan="2">Fault Mask. This variable is used to configure which drive events cause latching faults.<br>Setting fault mask bit to 1 causes associated drive event to cause latching fault when it occurs. Setting fault mask bit to 0 disables fault latching on associated event.<br>Latched faults may be cleared using Latching Fault Status Register (0xA4).</td></tr>
<tr><td>Bits</td><td>Fault Description</td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 0 | Data flash CRC failure. This bit is read-only and will always be set.<br>If drive detects corrupted flash data values on startup it will remain disabled and indicate a fault condition. |
| | | | | | 1 | Drive internal error. This bit is read-only and will always be set.<br>If drive fails its power-on self-test, it will remain disabled and indicate a fault condition. |
| | | | | | 2 | Short circuit. If set: programs drive to latch a fault when short circuit is detected on motor outputs.<br>If clear: programs drive to disable outputs for 100 ms after short circuit and then re-enable. |
| | | | | | 3 | Drive over temperature. If set: programs drive to latch a fault when drive over temperature event happens. If clear: programs drive to re-enable as soon as it cools sufficiently from over temperature event. |
| | | | | | 4 | Motor over temperature. If set: programs drive to latch a fault when motor temperature sensor input activates.<br>If clear: programs drive to re-enable as soon as over temperature input becomes inactive. |
| | | | | | 5 | Over voltage. If set: programs drive to latch a fault when excessive bus voltage is detected.<br>If clear: programs drive to re-enable as soon as bus voltage is within normal range. |
| | | | | | 6 | Under voltage. If set: programs drive to latch a fault when inadequate bus voltage is detected.<br>If clear: programs drive to re-enable as soon as bus voltage is within normal range. |
| | | | | | 7 | Feedback fault. If set: programs drive to latch a fault when feedback faults occur. Feedback faults occur if too much current is drawn from 5 V source on drive, resolver or analog encoder is disconnected or resolver or analog encoder has levels out of tolerance. |
| | | | | | 8 | Phasing error. If set: programs drive to latch a fault when phasing errors occur.<br>If clear: programs drive to re-enable when phasing error is removed. |
| | | | | | 9 | Following error. If set: programs drive to latch a fault and disable when following error occurs.<br>If clear: programs drive to abort current move and remain enabled when following error occurs. |
| | | | | | 10 | If set: programs drive to latch a fault when output current is limited by $I^2T$ algorithm. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| | | | | | 11 | FPGA failure. This bit is read-only. | |
| | | | | | 12 | Command input lost fault. If set: programs drive to latch a fault and disable when command input is lost. | |
| | | | | | 13 | Unable to initialize internal drive hardware. This bit is read-only. | |
| | | | | | 14 | If set, programs drive to latch a fault when there is safety circuit consistency check failure. | |
| | | | | | 15 | If set, programs drive to latch a fault when drive is unable to control motor current. | |
| | | | | | 16 | If set, programs drive to latch a fault when motor wiring is disconnected, see Open Motor Wiring Check Current (0x19D). | |
| | | | | | 17-31 | Reserved. | |
| 0xA8 | 0x4A8 | 0x2320 | RF | INT16 | Digital Input Command Configuration. Defines configuration of digital input commands when drive is running in a mode that uses them as a control source.<br>The lower 8 bits control PWM input configuration for controlling current and velocity modes. Upper 8 bits configure digital inputs when running in position mode. | | |
| | | | | | Bits | Description | |
| | | | | | 0 | If set, use PWM in signed/magnitude mode. If clear, use PWM in 50% duty cycle offset mode. | |
| | | | | | 1 | If set, invert the PWM input. | |
| | | | | | 2 | If set, invert the sign input. | |
| | | | | | 3 | If set, allow 100% duty cycle.<br>If clear, treat 100% duty cycle as zero command, providing measure of safety in case of controller failure or cable break. | |
| | | | | | 4 | If set, use PWM Input Frequency (0xB6) as deadband for PWM input.<br>Note: Some newer products have dedicated parameter, PWM Input Deadband (0x13F) to hold deadband | |
| | | | | | 5 | If set, allow longer PWM periods (up to 50 ms). | |
| | | | | | 6 | For 8367 DSP products, setting this bit will cause Motor Hall Offset (0x4F) to be added to angle calculated in UV mode. For Plus & AFS products, see UV Configuration (0x180). | |
| | | | | | 7 | Reserved for future use. | |
| | | | | | 8-9 | Input pin interpretation for position mode (see below). | |
| | | | | | | Value | Description |
| | | | | | | 0 | Step & Direction mode. |
| | | | | | | 1 | Separate up & down counters. |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="3">Description</th></tr>
<tr><td rowspan="11"></td><td rowspan="11"></td><td rowspan="11"></td><td rowspan="11"></td><td rowspan="11"></td><td rowspan="2"></td><td>2</td><td>Quadrature encoder input.</td></tr>
<tr><td>3</td><td>PWM input commands absolute position.</td></tr>
<tr><td>10-11</td><td colspan="2">Reserved.</td></tr>
<tr><td>12</td><td colspan="2">If set, pulses are counted on rising edge. If clear, pulses are counted on falling edge. This bit has no effect when inputs are configured as encoder inputs.</td></tr>
<tr><td>13</td><td colspan="2">Causes direction of input to be reversed. Works for all three modes.</td></tr>
<tr><td>14-15</td><td colspan="2">Identify which input pins to use.</td></tr>
<tr><td rowspan="5"></td><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Single ended high-speed inputs.</td></tr>
<tr><td>1</td><td>Multi-mode encoder port.</td></tr>
<tr><td>2</td><td>Differential high-speed inputs.</td></tr>
<tr><td>3</td><td>Use primary encoder inputs.</td></tr>
<tr><td>0xA9</td><td>0x4A9</td><td>0x2321</td><td>RF</td><td>INT32</td><td colspan="3">Digital Input Scaling Factor. This value gives amount of current to command at 100% PWM input. Scaling depends on what PWM input is driving:<br>Current mode: 0.01 A<br>Estimated Velocity: 0.01 RPM<br>Velocity: 0.1 encoder counts/second<br>In position mode scaling factor is a ratio of two 16-bit values. First word passed gives numerator and second word gives denominator. This ratio determines number of encoder units moved for each pulse (or encoder count) input.<br>For example, a ratio of 1/3 would cause motor to move 1 encoder unit for every three input steps.<br>When running in PWM position mode, scaling factor is single 32-bit integer which gives range of commanded position in encoder counts. Minimum PWM Pulse Width (0x13C) corresponds to an absolute position of 0, Maximum PWM Pulse Width (0x13D) corresponds to an absolute position equal to this scaling factor. Additionally, an offset may be added using Registration Offset for Pulse & Direction Mode (0x10F).</td></tr>
<tr><td>0xAA</td><td>0x4AA</td><td>0x2196</td><td>R*</td><td>U16</td><td colspan="3">Raw Input State. 16-bit value returned by this command gives current state (high/low) of drive’s input pins. Unlike Input Pin States (0xA6), no debounce is applied when reading inputs using this variable.<br>Bits are mapped in same order as Input Pin States (0xA6).</td></tr>
<tr><td>0xAB</td><td>0x4AB</td><td>0x2194</td><td>R</td><td>U16</td><td colspan="3">Output States and Program Control. When read, this parameter gives active/inactive state of drive’s general-purpose digital outputs. Each bit represents an input number. Bit 0 = digital output 0 (OUT1), bit 1 = output 1</td></tr>
</table>

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="2">(OUT2), etc., up to output n (OUT(n+1)), number of digital outputs on drive. Additional bits are ignored.<br>Outputs that have not been configured for external register control can be set by writing to this parameter (0x70 - 0x77). Set bit to activate output. It will be activated high or low according to how it was programmed (Bit 8 of 0x70-0x77). Clear bit to make output inactive. If an output was configured for internal register control, it will not be affected.</td></tr>
<tr><td>0xAC</td><td>0x4AC</td><td>0x2180</td><td>R*</td><td>U32</td><td colspan="2">Sticky Drive Event Status Register. This read-only parameter is bit-mapped in exactly same way as Event Status Register (0xA0), but instead of giving present status of drive, sticky version indicates any bits in event status that have been set since last reading of sticky register.<br>Sticky register is similar to Latched Event Status Register (0xA1), but latched register must be cleared explicitly, whereas sticky register is cleared automatically each time it is read.</td></tr>
<tr><td rowspan="25">0xAD</td><td rowspan="25">0x4AD</td><td rowspan="25">0x1018:2<br>or<br>0x2384:13</td><td rowspan="25">F*</td><td rowspan="25">INT16</td><td colspan="2">Drive Hardware Type. Also known as Product Code. Identifies specific drive model.<br>This is an augmented version of Product Family (0x87).</td></tr>
<tr><td>Value</td><td>Product</td></tr>
<tr><td>0x0000</td><td>ASC Accelus Card.</td></tr>
<tr><td>0x0001</td><td>ASP Accelus panel without pullup/pulldown on inputs</td></tr>
<tr><td>0x0002</td><td>ASP Accelus panel with pullup/pulldown on input pins</td></tr>
<tr><td>0x0100</td><td>JSP Junus Panel</td></tr>
<tr><td>0x0200</td><td>ACM Accelnet Module</td></tr>
<tr><td>0x0201</td><td>XSL Xenus Panel (Legacy)</td></tr>
<tr><td>0x0209</td><td>ACJ Accelnet Micro Panel</td></tr>
<tr><td>0x0210</td><td>ACJ-S Accelnet Micro Panel Sin/Cos</td></tr>
<tr><td>0x020C</td><td>ACK Accelnet Micro Module</td></tr>
<tr><td>0x0240</td><td>STM Stepnet Module</td></tr>
<tr><td>0x0242</td><td>STP Stepnet Panel</td></tr>
<tr><td>0x0243</td><td>STL Stepnet Micro Module</td></tr>
<tr><td>0x0300</td><td>ASP-X2 2-axis Accelus panel (Obsolete)</td></tr>
<tr><td>0x0310</td><td>XSJ Xenus Micro (Obsolete)</td></tr>
<tr><td>0x0314</td><td>XSJ Rev 01 Xenus Micro AFS</td></tr>
<tr><td>0x0320</td><td>XTL-R Xenus Resolver (Obsolete)</td></tr>
<tr><td>0x0324</td><td>XTL-R Rev 01 Xenus Resolver AFS</td></tr>
<tr><td>0x0330</td><td>XTL Xenus (Obsolete)</td></tr>
<tr><td>0x0334</td><td>XTL Rev 01 Xenus AFS</td></tr>
<tr><td>0x0340</td><td>XSJ-R Xenus Micro Resolver (Obsolete)</td></tr>
<tr><td>0x0344</td><td>XSJ-R Rev 01 Xenus Micro Resolver AFS</td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 0x0350 | STX Stepnet AC (Obsolete) |
| | | | | | 0x0354 | STX Rev 01 Stepnet AC AFS |
| | | | | | 0x0360 | ACJ-R Accelnet Micro Panel Resolver (Obsolete) |
| | | | | | 0x0370 | ACK-R Accelnet Micro Module Resolver (Obsolete) |
| | | | | | 0x0380 | AEP Accelnet EtherCAT Panel (Obsolete) |
| | | | | | 0x0390 | AMP Accelnet Macro Panel (Obsolete) |
| | | | | | 0x03A0 | ADP Accelnet Panel (Obsolete) |
| | | | | | 0x03A4 | ADP Rev 01 Accelnet Panel AFS |
| | | | | | 0x03B0 | ST3 3-axis Stepnet (Obsolete) |
| | | | | | 0x03C0 | 800-1638 Custom drive (Obsolete) |
| | | | | | 0x03D0 | ADP-R Accelnet Panel Resolver (Obsolete) |
| | | | | | 0x03E0 | ACM-R Accelnet Module (Obsolete) |
| | | | | | 0x03F0 | ACK Accelnet Micro Module AFS |
| | | | | | 0x0404 | CAN I/O Rev 01 Module |
| | | | | | 0x1000 | XEL Xenus Plus EtherCAT (Obsolete) |
| | | | | | 0x1001 | XEL Xenus Plus EtherCAT |
| | | | | | 0x1008 | XEL-R Xenus Plus EtherCAT Resolver |
| | | | | | 0x1010 | XML Xenus Plus MACRO |
| | | | | | 0x1020 | XPL Xenus Plus CAN |
| | | | | | 0x1030 | AEM Accelnet Plus EtherCAT Module |
| | | | | | 0x1040 | APM Accelnet Plus CAN module |
| | | | | | 0x1050 | AE2 2-axis Accelnet Plus EtherCAT module |
| | | | | | 0x1060 | AP2 2-axis Accelnet Plus CAN module |
| | | | | | 0x1070 | SEM Stepnet Plus EtherCAT module |
| | | | | | 0x1080 | SPM Stepnet Plus CAN module |
| | | | | | 0x1090 | SE2 2-axis Stepnet Plus EtherCAT module |
| | | | | | 0x10A0 | SP2 2-axis Stepnet Plus CAN module |
| | | | | | 0x10B0 | XE2 2-axis Xenus Plus EtherCAT |
| | | | | | 0x10C0 | BE2 2-axis Accelnet Plus EtherCAT Panel |
| | | | | | 0x10D0 | XP2 2-axis Xenus Plus CAN |
| | | | | | 0x10E0 | BP2 2-axis Accelnet Plus EtherCAT Panel |
| | | | | | 0x10F0 | TE2 2-axis Stepnet Plus EtherCAT Panel |
| | | | | | 0x1100 | TP2 2-axis Stepnet Plus CAN Panel |
| | | | | | 0x1110 | BEL Accelnet Plus EtherCAT Panel |
| | | | | | 0x1120 | BPL Accelnet Plus CAN Panel |
| | | | | | 0x1130 | TEL Stepnet Plus EtherCAT Panel |
| | | | | | 0x1150 | SP4 4-axis Stepnet CAN Module |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 0x1170 | XM2 2-axis Xenus Plus MACRO |
| | | | | | 0x1180 | BML Accelnet Plus MACRO |
| | | | | | 0x1190 | SE4 4-axis Stepnet EtherCAT Module |
| | | | | | 0x11B0 | XEC Xenus Plus Compact EtherCAT |
| | | | | | 0x11C0 | XPC Xenus Plus Compact CAN |
| | | | | | 0x11D0 | ME3 3-axis Module EtherCAT |
| | | | | | 0x11E0 | MP3 3-axis Module CANopen |
| | | | | | 0x11F0 | ME4 4-axis Module EtherCAT |
| | | | | | 0x1200 | MP4 4-axis Module CANopen |
| | | | | | 0x1240 | GEM Argus Plus EtherCAT Module |
| | | | | | 0x1248 | GEM Argus Plus EtherCAT Resolver |
| | | | | | 0x1250 | GPM Argus Plus CAN Module |
| | | | | | 0x1258 | GPM-R Argus Plus CAN Resolver |
| | | | | | 0x1260 | AEV Accelnet Plus Micro EtherCAT Module |
| | | | | | 0x1270 | APV Accelnet Plus Micro CAN Module |
| | | | | | 0x1280 | AEZ Panel |
| | | | | | 0x1290 | APZ Panel |
| | | | | | 0x2050 | IES Integrated Servo Drive |
| 0xAE | 0x4AE | 0x60F6:3 | RF | INT16 | Current Loop Offset. Units: 0.01 A.<br>This value is added to commanded motor current.<br>It can compensate for directional bias affecting current loop. | |
| 0xAF | 0x4AF | 0x2420 | RF | INT32 | Miscellaneous Drive Options Register.<br>Bit-mapped as follows: | |
| | | | | | Bits | Option |
| | | | | | 0 | If set, input pins 1, 2 and 3 are pulled high on drive.<br>If clear, pins are not pulled up.<br>Only available on Junus drive. |
| | | | | | 1 | Reserved. |
| | | | | | 2 | If set, limit switch inputs will only abort trajectory in progress but will not affect current output.<br>If clear, limit switches limit current. |
| | | | | | 3 | If set, save PDO configuration to file in CVM file system when “Save to Flash” command is received over CANopen network. If clear, PDO is not saved. |
| | | | | | 4 | If set, limit switch activation will be treated as fault in CANopen Status Word (CANopen index 0x6041 as described in *CANopen Programmer’s Manual*). |
| | | | | | 5-6 | When encoder wrap is enabled, these bits control direction of motion for absolute moves in |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | | trapezoidal and<br>S-curve profile modes.<br>0 – move in the shortest direction<br>1 – Always move in positive direction<br>2 – Always move in negative direction<br>3 – reserved. |
| | | | | | 7 | If set, analog command values will use digital data written to an SPI serial interface connected to drive input pins & multimode port. This is available on some plus drives for use in digitally interfacing with a Delta Tau controller. |
| | | | | | 8 | If set, brake delay will be applied even in case of latching faults. |
| | | | | | 9-31 | Reserved |
| 0xB0 | 0x4B0 | 0x2260 | R | INT16 | Motor Phase Angle. Units: degrees.<br>Writes are only useful when running in diagnostic micro-stepping mode. | |
| 0xB1 | 0x4B1 | 0x21C1 | RF | INT16 | Increment Rate for Phase Angle When In Micro Stepping Mode. Units: degrees/s.<br>Only used in diagnostic mode.<br>Desired State (0x24) = 42 (microstepping mode). | |
| 0xB2 | 0x4B2 | 0x21C0 | RF | U16 | Commutation Mode. Also known as Phasing Mode. Controls mechanism used by drive to compute motor phasing angle. Determines what inputs drive uses to initialize and maintain phase angle.<br>Bit-mapped as follows: | |
| | | | | | Value | Mode |
| | | | | | 0 | Standard mode. Encoder-based sinusoidal commutation for brushless motors. Use digital Hall inputs to initialize phase, then switch to an encoder to maintain phase. Encoder is primary sensing device with Hall Effect sensors used to monitor and adjust phase angle as necessary during operation. |
| | | | | | 1 | Trapezoidal (Hall based) phasing.<br>Hall Effect sensors are used for phasing at all times. This mode can be used if no encoder is available. |
| | | | | | 2 | Like mode 0 except that phase angle is not adjusted based on Hall inputs. Hall Effect sensors are still required to initialize phase angle at startup. |
| | | | | | 3 | Analog Halls (90 degrees).<br>Only available on drives with necessary analog Hall inputs. |
| | | | | | 4 | DC brush motor mode. Note preferred way to configure an axis to drive DC brushed motor is by setting Motor Type (0x40). This method will |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | | continue to be supported for backward compatibility however. |
| | | | | | 5 | Algorithmic Phase Initialization mode (wake & wiggle, no Halls).<br>See *CME User Guide* for more information on Algorithmic Phase Initialization. |
| | | | | | 6 | Encoder based phasing. Use with resolver or Servo-Tube motors. |
| | | | | | 7 | Trapezoidal commutation with phase angle interpolation. |
| 0xB3 | 0x4B3 | 0x2384:23 | F* | INT16 | Analog Encoder Scaling Factor. This parameter selects resolution of analog encoder input. Parameter not used for other encoder types. | |
| 0xB4 | 0x4B4 | 0x2263 | R* | INT16 | Encoder Phase Angle. For feedback types, such as resolver, that can also calculate phase angle information. This parameter allows phase information to be read directly. | |
| 0xB5 | 0x4B5 | 0x2353 | R* | INT32 | Homing Adjustment. Units: counts.<br>This parameter is updated after each successful homing operation. Value contained is size of actual position adjustment made in last home sequence. | |
| 0xB6 | 0x4B6 | 0x2322 | RF | U16 | PWM Input Frequency. This is frequency of PWM for use in UV commutation mode only. Units: 10 Hz. This parameter is also used to specify an optional PWM dead band when running in normal (not UV) PWM command modes. When used as deadband value, this input should be set in range 0 to 32767 which corresponds to deadband of 0 to 100% of PWM duty cycle.<br>On Plus and AFS models, PWM Input Deadband (0x13F) is dedicated to holding PWM Input Deadband value. On products supporting that parameter, writing to this parameter will still modify deadband setting for backward compatibility but use of PWM Input Deadband (0x13F) is recommended. | |
| 0xB7 | 0x4B7 | 0x2141 | R* | U32 | System Time. Time since start up. Units: ms. | |
| 0xB8 | 0x4B8 | 0x607D:2 | RF | INT32 | Positive Software Limit value. Units: counts.<br>This parameter is only available on drives that support trajectory generation and homing.<br>Software limits are only in effect after drive has been referenced (i.e. homing has been successfully completed).<br>Set to less than negative software limit to disable. | |
| 0xB9 | 0x4B9 | 0x607D:1 | RF | INT32 | Negative Software Limit. Units: counts.<br>Software limits are only in effect after drive has been referenced (i.e. homing has been successfully completed).<br>Set to greater than positive software limit to disable. | |
| 0xBA | 0x4BA | 0x2120 | RF | INT32 | Following Error Fault Limit. Units: counts.<br>If Position Loop Error (0x35) exceeds this value then *following error (bit 18)* of Event Status Register (0xA0) is set and motor is stopped. Using Fault Mask (0xA7), following error event can be configured to either disable drive | |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>immediately or abort present move and continue holding position.</td></tr>
<tr><td>0xBB</td><td>0x4BB</td><td>0x6065</td><td>RF</td><td>INT32</td><td>Following Error Warning Limit. Units: counts.<br>If Position Loop Error (0x35) exceeds this value then following warning (bit 19) of Event Status Register (0xA0) is set.</td></tr>
<tr><td>0xBC</td><td>0x4BC</td><td>0x6067</td><td>RF</td><td>INT32</td><td>Position Tracking Window Limit. Units: counts.<br>If Position Loop Error (0x35) exceeds this value then tracking window (bit 25) of Event Status Register (0xA0) is set.</td></tr>
<tr><td>0xBD</td><td>0x4BD</td><td>0x6068</td><td>RF</td><td>U16</td><td>Time Delay For Following Error Fault Limit (0xBA). Units: ms.<br>Tracking window (bit 25) of Event Status Register (0xA0) will not be cleared until Position Loop Error (0x35) has been within Following Error Fault Limit (0xBA) for at least this long.</td></tr>
<tr><td>0xBE</td><td>0x4BE</td><td>0x2253</td><td>RF</td><td>U32</td><td>Deceleration limit used with software limits. Set to 0 for non-trajectory based software limits.</td></tr>
<tr><td>0xBF</td><td>0x4BF</td><td>0x2351</td><td>RF</td><td>U16</td><td>Home to Hard Stop Delay Time. Units: ms.<br>When performing home to hard stop, drive will push against stop for this long before sampling home position.</td></tr>
<tr><td>0xC0</td><td>0x4C0</td><td>None</td><td>R*</td><td>INT16</td><td>CAN Network Node ID. This is drive’s present ID as read at system startup. Node ID is only read at system startup, so this value will not change unless drive is reset. See CAN Network Node ID Configuration (0xC1).</td></tr>
<tr><td>0xC1</td><td>0x4C1</td><td>0x21B0</td><td>RF</td><td>INT16</td><td>CAN Network Node ID Configuration. Defines how drive’s Node ID is calculated and specifies drive’s network bit rate. Node ID is calculated at startup (and only at startup) using a combination of general-purpose input pins and programmed offset value. On certain models, an address switch is also used. Resulting value is clipped to a 7-bit ID in range 0 to 127.<br>For EtherCAT, this parameter can optionally hold network alias value to be loaded into ESC at power-up. See Network Options (0x121) for details.<br>Plus drives with firmware 2.82 or greater have an optional new method of setting Node IDs on multi-axis drives. This new method allows each axis to be assigned its own ID, and Node IDs don't have to be consecutive. See descriptions of parameters Input Pin Mapping, Node ID Selection (0x103) and Network Options (0x121) for details of this new method.<br>For multi-axis CANopen drives, first axis Node ID is set using this parameter. Subsequent axes are assigned consecutive Node ID's. For example, if first axis were given Node ID 7 using this parameter, second would be Node 8, third would be Node 9, etc.<br>Bit-mapped as follows:
<table>
<tr><td>Bits</td><td>Description</td></tr>
<tr><td>0-6</td><td>Give Node ID offset value that will be added to value read from input pins</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="3">Description</th></tr>
<tr><td rowspan="19"></td><td rowspan="19"></td><td rowspan="19"></td><td rowspan="19"></td><td rowspan="19"></td><td>7</td><td colspan="2">Used only on DeviceNet firmware.<br>If set, drive will be software disabled on startup and will remain disabled until enabled by DeviceNet I/O message with enable bit set.</td></tr>
<tr><td>8-10</td><td colspan="2">Number of input pins (0-7) to read on startup for Node ID value. If input pins are used (i.e., value in bits 8-10 is not zero), inputs can be mapped to Node ID bits through Input Pin Mapping, Node ID Selection (0x103).</td></tr>
<tr><td>11</td><td colspan="2">This bit is ignored on drives that do not have an address switch.<br>On drives with an address switch, setting this bit programs drive to use address selector switch as part of address calculation. In this case, Node ID value is equal to sum of:<br>▪ Value read from designated input pins, shifted up 4 bits.<br>▪ Address switch value.<br>▪ Programmed offset value.<br>Note that since Node ID is always clipped to lowest 7 bits, no more than 3 input pins will ever have an effect on Node address when address switch is used.</td></tr>
<tr><td>12-15</td><td colspan="2">Network bit rate setting:</td></tr>
<tr><td></td><td>Value</td><td>Bit Rate (bits/s)</td></tr>
<tr><td></td><td>0</td><td>1,000,000.</td></tr>
<tr><td></td><td>1</td><td>800,000.</td></tr>
<tr><td></td><td>2</td><td>500,000.</td></tr>
<tr><td></td><td>3</td><td>250,000.</td></tr>
<tr><td></td><td>4</td><td>125,000.</td></tr>
<tr><td></td><td>5</td><td>50,000.</td></tr>
<tr><td></td><td>6</td><td>20,000.</td></tr>
<tr><td></td><td>7-15</td><td>Reserved</td></tr>
<tr><td rowspan="6">0xC2</td><td rowspan="6">0x4C2</td><td rowspan="6">0x2352</td><td rowspan="6">RF</td><td rowspan="6">INT16</td><td colspan="3">Homing Method Configuration. Bit-mapped as follows:</td></tr>
<tr><td>Bits</td><td colspan="2">Description</td></tr>
<tr><td>0-3</td><td colspan="2">Home function.</td></tr>
<tr><td></td><td>Value</td><td>Description</td></tr>
<tr><td></td><td>0</td><td>If bit 5 is not set, then just set current position as home.<br>If bit 5 is set, then move in direction specified by bit 4 and set location of first index pulse as home.<br>Bit 6 not used in this mode.</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| | | | | | | 1 | Move in direction specified by bit 4 until limit switch is encountered. Then move in other direction out of limit.<br>If bit 5 is clear, then edge location is home.<br>If bit 5 is set, then next index pulse is home.<br>Bit 6 not used in this mode. |
| | | | | | | 2 | Home on constant home switch. Initial move is made in direction specified by bit 4. When home switch is encountered, direction is reversed. If bit 5 is clear, edge of home switch is set as home. If bit 5 is set, then an index pulse is used as home position. Bit 6 is used to define which index pulse is used. |
| | | | | | | 3 | Home on intermittent home switch. This mode works same as mode 2 except that if limit switch is encountered when initially searching for home, then direction is reversed. In mode 2, hitting limit switch before finding home would be considered an error.<br>Bit 8 identifies which edge of home to search for (positive or negative). |
| | | | | | | 4 | Home to a hard stop. This moves in the direction specified in bit 4 until home current limit is reached. It then presses against hard stop using that current value until home delay time expires.<br>If bit 5 (index) is set, drive away from the hard stop until an index is found. |
| | | | | | | 5-14 | Reserved for future |
| | | | | | | 15 | Immediate home. This value causes the amp to be referenced immediately on power-up. Once encoder is initialized, home offset value is added to encoder position and result is set as current referenced position. This is primarily useful with absolute encoders. |
| | | | | | 4 | Initial move direction (0=positive, 1=negative). | |
| | | | | | 5 | Home on index pulse if set. | |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 6 | Selects which index pulse to use.<br>If set, use pulse on DIR side of sensor edge.<br>DIR is direction specified by bit 4 of this word. |
| | | | | | 7 | If set, capture falling edge of index. If clear, capture rising edge. |
| | | | | | 8 | When using momentary home switch, this bit identifies which edge of home switch to reference on.<br>If set, use negative edge.<br>If clear, use positive edge. |
| | | | | | 9 | If set, move to zero position when homing is finished. If clear, zero position is found, but not moved to. |
| | | | | | 10 | If set, homing sequence will run as normal, but actual position will not be adjusted at end of homing. Note that even though actual position is not adjusted, Homing Adjustment (0xB5) is updated with size of adjustment (in counts) that would have been made.<br>Also, if bit 10 is set then no move to zero is made regardless of setting of bit 9. |
| | | | | | 11 | If this bit is set, at end of home routine home configuration stored in flash will be set to 15, and home offset stored in flash will be updated to correct value necessary to calibrate an absolute encoder based on most recent home operation. This bit is used to automate calibration of absolute encoders. |
| 0xC3 | 0x4C3 | 0x6099:1 | RF | INT32 | Homing Velocity (fast moves). Units: 0.1 counts/s.<br>This velocity value is used during segments of homing procedure that may be handled at high speed. Generally, this means moves in which home sensor is being located, but edge of sensor is not being found. | |
| 0xC4 | 0x4C4 | 0x6099:2 | RF | INT32 | Homing Velocity (slow moves). Units: 0.1 counts/s.<br>This velocity value is used for homing segments that require low speed, such as cases where edge of a homing sensor is being sought. | |
| 0xC5 | 0x4C5 | 0x609A | RF | U32 | Homing Acceleration/Deceleration.<br>Units: 10 counts/s$^2$.<br>This value defines acceleration used for all homing moves. Same value is used at beginning and ending of moves (i.e. no separate deceleration value). | |
| 0xC6 | 0x4C6 | 0x607C | RF | INT32 | Home Offset. Units: counts.<br>Home offset is difference between zero position for application and machine home position (found during homing). Once homing is completed, new zero position determined by homing state machine will be located sensor position plus this offset. All subsequent absolute moves shall be taken relative to this new zero position. | |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0xC7 | 0x4C7 | 0x2350 | RF | INT16 | Homing Current Limit. Units: 0.01 A.<br>Used in Home to Hard Stop mode only, this current is used to determine when drive has reached end of travel (hard stop). Used in conjunction with Home to Hard Stop Delay Time (0xBF). |
| 0xC8 | 0x4C8 | None | RF | INT16 | Trajectory Profile Mode. To set profile in CANopen see CAN object 0x6086 in *CANopen Programmers Manual*. Bit-mapped as follows: |
| | | | | | Bits: Description |
| | | | | | 0-2: Give trajectory profile mode. Possible trajectory modes are described below. |
| | | | | | Value: Description |
| | | | | | 0: Trapezoidal profile mode. Uses position/distance, velocity, acceleration and deceleration. Any parameters may be changed during move. Jerk is not used in this mode. |
| | | | | | 1: S-curve profile mode. Uses position/distance, velocity, acceleration, and jerk. No parameters may be changed while move is in progress (although move may be aborted). Acceleration parameter will be used for deceleration. |
| | | | | | 2: Velocity mode. Uses velocity, acceleration, and deceleration. Jerk is not used in this mode, and position is only used to define direction of move (zero or positive to move with a positive velocity, negative to move with a negative velocity). Any parameter may be changed during move. Set velocity to zero to stop. |
| | | | | | 3: PVT profile mode. Use of this mode through serial interface is not presently supported. |
| | | | | | 8: If set, relative move.<br>If clear, absolute move. |
| 0xC9 | 0x4C9 | 0x2252 | R* | INT16 | Trajectory Status Register. This parameter gives status information about the trajectory generator. Bit-mapped as follows: |
| | | | | | Bits: Description |
| | | | | | 0-8: Reserved. |
| | | | | | 9: Cam table underflow. |
| | | | | | 10: Reserved. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 11 | Homing error. If set, an error occurred in last home attempt. Cleared by a home command. |
| | | | | | 12 | Referenced. Set when homing command has been successfully executed. Cleared by home command. |
| | | | | | 13 | Homing. If set, drive is running home command. |
| | | | | | 14 | Set when move is aborted.<br>Cleared at start of next move. |
| | | | | | 15 | In-Motion Bit. If set, trajectory generator is presently generating profile. |
| 0xCA | 0x4CA | 0x607A | RF | INT32 | Trajectory Generator Position Command.<br>Units: Counts.<br>This value gives destination position for absolute moves or move distance for relative moves. | |
| | | | | | Type | Meaning |
| | | | | | Relative | Move distance. |
| | | | | | Absolute | Target position. |
| | | | | | Velocity | Direction: 1 for positive, -1 for negative. |
| 0xCB | 0x4CB | 0x6081 | RF | INT32 | Trajectory Maximum Velocity. Trajectory generator will attempt to reach this velocity during a move. Units: 0.1 counts/s. | |
| 0xCC | 0x4CC | 0x6083 | RF | U32 | Trajectory Maximum Acceleration.<br>Units: 10 counts/$s^2$.<br>Trajectory generator will attempt to reach this acceleration during a move. For s-curve profiles,<br>this value also used to decelerate at end of move. | |
| 0xCD | 0x4CD | 0x6084 | RF | U32 | Trajectory Maximum Deceleration.<br>Units: 10 counts/$s^2$.<br>In trapezoidal trajectory mode, this value used to decelerate at end of move. | |
| 0xCE | 0x4CE | 0x2121 | RF | U32 | Trajectory Maximum Jerk. Units: 100 counts/$s^3$.<br>Also known as Trajectory Jerk Limit. S-curve profile generator uses this value as jerk (rate of change of acceleration/deceleration) during moves.<br>Other profiles types do not use jerk limit. | |
| 0xCF | 0x4CF | 0x6085 | RF | U32 | Trajectory Abort Deceleration. Units: 10 counts/$s^2$.<br>If move is aborted, this value will be used by trajectory generator to decelerate to stop. | |
| 0xD0 | 0x4D0 | 0x2192:9 | RF | U16 | Input 8 Configuration. See Input 0 Configuration (0x78). | |
| 0xD1 | 0x4D1 | 0x2192:10 | RF | U16 | Input 9 Configuration. See Input 0 Configuration (0x78). | |
| 0xD2 | 0x4D2 | 0x2192:11 | RF | U16 | Input 10 Configuration. See Input 0 Configuration (0x78). | |
| 0xD3 | 0x4D3 | 0x2192:12 | RF | U16 | Input 11 Configuration. See Input 0 Configuration (0x78). | |
| 0xD4 | 0x4D4 | 0x2192:13 | RF | U16 | Input 12 Configuration. See Input 0 Configuration (0x78). | |
| 0xD5 | 0x4D5 | 0x2192:14 | RF | U16 | Input 13 Configuration. See Input 0 Configuration (0x78). | |
| 0xD6 | 0x4D6 | 0x2192:15 | RF | U16 | Input 14 Configuration. See Input 0 Configuration (0x78). | |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0xD7 | 0x4D7 | 0x2192:16 | RF | U16 | Input 15 Configuration. See Input 0 Configuration (0x78). |
| 0xD8 | 0x4D8 | 0x2150 | RF | U16 | Regen Resistor Resistance. Units: 0.1 Ω. |
| 0xD9 | 0x4D9 | 0x2151 | RF | U16 | Regen Resistor, Continuous Power. Units: W. |
| 0xDA | 0x4DA | 0x2152 | RF | U16 | Regen Resistor, Peak Power. Units: W. |
| 0xDB | 0x4DB | 0x2153 | RF | U16 | Regen Resistor, Time at Peak. Units: ms. |
| 0xDC | 0x4DC | 0x2154 | RF | INT16 | Regen Turn on Voltage Units: 0.1 V. |
| 0xDD | 0x4DD | 0x2155 | RF | INT16 | Regen Turn off Voltage. Units: 0.1 V. |
| 0xDE | 0x4DE | 0x2384:20 | F* | INT16 | Drive's Peak Current Rating for Internal Regen Transistor. Units: 0.01 A. |
| 0xDF | 0x4DF | 0x2384:21 | F* | INT16 | Drive's Continuous Current Rating for Internal Regen Transistor. Units: 0.01 A. |
| 0xE0 | 0x4E0 | 0x2384:22 | F* | INT16 | Drive's Time at Peak Current for Internal Regen Transistor. Units: ms. |
| 0xE1 | 0x4E1 | 0x2156 | F | String | Regen Resistor Model Number String. |
| 0xE2 | 0x4E2 | 0x2157 | R* | INT16 | Regen Resistor Status. Bit-mapped as follows: |
| | | | | | Bits: Description |
| | | | | | 0: Set if regen circuit is currently closed. |
| | | | | | 1: Set if regen is required based on bus voltage. |
| | | | | | 2: Set if regen circuit is open due to an overload condition. Overload may be caused by either resistor settings or internal drive protections. |
| | | | | | 3-15: Reserved. |
| 0xE3 | 0x4E3 | 0x2382 | RF | U16 | Position Loop Output Gain Multiplier. Output of position loop is multiplied by this value before being passed to velocity loop. This scaling factor is calculated such that a value of 100 is a 1.0 scaling factor.<br>This parameter is most useful in dual loop systems. |
| 0xE4 | 0x4E4 | 0x21C2 | RF | INT16 | Maximum Current to use with algorithmic phase initialization. See *Value 5* of Commutation Mode (0xB2). Units: 0.01 A. |
| 0xE5 | 0x4E5 | 0x21C3 | RF | U16 | Algorithmic Phase Initialization Timeout. See *Value 5* of Commutation Mode (0xB2). Units: ms. |
| 0xE6 | 0x4E6 | 0x21D8 | RF | INT32 | Max Step Rate. This is maximum velocity adjustment made by stepper outer position loop when enabled. This parameter is only used when stepper outer loop is engaged (*bit 1* of Stepper Configuration & Status (0xEE) is set). Units: 0.1 steps/s. |
| 0xE7 | 0x4E7 | 0x21D7 | RF | U16 | Proportional Gain (ECp) for Stepper Outer Loop. This parameter gives gain used for calculating velocity adjustment based on Position Loop Error (0x35). This parameter is only used when stepper outer loop is engaged (*bit 1* of Stepper Configuration & Status (0xEE) is set). |
| 0xE8 | 0x4E8 | 0x21D0 | RF | INT16 | Holding Current for Microstepping Mode. Units: 0.01 A. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0xE9 | 0x4E9 | 0x21D1 | RF | U16 | Run to Hold Time for Microstepping Mode. Units: ms. |
| 0xEA | 0x4EA | 0x21D2 | RF | U16 | Detent Correction Gain Factor for Microstepping mode. |
| 0xEB | 0x4EB | 0x21D3 | RF | U16 | Damping Correction Gain Factor for Microstepping mode |
| 0xEC | 0x4EC | 0x21D4 | RF | 9 or 14 | Damping Correction biquad filter structure for Microstepping mode.<br>For details on encoding of filter structure, please see Filter Coefficients. |
| 0xED | 0x4ED | 0x21D5 | RF | U16 | Holding Current to Fixed Voltage Output Time for Microstepping Mode. Time delay from entering hold current before entering special voltage control mode of operation. This mode trades normal tight control of current for very low jitter on motor position.<br>Used in stepper mode only. Set to 0 to disable this feature. Units: ms. |
| 0xEE | 0x4EE | 0x21D6 | RF | INT16 | Stepper Configuration & Status. Bit-mapped as follows: |
| | | | | | Bits: Description |
| | | | | | 0: Use encoder input for phase compensation if enabled.<br>Pure stepper mode if disabled. |
| | | | | | 1: Use outer position loop to adjust stepper position based on Position Loop Error (0x35). When this bit is set, gain value Proportional Gain (ECp) (0xE7) is multiplied by Position Loop Error (0x35) and result is velocity that is added to Microstepping position limited by Max Step Rate (0xE6). |
| | | | | | 2-15: Reserved. |
| 0xF0 | 0x4F0 | 0x2195:1 | RF | U16 | Debounce Time For Input 0. Units: ms. |
| 0xF1 | 0x4F1 | 0x2195:2 | RF | U16 | Debounce Time For Input 1. Units: ms. |
| 0xF2 | 0x4F2 | 0x2195:3 | RF | U16 | Debounce Time For Input 2. Units: ms. |
| 0xF3 | 0x4F3 | 0x2195:4 | RF | U16 | Debounce Time For Input 3. Units: ms. |
| 0xF4 | 0x4F4 | 0x2195:5 | RF | U16 | Debounce Time For Input 4. Units: ms. |
| 0xF5 | 0x4F5 | 0x2195:6 | RF | U16 | Debounce Time For Input 5. Units: ms. |
| 0xF6 | 0x4F6 | 0x2195:7 | RF | U16 | Debounce Time For Input 6. Units: ms. |
| 0xF7 | 0x4F7 | 0x2195:8 | RF | U16 | Debounce Time For Input 7. Units: ms. |
| 0xF8 | 0x4F8 | 0x2195:9 | RF | U16 | Debounce Time For Input 8. Units: ms. |
| 0xF9 | 0x4F9 | 0x2195:10 | RF | U16 | Debounce Time For Input 9. Units: ms. |
| 0xFA | 0x4FA | 0x2195:11 | RF | U16 | Debounce Time For Input 10. Units: ms. |
| 0xFB | 0x4FB | 0x2195:12 | RF | U16 | Debounce Time For Input 11. Units: ms. |
| 0xFC | 0x4FC | 0x2195:13 | RF | U16 | Debounce Time For Input 12. Units: ms. |
| 0xFD | 0x4FD | 0x2195:14 | RF | U16 | Debounce Time For Input 13. Units: ms. |
| 0xFE | 0x4FE | 0x2195:15 | RF | U16 | Debounce Time For Input 14. Units: ms. |
| 0xFF | 0x4FF | 0x2195:16 | RF | U16 | Debounce Time For Input 15. Units: ms. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0x100 | 0x500 | 0x2184 | RF | U32 | CANopen Limit Status Mask. This parameter defines which bits in Event Status Register (0xA0) can set limit bit (bit 11) of CANopen Status Word (CANopen index 0x6041 as described in *CANopen Programmer's Manual*). If Event Register Status (0xA0) and its corresponding Limit Mask bit are both set, then CANopen Status Word limit bit is set. If all selected Event Status Register (0xA0) bits are clear, then limit bit is clear. |
| 0x101 | 0x501 | 0x2197 | R* | INT16 | Network Address Switch Value. This gives current state of address switch. For drives without a switch, value returned is undefined. |
| 0x102 | 0x502 | 0x21B4 | R* | INT16 | Network Status Word. Bit-mapped as follows: |

| CANopen | | |
|---|---|---|
| Bits | Meaning | |
| 0-1 | CANopen node status. This field will take one of following values: | |
| | Value | Status |
| | 0 | CANopen interface is disabled. |
| | 1 | Stopped mode. |
| | 2 | Preoperational mode. |
| | 3 | Operational mode. |
| 4 | Set if CANopen SYNC message is missing. | |
| 5 | Set on CANopen guard error. | |
| 8 | Set if CAN port is in 'bus off' state. | |
| 9 | Set if CAN port is in 'transmit error passive' state. | |
| 10 | Set if CAN port is in 'receive error passive' state. | |
| 11 | Set if CAN port is in 'transmit warning' state. | |
| 12 | Set if CAN port is in 'receive warning' state. | |
| 15 | Always clear for CANopen | |

| DeviceNet | |
|---|---|
| Bits | Meaning |
| 0 | Set if duplicate MAC ID check failed. |
| 1 | Set if device is online. |
| 2 | Set if at least one communication object timed out. |
| 3 | Set if at least one communication object has been established. |
| 4-7 | Reserved. |
| 8-14 | Same bit mapping as for CANopen. |
| 15 | Always set for DeviceNet. |

| EtherCAT |
|---|

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 0 | Set if distributed clock is enabled |
| | | | | | 1 | Set if distributed clock is locked |
| | | | | | 2 | Set if SYNC0 period is multiple of drive's servo period |
| | | | | | 3 | Set if invalid SYNC0 time |
| | | | | | 4-15 | Reserved for future use. |
| | | | | | MACRO | |
| | | | | | 0 | Set if MACRO network is detected. |
| | | | | | 1 | Set if drive is being disabled by MACRO master. |
| | | | | | 2 | Set if MACRO network has been broken (i.e. once detected but now gone). |
| | | | | | 3 | Set on heartbeat error. |
| | | | | | 4 | Ring break error received from upstream device. |
| | | | | | 5-15 | Reserved. |
| 0x103 | 0x503 | 0x21B1 | F | U32 | Input Pin Mapping for Node ID Selection.<br>When CAN Network Node ID Configuration (0xC1) indicates that 1 or more input pins will be used to select Node ID, this parameter is used to map input pins to ID bits. | |
| | | | | | Bits | Meaning |
| | | | | | 0-3 | Identify the general-purpose input pin associated with ID bit 0. |
| | | | | | 4-7 | Identify the general-purpose input pin associated with ID bit 1. |
| | | | | | 8-11 | Identify the general-purpose input pin associated with ID bit 2. |
| | | | | | 12-15 | Identify the general-purpose input pin associated with ID bit 3. |
| | | | | | 16-19 | Identify the general-purpose input pin associated with ID bit 4. |
| | | | | | 20-23 | Identify the general-purpose input pin associated with ID bit 5. |
| | | | | | 24-27 | Identify the general-purpose input pin associated with ID bit 6. |
| | | | | | 28-30 | Reserved. |
| | | | | | 31 | Set to enable this register. Clear to use default mapping. |
| | | | | | If bit 31 is zero, then default bit mapping is used, and rest of this register is ignored. Default bit mapping uses top N input pins and maps them such that high numbered pins are used for higher numbered bits in Node ID. For example; Accelnet Panel drive has 12 general-purpose input pins (0 to 11). If 3 of these pins are used for Node ID configuration and default mapping is used, then highest 3 pins (9, 10 and 11) will be | |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>used for Node ID. In this case, pin 9 will be bit 0, pin 10 will be bit 1 and pin 11 will be bit 2.<br><br>If bit 31 is set, then rest of this register will be used to define which input pin will be assigned to which bit of Node ID. Input pins are numbered from 0 to 15 and each nibble of register gives input pin number associated with one bit of Node ID.<br><br>For example, if three input pins are configured for address selection and the mapping register is set to 0x80000012, then input pin 2 will be used for Node ID bit 0, input pin 1 will be used for Node ID bit 1 and input pin 0 will be used for Node ID bit 2.<br><br>Note that CAN Node ID is calculated at startup only. Input pins assigned to Node ID will be sampled once during power up and used to calculate Node ID. These pins may be assigned other uses after power up if necessary.<br><br>Starting with plus drive firmware version 2.82, new optional method of setting Node IDs of multi-axis drives is supported. This new method is<br>enabled by setting bit 3 of Network Options (0x121). If this method of setting Node IDs is enabled, then parameter 0xC1 is not used for setting Node IDs. Instead, Node IDs of all nodes are set using this parameter. When this optional method of setting Node IDs is used, this parameter is bit mapped as follows:<br><br>This optional method of setting Node IDs allows multi-axis drives to have non-consecutive Node IDs. Note that it is possible to set multiple axes to same Node ID using this method which would result in errors.
<table>
<tr><td>Bit</td><td>Meaning</td></tr>
<tr><td>0-6</td><td>Node ID of axis 1</td></tr>
<tr><td>8-14</td><td>Node ID of axis 2</td></tr>
<tr><td>16-22</td><td>Node ID of axis 3</td></tr>
<tr><td>24-30</td><td>Node ID of axis 4</td></tr>
</table>
</td></tr>
<tr><td>0x104</td><td>0x504</td><td>0x21C4</td><td>RF</td><td>INT16</td><td>Algorithmic Phase Initialization Config. See Value 5 of Commutation Mode (0xB2). This parameter is bit-mapped as follows:
<table>
<tr><td>Bits</td><td>Meaning</td></tr>
<tr><td>0</td><td>If set, don't try to guess phase angle at startup, just force initial phase angle</td></tr>
<tr><td>1</td><td>If set, increment initial phase angle by 90 degrees on each failed attempt.</td></tr>
<tr><td>2</td><td>If set, use Motor Hall Offset (0x4F) as the initial angle for first phase initialization attempt. If clear, first phase angle is zero.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="3">Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>3</td><td colspan="2">Ignore limit switches during phase initialization if switch is configured as trajectory based.<br>Available in Feature set C only.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>4-15</td><td colspan="2">Reserved.</td></tr>
<tr><td>0x105</td><td>0x505</td><td>0x2360</td><td>RF</td><td>U16</td><td colspan="3">Camming Configuration. Bit-mapped as follows.<br>For more information, see Copley Camming User Guide.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Bits</td><td colspan="2">Description</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>0-3</td><td colspan="2">ID Number of Cam Table to use (0-9).</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>4</td><td colspan="2">Reserved.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>5</td><td colspan="2">If set, only allow forward motion through CAM table.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>6</td><td colspan="2">If set, use Camming Internal Generator. Internal generator runs at constant velocity programmed in Camming Master Velocity (0x109).<br>If clear, use digital command input as configured in CME software camming controls or Input Pin States (0xA6).</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>7</td><td colspan="2">If set, run tables stored in RAM.<br>If clear, use tables stored in flash file system.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>8-11</td><td colspan="2">Input number to use as Cam Trigger. Note: a value of 0 selects In1, 1 selects In2, etc.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td rowspan="7">12-14</td><td colspan="2">Cam Trigger type:</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Value</td><td>Type</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>0</td><td>None (Continuous): Active Cam Table is repeated continuously.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>1</td><td>Use Input, Edge: Active Cam Table begins executing on rising edge of input pin selected by bits 8-11.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>2</td><td>Use Input, Level: Active Cam Table will run if input selected by bits 8-11 is high.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>3</td><td>Use Master (Secondary) Encoder Index: Active Cam Table is executed when drive receives an index pulse from Master encoder. Index pulses received during execution are ignored.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>7</td><td>Never trigger. This can be used to stop CAM currently in progress</td></tr>
<tr><td>0x106</td><td>0x506</td><td>0x2361</td><td>RF</td><td>INT16</td><td colspan="3">Camming delay, forward motion.<br>Units: master command counts.<br>This gives delay used when entering cam table in forward direction.</td></tr>
<tr><td>0x107</td><td>0x507</td><td>0x2362</td><td>RF</td><td>INT16</td><td colspan="3">Camming delay, reverse motion.<br>Units: master command counts.<br>This gives delay used when entering a cam table in reverse</td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| | | | | | direction. |
| 0x108 | 0x508 | None | R | INT16 | Writing any value to this parameter will cause any CANopen PDO objects configured with type code 254 to be sent. This parameter is primarily useful for triggering a PDO from within CVM program.<br>Reading this parameter does not return any useful information. |
| 0x109 | 0x509 | 0x2363 | RF | INT 32 | Camming Master Velocity. Units: 0.1 counts/s.<br>Constant velocity of Camming Internal Generator. |
| 0x10A | 0x50A | 0x2403 | R* | INT 32 | Captured Home Position. Units: counts.<br>Provides position that axis was in when an input pin configured as home switch input became active. Configured by setting bits in Position Capture Control Register (0x6C). Status of captured data can be checked in Position Capture Status Register (0x6D). Reading this variable resets bits 4 & 7 of Position Capture Status Register (0x6D). |
| 0x10B | 0x50B | 0x2422 | R* | U32 | Firmware Version Number (extended). Upper 16 bits give same major/minor version number as Firmware Version Number (0x94). Lower 16 bits hold release number (upper byte) and reserved byte (lower). |
| 0x10C | 0x50C | 0x1017 | RF | U16 | CANopen Heartbeat Time. Units: ms.<br>Frequency at which drive will produce heartbeat messages. This parameter may be set to zero to disable heartbeat production. Note that only one of the two node-guarding methods may be used at once. If Heartbeat Time is non-zero, then heartbeat protocol is used regardless of settings of CANopen Node Guarding Time (0x10D) and CANopen Node Guarding Time Life Factor (0x10E). |
| 0x10D | 0x50D | 0x100C | RF | U16 | CANopen Node Guarding Time. Units: ms.<br>This parameter gives time between node-guarding requests that are sent from CANopen master to drive. Drive will respond to each request with<br>node-guarding message indicating internal state of drive.<br>If drive has not received node-guarding request within time period defined by product of Node Guarding Time and CANopen Node Guarding Life Time Factor (0x10E), drive will treat this lack of requests as fault. |
| 0x10E | 0x50E | 0x100D | RF | U8 | CANopen Node Guarding Life Time Factor. This object gives multiple of CANopen Node Guarding Time (0x10D). Drive expects to receive<br>node-guarding request within time period defined by product of CANopen Node Guarding Time (0x10D) and Lifetime Factor. If drive has not received node-guarding request within this time, it treats lack of requests as fault. |
| 0x10F | 0x50F | 0x2325 | R | INT 32 | Registration Offset for Pulse & Direction Mode.<br>When running in pulse & direction mode (Desired State (0x24) = 23), this parameter may be used to inject an offset into master position. Offset will immediately be cleared once it has been applied to master position, so this parameter will |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>normally be read back as zero when running in pulse and direction mode 23.<br>When running in PWM position mode, offset value is added to absolute position calculated using Minimum PWM Pulse Width (0x13C) and Maximum PWM Pulse Width (0x13D) and Digital Input Scaling Factor (0xA9).</td></tr>
<tr><td>0x110</td><td>0x510</td><td>0x2404</td><td>R</td><td>INT 32</td><td>Time Stamp of Last High Speed Position Capture.<br>Units: us.<br>If high speed position capture is enabled, this parameter gives time of last capture.<br>Setting this parameter causes drive to calculate its position at set time if position capture is enabled and time is recent enough for data to be available. Calculated position may be read from Captured Position for High Speed Position Capture (0x111). This feature is mainly used when capturing position on multiple drives across network.</td></tr>
<tr><td>0x111</td><td>0x511</td><td>0x2405</td><td>R*</td><td>INT 32</td><td>Captured Position for High Speed Position Capture. Units: counts.</td></tr>
<tr><td>0x112</td><td>0x512</td><td>0x2242</td><td>R</td><td>INT 32</td><td>Load Encoder Position. Units: counts.<br>This returns position of load encoder if configured. This is also passive load position when used in passive mode.</td></tr>
<tr><td>0x113</td><td>0x513</td><td>0x1015</td><td>RF</td><td>INT16</td><td>CANopen emergency inhibit time. Units: ms.</td></tr>
<tr><td>0x114</td><td>0x514</td><td>0x2381:5</td><td>RF</td><td>U16</td><td>Velocity Loop Drain (integral bleed).<br>Range: 0 to 32767, Default: 0.<br>Modifies effect of velocity loop integral gain.<br>Higher Vi Drain value, faster integral sum is lowered.</td></tr>
<tr><td>0x115</td><td>0x515</td><td>0x2010</td><td>R</td><td>5 Words</td><td>Trajectory Buffer Access. This object can be used to load data into drive's internal trajectory buffer or send commands used to control buffer. Trajectory buffer holds trajectory segments used in PVT mode. Data passed to this parameter consists of a 16-bit command code, followed by up to two 32-bit parameters. First word passed to this parameter is bit-mapped. Data contained in this word identifies this access as either buffer command or trajectory segment to be loaded into buffer. If most significant bit of first word is set, then write is treated as command code. In this case no additional data is passed and first word is formatted as follows:
<table>
<tr><td>Bits</td><td>Description</td></tr>
<tr><td>0-7</td><td>Command data.</td></tr>
<tr><td>8-9</td><td>Command code.</td></tr>
<tr><td>10-14</td><td>Reserved.</td></tr>
<tr><td>15</td><td>Always set for buffer commands.</td></tr>
<tr><td colspan="2">Following command values are supported:</td></tr>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Clear buffer and abort any move in progress.</td></tr>
<tr><td>1</td><td>Pop N most recently sent segments off buffer. PVT profiles will continue to run as long as buffer doesn't underflow. Number of segments</td></tr>
</table></td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| | | | | | | to pop (N) is passed in command data area. If there are less than N segments on buffer, this acts same as buffer clear, except that profile is not stopped except by underflow. | |
| | | | | | To write data to trajectory buffer, most significant bit of first word must be clear. In this case, first word is formatted as follows: | | |
| | | | | | Bits | Description | |
| | | | | | 0-7 | Segment time in ms. | |
| | | | | | 8-11 | Reserved. | |
| | | | | | 12 | Set for relative positions.<br>clear for absolute positions. | |
| | | | | | 13-14 | Reserved. | |
| | | | | | 15 | Always zero for data writes. | |
| | | | | | When writing new PVT segment to trajectory buffer, first word is always followed by a 32-bit position value. Position is specified in units of encoder counts and can be interpreted as either absolute or relative based on bit 12 of command word.<br>Optionally, position can be followed by a 32-bit velocity value. Velocity is specified in units of 0.1 encoder counts/second. If velocity value is<br>supplied, then drive will use cubic polynomial interpolation between points when running trajectory (PVT mode). If velocity is not supplied, then linear interpolation will be used (PT mode).<br>It is acceptable to mix PVT and PT segments within same move.<br>Reading this parameter always returns three words of status information about trajectory buffer.<br>First returned word is formatted as follows: | | |
| | | | | | Bits | Description | |
| | | | | | 0-7 | Number of free locations in trajectory buffer. | |
| | | | | | 8-15 | Reserved. | |
| | | | | | Second two words are reserved for future use. | | |
| 0x116 | 0x516 | 0x605A | RF | INT16 | CANopen Quick Stop Option code. | | |
| 0x117 | 0x517 | 0x605B | RF | INT16 | CANopen Shutdown Option code. | | |
| 0x118 | 0x518 | 0x605C | RF | INT16 | CANopen Disable Option code. | | |
| 0x119 | 0x519 | 0x605D | RF | INT16 | CANopen Halt Option code. | | |
| 0x11A | 0X51A | 0x2080 | F* | U32 | Drive Scaling Configuration. Defines units used for current and voltage readings from drive: | | |
| | | | | | Bits | Description | |
| | | | | | 0-1 | Identify units for current readings: | |
| | | | | | | 0 | 0.01 A |
| | | | | | | 1 | 0.001 A |
| | | | | | | 2 | 0.0001 A |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| | | | | | | 3 | 0.00001 A |
| | | | | | 2-7 | | Reserved |
| | | | | | 8-9 | Identify units for voltage readings: | |
| | | | | | | 0 | 0.1 V |
| | | | | | | 1 | 0.01 V |
| | | | | | | 2 | 0.001 V |
| | | | | | | 3 | 0.0001 V |
| | | | | | 10-31 | Reserved | |
| 0x11B | 0x51B | 0x6082 | R | INT32 | Trajectory Ending Velocity. For use with trap profile mode, gives velocity at end of moves. Primarily used when linking multiple moves together. | | |
| 0x11C | 0x51C | 0x2256 | R | U32 | Trajectory Sequence Buffer Status. Trajectory sequence buffer is used in CANopen profile position mode and stores trajectory segments added using the 'set of setpoints' method described in the CANopen specification. This parameter allows buffer status to be queried. It's bitmapped as follows: | | |
| | | | | | Bits | Description | |
| | | | | | 0-7 | Number of free locations in buffer | |
| | | | | | 8-15 | Number of full locations in buffer | |
| | | | | | 16-31 | Reserved | |
| 0x11D | 0x51D | 0x222B | RF | U32 | Encoder Error Filter Configuration. Encoder error filter can be used to detect and ignore bad position data from an encoder or temporary encoder errors. Bad encoder readings are detected by comparing an expected position (based on extrapolation of previous readings) and actual reading from encoder. | | |
| | | | | | Bits | Description | |
| | | | | | 0-3 | Maximum number of consecutive bad samples to ignore. If zero then filter is disabled. | |
| | | | | | 4-15 | Reserved | |
| | | | | | 16-27 | Maximum error between extrapolated reading and actual reading to consider reading bad | |
| | | | | | 28-31 | Reserved | |
| 0x11E | 0x51E | 0x222C | R | U32 | Encoder Error Filter Status. This can be cleared by writing zero to it. | | |
| | | | | | Bit | Description | |
| | | | | | 0-3 | Count of consecutive bad readings | |
| | | | | | 4-7 | Reserved | |
| | | | | | 8 | Set if encoder fault was generated by filter | |
| | | | | | 9-15 | Reserved | |
| | | | | | 16-31 | Total number of times extrapolated position has been used due to detected error | |
| 0x120 | 0x520 | 0x2384:25 | R* | INT16 | Returns number of axis implemented by this drive. | | |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| 0x121 | 0x521 | 0x21B3 | RF | INT16 | Network Options. Configures drive's network. Details of its meaning depend on type of network implemented in drive. | |
| | | | | | CANopen | |
| | | | | | Bits | Meaning |
| | | | | | 0 | Must be clear to select CANopen networking. |
| | | | | | 1-2 | Reserved. |
| | | | | | 3 | If set, use an alternative method of assigning Node IDs to each axis.<br>See Input Pin Mapping for Node ID Selection (0x103) for details |
| | | | | | 4-7 | Reserved |
| | | | | | 8 | If set, PDO mapping will be saved to flash when object 0x1010 is used to save drive state. |
| | | | | | 9 | If set, PDO communications settings will be stored to flash when object 0x1010 is used to save drive state |
| | | | | | 10-15 | Reserved |
| | | | | | DeviceNet | |
| | | | | | Bits | Meaning |
| | | | | | 0 | Must be set to select DeviceNet networking. |
| | | | | | 1-15 | Reserved. |
| | | | | | MACRO | |
| | | | | | Bits | Meaning |
| | | | | | 0 | If set, position data sent over MACRO network is shifted up 5 bits for compatibility with Delta-Tau controllers. |
| | | | | | 1 | If set, drive will be disabled on startup until it is enabled through MACRO interface.<br>If clear, drive can be used without MACRO interface connected until it starts receiving MACRO messages. |
| | | | | | 2 | If set, return primary encoder index state (high/low) in the home status bit of MACRO status word.<br>If clear, state of any general-purpose input configured as home input will be used. |
| | | | | | 3 | If set, drive will attempt to synchronize its current loop update period to MACRO ring period. Ring period must be an integer multiple of drive's PWM Period (0x85). |
| | | | | | 4-7 | Defines what type of additional data is transmitted in the first auxiliary data register of every MACRO response message:<br>0 – Send digital input value |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | | 1 – Send secondary analog reference value<br>2 – Send unfiltered secondary analog reference value<br>3 – Send motor encoder reading<br>4 – Send load encoder reading |
| | | | | | 8-11 | Defines what type of additional data is transmitted in second auxiliary data register of every MACRO response message:<br>0 – send analog input value<br>1 – send primary encoder reading<br>2 – send secondary encoder reading<br>3 – Pulse & direction hardware count.<br>4 – Unfiltered analog reference value |
| | | | | | 12 | If set, push synchronization point back ½ current loop period. |
| | | | | | 13-15 | Reserved. |
| | | | | | EtherCAT | |
| | | | | | Bits | Meaning |
| | | | | | 0 | If set, disable some extra checks of SYNC0 configuration which were added for improved network conformance. |
| | | | | | 1 | If set, drive will follow EtherCAT state machine even when running in a non EtherCAT mode of operation. |
| | | | | | 2 | If set, object 0x1002 is bit-wise OR of all axes Event Status Register (0xA0) for multi-axis drives.<br>If clear, 0x1002 is for axis 1 only. |
| | | | | | 3 | If set, value of Network Node Id Configuration (0xC1) will be used as network alias on powerup.<br>If clear, alias will be set from address switches |
| | | | | | 4-7 | Reserved. |
| | | | | | 8 | If set, PDO mapping will be saved to flash when parameters are saved<br>using object 0x1010 |
| | | | | | 9 | If set, use UDP command mode rather than standard EtherCAT operation. |
| | | | | | 10-15 | Reserved |
| 0x122 | 0x522 | 0x2384:26 | F* | INT16 | Internal Regen Current. Units : mA.<br>Internal drive constant for factory use. | |
| 0x123 | 0x523 | 0x2220 | RF | INT32 | Motor Encoder Wrap Position. Units: counts<br>Actual motor position will wrap back to zero when this value is reached.<br>Setting this value to zero disables this feature. | |
| 0x124 | 0x524 | 0x2221 | RF | INT32 | Load Encoder Wrap Position. Units: counts<br>Actual load position will wrap back to zero when this value is reached.<br>Setting this value to zero disables this feature. | |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| 0x125 | 0x525 | None | RF | INT16 | Configures MACRO drive's encoder capture circuit. This parameter is only used on MACRO drives.<br>Bit-mapped as follows: | | |
| | | | | | Bits | Meaning | |
| | | | | | 0-3 | Type of capture to use. | |
| | | | | | | Value | Description |
| | | | | | | 0 | Capture on edge of encoder index. |
| | | | | | | 1 | Capture using a general-purpose input pin. |
| | | | | | | 2-15 | Reserved. |
| | | | | | 4-7 | Input pin number to use if using capture type 1. | |
| | | | | | 8 | Active level; high if clear, low if set. | |
| | | | | | 9 | If set, capture is re-enabled immediately when the capture position is read (using I-variable 921).<br>If clear, capture is only re-enabled on an explicit clear instruction. | |
| | | | | | 10 | If set, passive load encoder, if configured, will be captured. Passive load encoder currently only supports capture type 1 (general purpose input). | |
| | | | | | 11-15 | Reserved. | |
| 0x126 | 0x526 | 0x2384:27 | R* | INT16 | FPGA Version Number. | | |
| 0x127 | 0x527 | 0x2370 | RF | U32 | Gain Scheduling Configuration: | | |
| | | | | | Bits | Meaning | |
| | | | | | 0-2 | Key parameter for gain scheduling. | |
| | | | | | | Value | Description |
| | | | | | | 0 | None. Setting key parameter to zero disables gain scheduling. |
| | | | | | | 1 | Use value written to Gain Schedule Key Parameter (0x128) as the key. |
| | | | | | | 2 | Use Instantaneous Commanded Velocity (0x3B). |
| | | | | | | 3 | Use Load Encoder Velocity (0x5F). |
| | | | | | | 4 | Use Commanded Position (0x2D). |
| | | | | | | 5 | Use Actual Position (0x17). |
| | | | | | | 6-7 | Reserved. |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="3">Description</th></tr>
<tr><td rowspan="16"></td><td rowspan="16"></td><td rowspan="16"></td><td rowspan="16"></td><td rowspan="16"></td><td rowspan="8"></td><td colspan="2">In addition to setting this parameter, a gain table must be loaded into the CVM file system. The table must be given the name '_GAINS' in CVM file system. When gain scheduling is active, drive will linearly interpret between rows of the table based on current value of key parameter. This table should contain at least two rows of gains. Each row must contain the following information:</td></tr>
<tr><td>1</td><td>Key value. This is 32-bit value which must increase for each entry in table. Most significant word is stored first.</td></tr>
<tr><td>2</td><td>Position loop proportional gain.</td></tr>
<tr><td>3</td><td>Velocity loop proportional gain.</td></tr>
<tr><td>4</td><td>Velocity loop integral gain</td></tr>
<tr><td>5</td><td>Current offset value</td></tr>
<tr><td>6</td><td>Position loop integral gain</td></tr>
<tr><td>7</td><td>Position loop derivative gain</td></tr>
<tr><td>3-7</td><td colspan="2">Reserved.</td></tr>
<tr><td>8</td><td colspan="2">If set, use absolute value of key parameter for gain lookup.</td></tr>
<tr><td>9</td><td colspan="2">If set, disable gain scheduling until position encoder is referenced.</td></tr>
<tr><td>10-15</td><td colspan="2">Reserved.</td></tr>
<tr><td>16</td><td colspan="2">Table includes position loop PP if set.</td></tr>
<tr><td>17</td><td colspan="2">Table includes velocity loop VP if set</td></tr>
<tr><td>18</td><td colspan="2">Table includes velocity loop VI if set</td></tr>
<tr><td>19</td><td colspan="2">Table includes current loop offset if set</td></tr>
<tr><td>20</td><td colspan="2">Table includes position loop PI if set</td></tr>
<tr><td>21</td><td colspan="2">Table includes position loop PD if set</td></tr>
<tr><td>22-31</td><td colspan="2">Reserved.</td></tr>
<tr><td>0x128</td><td>0x528</td><td>0x2371</td><td>R</td><td>INT32</td><td colspan="3">Gain Scheduling Key Parameter Value.<br>When gain scheduling is enabled, current value of key parameter is stored here. When this parameter is selected as key parameter for gain scheduling, then it may be written to manually move through entries in gain scheduling table.</td></tr>
<tr><td>0x129</td><td>0x529</td><td>0x2384:29</td><td>R</td><td>U32</td><td colspan="3">Drive Hardware Options. Reserved for Copley Controls use.</td></tr>
<tr><td>0x12A</td><td>0x52A</td><td>0x2222</td><td>F</td><td>U32</td><td colspan="3">Motor Encoder Options.<br>Used to specify various configuration options for motor encoder. Mapping of option bits to function depends on encoder type. Any bit not defined for an encoder should be considered reserved. Reserved bits should be set to zero to ensure compatibility with future firmware updates. Bit-mapped as follows:</td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | Quadrature Encoder | |
| | | | | | Bits | Description |
| | | | | | 0 | If set, ignore differential signal errors (if detected in hardware). |
| | | | | | 1 | If set, select single ended encoder inputs (if available in hardware). |
| | | | | | 2 | If set, ignore differential signal errors on index input only (if supported by hardware). |
| | | | | | 3 | If set, don't use index input at all. Useful when index input is being used by a different encoder interface. |
| | | | | | EnDat Encoder (Type 11) | |
| | | | | | Bits | Description |
| | | | | | 0-5 | Number of bits of single turn data available from encoder. |
| | | | | | 8-12 | Number of bits of multi-turn data available from encoder. |
| | | | | | 16 | If set, analog inputs are supplied by encoder. |
| | | | | | 17 | If set, use multi-mode port. |
| | | | | | 18 | If set, read position using EnDat 2.2 style commands rather than default 2.1 style. |
| | | | | | 19 | If set, read encoder at current loop update rate. Otherwise, encoder is read at servo loop period. |
| | | | | | 20-23 | Number of least significant bits of encoder reading to discard. |
| | | | | | SSI Encoder (Type 12) | |
| | | | | | Bits | Description |
| | | | | | 0-5 | Number of bits of position data available. |
| | | | | | 8-11 | Number of extra bits sent with position data. |
| | | | | | 12 | *If set, ignore first bit of data sent by encoder. |
| | | | | | 13 | If set, encoder outputs position data using Gray code. |
| | | | | | 14 | *If set, pull clock low briefly after data (custom for Codechamp encoder). |
| | | | | | 15 | If set, data is sent least signification bit first. |
| | | | | | 16-21 | Encoder Bit Rate.<br>If set, use 100 kHz units.<br>If zero, use default 1 MHz units. |
| | | | | | 22 | *If set, use setting of Motor Encoder Counts/Rev (0x62) to determine how many data bits to use. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 23 | If set, extra status bits are before position data.<br>If clear, extra status bits are after position data.<br>Default is clear. |
| | | | | | 24 | If set, first bit sent is 'data valid' bit. |
| | | | | | 25 | If set, use multi-mode port for SSI interface. |
| | | | | | 26 | If set, extra bits after position data are treated as fault bits and generate an encoder fault if any are set. |
| | | | | | * NOTE – these three bits are depreciated and will be removed in future firmware versions. | |
| | | | | | BiSS (Type 13) | |
| | | | | | Bits | Description |
| | | | | | 0-5 | Number of bits of single turn data |
| | | | | | 8-12 | Number of bits of multiturn data. |
| | | | | | 15 | If set, assume encoder position data wraps after number of encoder counts programmed in Motor Encoder Counts/Rev (0x62). |
| | | | | | 16 | Set for modeC encoder format |
| | | | | | 17 | Set to sample at servo loop rate (default at current loop rate) |
| | | | | | 19 | Set to treat the encoder error bit as a warning (no fault). |
| | | | | | 20 | If set, encoder error and warning bits are active low |
| | | | | | 21 | Set if encoder status bits are sent before position data, clear if status bits are sent after position data |
| | | | | | 22 | If set, encoder error bit is transmitted before warning bit.<br>If clear, warning bit sent first. |
| | | | | | 23 | If set, error bits are sent after alignment bits.<br>If clear, encoder error bits are sent between alignment bits and position data. |
| | | | | | 24-27 | Number of alignment bits (reserved bits sent before position info). |
| | | | | | 28 | If set, use multi-mode encoder.<br>If clear, use primary encoder. |
| | | | | | 30 | If set, use 2.5 MHz baud rate.<br>If clear, use 4 MHz baud rate. |
| | | | | | BiSS encoders are not always consistent with order in which data is sent. We treat data as consisting of three fields, position data <P>, 2 status bits <S> and optional alignment bits <A> which we ignore. Formatting bits identify order of these three fields.<br>Chart below shows order of fields based on format code. Note that data is always sent most significant bit first, so leftmost field is first transmitted.<br>**Format Order of fields**<br>0 <P> <S> <A><br>1 <S> <P> <A> | |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 2 <P> <A> <S><br>3 <S> <A> <P> | |
| | | | | | Tamagawa, Panasonic, Harmonic Drives, etc. (Type 14) | |
| | | | | | Bits | Description |
| | | | | | 0-5 | Number of bits of single turn data. |
| | | | | | 8-12 | Number of bits of multi-turn data. |
| | | | | | 16-19 | Number of LSB to discard from reading. |
| | | | | | 20-22 | Number of consecutive CRC errors to ignore before generating an error. |
| | | | | | 24-27 | Encoder sub-type (0=Tamagawa, 1=Panasonic absolute, 2=HD systems, 3=Panasonic Incremental, 4=Sanyo Denki, 5=Tamagawa Single Turn). |
| | | | | | 28 | Bit rate (set for 4 Mbit, clear for 2.5 Mbit). |
| | | | | | 30 | If set, treat encoder battery errors as warnings. |
| | | | | | 31 | Read the encoder's internal temperature sensor. Currently only for Sanyo Denki encoders. Temperature value read from encoder can be read as encoder register 0. |
| | | | | | Incremental Type E (Type 15) | |
| | | | | | Bit | Description |
| | | | | | 0 | If set, incremental encoder.<br>If clear, absolute encoder. |
| | | | | | 8 | If set, disable interpolation of position |
| | | | | | Gurley Virtual Absolute (Type 17) | |
| | | | | | 0 | If set, invert sine/cosine signals |
| | | | | | 1 | If set, invert virtual absolute signal |
| | | | | | 2 | If set, use custom interface board (customer specific). |
| | | | | | 3 | If set, use encoder digital index input for VABS.<br>If clear, use encoder analog index (if available). |
| | | | | | 8 | If set, switch from algorithmic phase initialization to encoder-based phasing as soon as absolute position is found |
| | | | | | 9 | If set, treat any VABS warnings as encoder fault. If clear, these warnings set status bits but aren't treated as encoder errors. |
| | | | | | Custom Absolute Encoder (Type 18) | |
| | | | | | Bit | Description |
| | | | | | 28 | If set, use multimode input.<br>If clear, use primary encoder input. |
| | | | | | S2 Custom Encoder (Type 19) | |
| | | | | | Bit | Description |
| | | | | | 0-4 | Number of bits of single turn position data / rev. |
| | | | | | 8 | Set for incremental encoders, clear for absolute |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 9 | Use multimode port if set. If clear use primary encoder interface |
| | | | | | 10 | If set, treat encoder battery errors as warnings. |
| | | | | | Wire saving incremental (type 21): | |
| | | | | | Bit | Description |
| | | | | | 0-15 | These bits are the same as a normal incremental encoder (type 0). |
| | | | | | 16 | If set, reverse direction of simulated hall signals after powerup |
| | | | | | 17 | If set, sample halls 10ms after they stabilize on power-up. If clear, sample them after 100ms. |
| | | | | | 18 | If set, force the simulated hall signals to transition coincident with the index signal. |
| | | | | | Sankyo Absolute Encoder (Type 22): | |
| | | | | | Bit | Description |
| | | | | | 0 | Ignore battery errors if set |
| | | | | | 1-31 | Reserved |
| | | | | | Custom Absolute Encoder M (Type 23) | |
| | | | | | Bit | Description |
| | | | | | 0 | Use the multimode port if set |
| | | | | | 1 | Ignore battery errors |
| 0x12B | 0x52B | 0x2223 | F | U32 | Load Encoder Options. Same details as Motor Encoder Options (0x12A) but affects load or position encoder. | |
| 0x12C | 0x52C | 0x2384:28 | R* | U32 | Nios Processor Firmware Version Number.<br>Only used on three-axis drive for now. | |
| 0x12D | 0x52D | 0x2109 | RF | 9 or 14 | Analog Input Filter Coefficients. A biquad filter which acts on the analog reference input. 9- or 14-word parameters, see *Analog Input Filters* in *CME 2 User Guide.* 14-word parameter (Plus and AFS products only), see Filter Coefficients. | |
| 0x12E | 0x52E | 0x2224 | R* | U32 | Motor Encoder Status. This parameter gives additional status information for encoder. Bits set in status word are latched and cleared when status value is read. Format of this status word is dependent on encoder type. Many error bits are taken directly from encoder data stream. For full description of what these error bits mean, please consult encoder manufacturer. | |
| | | | | | Quadrature | |
| | | | | | Bits | Description |
| | | | | | 0 | Only used for custom incremental encoders. Set on startup if encoder did not transmit hall information successfully. |
| | | | | | 1 | Set on bad differential signal levels on any of encoder inputs. |
| | | | | | EnDAT (Type 11) | |
| | | | | | Bits | Description |
| | | | | | 0 | CRC error on data received from encoder. |
| | | | | | 1 | Failed to detect encoder connected to drive. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 2 | Error bit on encoder stream is active. |
| | | | | | 3 | Encoder failed to respond to request for position. |
| | | | | | SSI (Type 12) | |
| | | | | | Bits | Description |
| | | | | | 0-6 | Fault flags returned from encoder. |
| | | | | | 15 | Encoder data invalid bit set. |
| | | | | | BiSS (Type 13) | |
| | | | | | Bits | Description |
| | | | | | 0 | CRC error on data received from encoder. |
| | | | | | 1 | Encoder failed to transmit data to drive. |
| | | | | | 2 | Error bit on encoder stream is active. |
| | | | | | 3 | Warning bit on encoder stream is active. |
| | | | | | 4 | Encoder transmission delay is too long. |
| | | | | | Tamagawa & Panasonic (Type 14) | |
| | | | | | Bits | Description |
| | | | | | 0 | Over speed error reported by encoder. |
| | | | | | 1 | Absolute position error reported by encoder. |
| | | | | | 2 | Counting error reported by encoder. |
| | | | | | 3 | Counter overflow reported by encoder. |
| | | | | | 5 | Multi-turn error reported by encoder. |
| | | | | | 6 | Battery error reported by encoder. |
| | | | | | 7 | Battery warning reported by encoder. |
| | | | | | 8 | Error bit 0 reported by encoder. |
| | | | | | 9 | Error bit 1 reported by encoder. |
| | | | | | 10 | Comm error 0. |
| | | | | | 11 | Comm error 1. |
| | | | | | 15 | CRC error on data received from encoder. |
| | | | | | Sanyo Denki & Harmonic Drives (Type 14) | |
| | | | | | Bits | Description |
| | | | | | 0 | Battery warning reported by encoder. |
| | | | | | 1 | Battery error reported by encoder. |
| | | | | | 3 | Over speed reported by encoder. |
| | | | | | 4 | Memory error reported by encoder. |
| | | | | | 5 | STERR reported by encoder. |
| | | | | | 6 | PSERR reported by encoder. |
| | | | | | 7 | Busy error reported by encoder. |
| | | | | | 8 | Memory busy reported by encoder. |
| | | | | | 9 | Over temperature reported by encoder. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 15 | CRC error on data received from encoder. |
| | | | | | Harmonic Drives (Encoder Type 15) | |
| | | | | | Bit | Description |
| | | | | | 0 | System error reported by encoder |
| | | | | | 1 | Overflow error reported by encoder |
| | | | | | 2 | Mode error reported by encoder |
| | | | | | 3 | Battery error reported by encoder |
| | | | | | 4 | CRC error on data received from encoder |
| | | | | | 5 | No data received from encoder on read |
| | | | | | Gurley Virtual Absolute (Encoder type 17) | |
| | | | | | Bit | Description |
| | | | | | 0 | Amplitude of Sine/ Cosine signals is out of range |
| | | | | | 1 | Encoder power current limited |
| | | | | | 2 | Encoder moving too fast during initialization |
| | | | | | 3 | Missing trigger signal (only occurs when using custom interface hardware). |
| | | | | | 4 | Virtual absolute signal changed state at incorrect time |
| | | | | | 5 | Invalid virtual absolute data received. |
| | | | | | 6 | Encoder has not finished initializing position |
| | | | | | Custom Absolute Encoder K (Type 18) | |
| | | | | | Bit | Description |
| | | | | | 0 | Busy bit from encoder set |
| | | | | | 1 | ABSALM bit from encoder set |
| | | | | | 2 | INPALM bit from encoder set |
| | | | | | 8 | CRC error on data received from encoder |
| | | | | | S2 Custom Encoder (Type 19) | |
| | | | | | Bit | Description |
| | | | | | 0 | Battery error alarm bit from encoder |
| | | | | | 1 | Encoder error alarm bit from encoder |
| | | | | | 2 | Battery warning alarm bit from encoder |
| | | | | | 3 | Absolute error alarm bit from encoder |
| | | | | | 4 | Over speed error alarm bit from encoder |
| | | | | | 5 | Over heat error alarm bit from encoder |
| | | | | | 8 | CRC error on data received from encoder |
| | | | | | 9 | Encoder not responding to queries from drive. |
| | | | | | Sankyo Absolute Encoder (Type 22) | |
| | | | | | Bit | Description |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 0 | Set if encoder is not responding to commands |
| | | | | | 1 | Set if error bit is returned by encoder |
| | | | | | 2 | Set if encoder returns incorrectly formatted data |
| | | | | | 3-7 | Reserved |
| | | | | | 8 | Encoder reports “MR sensor amplitude error” |
| | | | | | 9 | Encoder reports “Multi rotation data error” |
| | | | | | 10 | Encoder reports “battery error” |
| | | | | | 11 | Encoder returned reserved error bit |
| | | | | | 12 | Encoder reports “MR sensor error” |
| | | | | | 13 | Encoder reports “Over speed error” |
| | | | | | 14 | Encoder reports “Temperature error” |
| | | | | | 15 | Encoder returned reserved error bit |
| | | | | | Custom Absolute Encoder M (Type 23) | |
| | | | | | Bit | Description |
| | | | | | 0 | Encoder reported 'CPU alarm' |
| | | | | | 1 | Encoder reported reserved alarm bitmapped |
| | | | | | 2 | Encoder reported 'Data alarm' |
| | | | | | 3 | Encoder reported 'Thermal alarm' |
| | | | | | 4 | Encoder reported 'Thermal warning |
| | | | | | 5 | Encoder reported 'Multi revolution alarm' |
| | | | | | 6 | Encoder reported 'Absolute position lost warning' |
| | | | | | 7 | Encoder reported 'Battery disconnect' |
| | | | | | 8-12 | Reserved |
| | | | | | 13 | Incorrect data type returned from encoder |
| | | | | | 14 | Encoder not responding to reads |
| | | | | | 15 | Encoder CRC data error |
| 0x12F | 0x52F | 0x2225 | R* | U32 | Load Encoder Status. Same details as Motor Encoder Status (0x12E), but for load encoder. | |
| 0x130 | 0x530 | 0x2114 | RF | INT16 | RMS Current Calculation Period. Units: ms.<br>This sets period over which RMS current is calculated. If this value is set to zero, then RMS current will be updated each time it is read for period since the last read. In this case, RMS current must be read at least once every 65536 current loop periods (about every 4 seconds) for returned RMS values to be accurate. | |
| 0x131 | 0x531 | 0x2115 | R* | INT16 | RMS Current over period set in RMS Current Calculation Period (0x130). Units: 0.01 A. | |
| 0x132 | 0x532 | 0x2116 | R* | INT16 | Running Sum of User Current Limit. Units: 0.01%. Values will be 0 to 10000. | |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th>Description</th></tr>
<tr><td>0x133</td><td>0x533</td><td>0x2117</td><td>R*</td><td>INT16</td><td>Running Sum of Drive Current Limit. Units: 0.01% Values will be 0 to 10000.</td></tr>
<tr><td>0x134</td><td>0x534</td><td>0x21E0</td><td>RF</td><td>U32</td><td>Analog Output D/A converter configuration.<br>This parameter sets mode for D/A converter on drives so equipped.<br><br>Bits | Description<br>0-3 | Defines mode of D/A converter<br>16 | If set, current outputs will be scaled based on motor peak current setting rather than drive's internal scaling.<br>Currently supported modes are:<br>Mode | Description<br>0 | Manual configuration. Set using Analog Output D/A (0x135)<br>1 | Actual Current of configured axis. If bit 16 is clear, then output voltage is scaled so that full 5V output on D/A will correspond to Current Corresponding to Max A/D Reading (0x84). If bit 16 is set, then voltage is scaled based on motor peak current setting<br>2 | Actual Velocity of configured axis, ratio of actual velocity to Velocity Loop Velocity Limit (0x3A).<br>3 | U winding current, scaled same as mode 1.<br>4 | V winding current, scaled same as mode 1.<br>5 | W winding estimated current, scaled same as mode 1.</td></tr>
<tr><td>0x135</td><td>0x535</td><td>0x21E1</td><td>R</td><td>INT16</td><td>Analog Output D/A Converter Output Value.<br>Units: mV.<br>For drives that support auxiliary D/A converter, this sets output value when D/A is in manual mode. In other modes, current value being output on D/A can be read here.</td></tr>
<tr><td>0x136</td><td>0x536</td><td>0x2208</td><td>R*</td><td>INT16</td><td>Second Analog Input. Units: mV.<br>Also known as Secondary analog reference value.</td></tr>
<tr><td>0x137</td><td>0x537</td><td>0x2314</td><td>RF</td><td>INT16</td><td>Offset for secondary analog reference input.<br>Units: mV.</td></tr>
<tr><td>0x138</td><td>0x538</td><td>0x2315</td><td>RF</td><td>INT16</td><td>Calibration offset, second analog input. Units: mV.<br>Factory-calibrated to give zero reading for zero input voltage.</td></tr>
<tr><td>0x139</td><td>0x539</td><td>0x219D</td><td>R</td><td>INT32</td><td>Drive Safety Circuit Status.<br>This parameter allows status of safety circuit built into some drives to be queried. For drives without safety circuit, this parameter is reserved.<br><br>Bits | Description<br>0 | Set when safety input 0 is preventing drive from enabling.<br>1 | Set when safety input 1 is preventing drive from enabling.</td></tr>
</table>

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>8 | This read/write bit can be used to force ‘drive is unsafe’ output of safety circuit to go active for testing purposes. Write 1 to force this output active. Write zero for normal operation.</td></tr>
<tr><td>0x13A</td><td>0x53A</td><td>0x2209</td><td>R*</td><td>INT16</td><td>Present Voltage at Analog Motor Temperature Sensor.<br>Units: mV.<br>If thermistor characteristics have been programmed in Steinhart Contants (0x19A), then temperature is returned in degrees C.<br>Note that this parameter is only valid for drives that include an analog temperature sensor input.</td></tr>
<tr><td>0x13B</td><td>0x53B</td><td>0x220A</td><td>RF</td><td>INT16</td><td>Limit for Analog Motor Temperature Sensor.<br>Units: mV.<br>If this parameter is set to zero, then analog motor temperature sensor is disabled.<br>If this parameter is set to positive value, then motor temperature error will occur any time voltage on motor temperature input exceeds this value.<br>If this parameter is set to negative value, then motor temperature error will occur any time voltage on the motor temperature input is lower than absolute value of this limit.<br>If thermistor characteristics have been programmed in Steinhart Constants (0x19A), then this gives maximum motor temperature in degrees C.</td></tr>
<tr><td>0x13C</td><td>0x53C</td><td>0x2323</td><td>RF</td><td>INT16</td><td>Minimum PWM Pulse Width. Units: ms.<br>Used when running in PWM position mode. In this mode PWM input pulse width is captured by drive and used to calculate an absolute position using following formula:<br>pos = ((PW-MIN) / (MAX-MIN)) * SCALE + OFFSET<br>where this parameter is minimum pulse width (MIN), parameter 0x13D is maximum pulse width (MAX), parameter 0xA9 is scaling factor (SCALE) and parameter 0x10F is offset (OFFSET).</td></tr>
<tr><td>0x13D</td><td>0x53D</td><td>0x2324</td><td>RF</td><td>INT16</td><td>Maximum PWM Pulse Width. Units: us.<br>Used only when running in PWM position mode.</td></tr>
<tr><td>0x13E</td><td>0x53E</td><td>0x222A</td><td>RF</td><td>U32</td><td>Encoder Adjustment Table Configuration.<br>See applications note for additional details.<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Set to enable encoder adjustment table</td></tr><tr><td>1</td><td>If set, use resolver angle adjustment tables.<br>If clear, use normal encoder adjustment tables</td></tr><tr><td>2</td><td>Set for cogging compensation mode (see applications note for details)</td></tr></table></td></tr>
<tr><td>0x13F</td><td>0x53F</td><td>0x232B</td><td>RF</td><td>INT16</td><td>PWM Input Deadband. Range of 0 to 32767 equals deadband of 0 to 100%.</td></tr>
</table>

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th>Description</th></tr>
<tr><td>0x141</td><td>0x541</td><td>0x2243</td><td>R</td><td>INT16</td><td>Resolver angle scaled so 180 deg is 32767.<br>Only valid when using resolver as motor encoder feedback.<br>Reserved for other encoder types.</td></tr>
<tr><td>0x150</td><td>0x550</td><td>0x210A</td><td>RF</td><td>14</td><td>Second chained biquad filter on output of velocity loop. 14-word parameter, see Filter Coefficients.</td></tr>
<tr><td>0x151</td><td>0x551</td><td>0x210B</td><td>RF</td><td>14</td><td>Third chained biquad filter on output of velocity loop. 14-word parameter, see Filter Coefficients.</td></tr>
<tr><td>0x152</td><td>0x552</td><td>0x210C</td><td>RF</td><td>14</td><td>First chained biquad filter on input of current loop. 14-word parameter, see Filter Coefficients.</td></tr>
<tr><td>0x153</td><td>0x553</td><td>0x210D</td><td>RF</td><td>14</td><td>Second chained biquad filter on input of current loop. 14-word parameter, see Filter Coefficients.</td></tr>
<tr><td>0x154</td><td>0x554</td><td>0x2301</td><td>RF</td><td>INT32</td><td>Servo Loop Configuration. This parameter allows various parts of drive servo loops to be enabled/disabled. Bit-mapped as follows:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>If set, this disables Velocity loop gains.<br>Velocity Feed Forward (0x157) is still active as are velocity loop output filters.</td></tr>
<tr><td>1</td><td>If set, this enables Position Loop I (0x155) and Position Loop D (0x156) gains.<br>If clear, these are treated as zeros.</td></tr>
<tr><td>2</td><td>If set, velocity error windows will be calculated using filtered version of the motor velocity.<br>If clear, unfiltered velocity will be used.</td></tr>
<tr><td>8</td><td>If set, initialize velocity loop integral sum when switching modes to prevent glitches on output current</td></tr>
<tr><td>Other</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>0x155</td><td>0x555</td><td>0x2382:5</td><td>RF</td><td>INT16</td><td>Integral Gain, Position Loop (PI).</td></tr>
<tr><td>0x156</td><td>0x556</td><td>0x2382:6</td><td>RF</td><td>INT16</td><td>Derivative Gain, Position Loop (PD).</td></tr>
<tr><td>0x157</td><td>0x557</td><td>0x2381:6</td><td>RF</td><td>INT16</td><td>Velocity Loop Command Feed Forward (Vcff).<br>Input command (after limiting) to velocity loop is scaled by this value and added into output of velocity loop.</td></tr>
<tr><td>0x158</td><td>0x558</td><td>0x2382:7</td><td>RF</td><td>INt16</td><td>Integral Drain, Position Loop (Pi Drain).</td></tr>
<tr><td>0x159</td><td>0x559</td><td>0x6007</td><td>RF</td><td>INT16</td><td>Abort Option Code, CANopen/EtherCAT drives.</td></tr>
<tr><td>0x15A</td><td>0x55A</td><td>0x2198</td><td>RF</td><td>U32</td><td>I/O Options. This parameter is used to configure optional features of general purpose I/O.
<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>0-3</td><td colspan="2">For Plus Module drives, these bits determine whether several I/O pins are used as serial interface for expanded I/O features, and if so how they are configured.</td></tr>
<tr><td></td><td>0</td><td>Normal I/O</td></tr>
<tr><td></td><td>1</td><td>Plus Module drive development board LEDs and address switches</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>4-7</td><td>Reserved</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>8</td><td>For Plus module drives, setting this bit allows the STO LED to be illuminated even if the drive is disabled by firmware if the STO inputs are connected</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>9-15</td><td>Reserved</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>16</td><td>On AC powered Plus drives, this bit disables AC line drop detection if set</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>17-31</td><td>Reserved</td></tr>
<tr><td>0x15B</td><td>0x55B</td><td>0x2199</td><td>F</td><td>INT16</td><td colspan="2">Motor Brake Enable Delay Time. Units: ms.<br>This parameter gives delay between enabling drive PWM outputs and releasing brake. Positive values mean PWM is enabled first and brake is released N ms later. Negative values cause brake to<br>be released before PWM outputs are enabled.</td></tr>
<tr><td>0x15C</td><td>0x55C</td><td>0x219A</td><td>R*</td><td>U32</td><td colspan="2">Input Pin States, 32-bit.<br>32-bit version of Input Pin States (0xA6).<br>Each bit gives high/low state of one general purpose input pin. Lower 16 bits of this parameter are equivalent to value returned by Input Pin States (0xA6). This parameter is primarily used for drives with more than 16 general purpose input pins.</td></tr>
<tr><td>0x15D</td><td>0x55D</td><td>0x219B</td><td>R*</td><td>U32</td><td colspan="2">Raw Input State, 32-bit.<br>32-bit version of Raw Input State (0xAA).<br>Gives current high/low state of all general-purpose inputs before any debounce is applied.</td></tr>
<tr><td>0x15E</td><td>0x55E</td><td>0x219C</td><td>RF</td><td>U32</td><td colspan="2">Input Pin Configuration, 32-bit.<br>32-bit version of Input Pin Configuration (0xA5). Used to configure pull up/down resistors on drives with more than 16 such resistors.</td></tr>
<tr><td>0x15F</td><td>0x55F</td><td>0x237B</td><td>RF</td><td>U32</td><td colspan="2">Motor Cogging Compensation. This was added to Plus drives starting with version 3.18 firmware.<br>Scales current command to motor based on sine of phase angle plus programmable offset.<br>Bit-mapped as follows:</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Bit</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>0-7</td><td>Gives an angular offset in units of 360/256 deg</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>8-15</td><td>Reserved</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>16-31</td><td>Gives scaling value. Scale = 1.0 + X/16384 where X is unsigned value programmed in these bits. Resulting scale ranges from 0 <= scale < 5.0</td></tr>
<tr><td>0x160</td><td>0x560</td><td>0x2192:17</td><td>RF</td><td>U16</td><td colspan="2">Input Pin Configuration, General Purpose Input 17. See Input 0 Configuration (0x78).<br>For notes on Input numbering, see Input/Output Numbering.</td></tr>
</table>

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th>Description</th></tr>
<tr><td>0x161</td><td>0x561</td><td>0x2192:18</td><td>RF</td><td>U16</td><td>Input Pin Configuration, General Purpose Input 18. See Input 0 Configuration (0x78)</td></tr>
<tr><td>0x162</td><td>0x562</td><td>0x2192:19</td><td>RF</td><td>U16</td><td>Input Pin Configuration, General Purpose Input 19. See Input 0 Configuration (0x78)</td></tr>
<tr><td>0x163</td><td>0x563</td><td>0x2192:20</td><td>RF</td><td>U16</td><td>Input Pin Configuration, General Purpose Input 20. See Input 0 Configuration (0x78)</td></tr>
<tr><td>0x164</td><td>0x564</td><td>0x2192:21</td><td>RF</td><td>U16</td><td>Input Pin Configuration, General Purpose Input 21. See Input 0 Configuration (0x78)</td></tr>
<tr><td>0x165</td><td>0x565</td><td>0x2192:22</td><td>RF</td><td>U16</td><td>Input Pin Configuration, General Purpose Input 22. See Input 0 Configuration (0x78)</td></tr>
<tr><td>0x166</td><td>0x566</td><td>0x2192:23</td><td>RF</td><td>U16</td><td>Input Pin Configuration, General Purpose Input 23. See Input 0 Configuration (0x78)</td></tr>
<tr><td>0x167</td><td>0x567</td><td>0x2192:24</td><td>RF</td><td>U16</td><td>Input Pin Configuration, General Purpose Input 24. See Input 0 Configuration (0x78)</td></tr>
<tr><td>0x170</td><td>0x570</td><td>0x2195:17</td><td>RF</td><td>U16</td><td>Debounce Time, General Purpose Input 17.<br>Units: ms.</td></tr>
<tr><td>0x171</td><td>0x571</td><td>0x2195:18</td><td>RF</td><td>U16</td><td>Debounce Time, General Purpose Input 18.<br>Units: ms.</td></tr>
<tr><td>0x172</td><td>0x572</td><td>0x2195:19</td><td>RF</td><td>U16</td><td>Debounce Time, General Purpose Input 19.<br>Units: ms.</td></tr>
<tr><td>0x173</td><td>0x573</td><td>0x2195:20</td><td>RF</td><td>U16</td><td>Debounce Time, General Purpose Input 20.<br>Units: ms.</td></tr>
<tr><td>0x174</td><td>0x574</td><td>0x2195:21</td><td>RF</td><td>U16</td><td>Debounce Time, General Purpose Input 21.<br>Units: ms.</td></tr>
<tr><td>0x175</td><td>0x575</td><td>0x2195:22</td><td>RF</td><td>U16</td><td>Debounce Time, General Purpose Input 22.<br>Units: ms.</td></tr>
<tr><td>0x176</td><td>0x576</td><td>0x2195:23</td><td>RF</td><td>U16</td><td>Debounce Time, General Purpose Input 23.<br>Units: ms.</td></tr>
<tr><td>0x177</td><td>0x577</td><td>0x2195:24</td><td>RF</td><td>U16</td><td>Debounce Time, General Purpose Input 24.<br>Units: ms.</td></tr>
<tr><td>0x180</td><td>0x580</td><td>0x2326</td><td>RF</td><td>U32</td><td>UV configuration. Used to configure drive when running in UV mode, Desired State (0x24), Mode 5.<br>Bit-mapped as follows (undocumented bits reserved for future use):
<table>
<tr><td>Bits</td><td colspan="2">Meaning</td></tr>
<tr><td>0-1</td><td colspan="2">Define source of UV command inputs:</td></tr>
<tr><td></td><td>Value</td><td>Description</td></tr>
<tr><td></td><td>0</td><td>PWM inputs.</td></tr>
<tr><td></td><td>1</td><td>Analog reference inputs<br>(for drives with two analog reference inputs).</td></tr>
<tr><td></td><td>2</td><td>Analog encoder inputs.</td></tr>
<tr><td></td><td>3</td><td>Directly set over serial/network interface.</td></tr>
<tr><td>2-7</td><td colspan="2">Reserved.</td></tr>
<tr><td>8-9</td><td colspan="2">Define format of UV inputs:</td></tr>
<tr><td></td><td>Value</td><td>Description</td></tr>
<tr><td></td><td>0</td><td>120 degree current commands.</td></tr>
<tr><td></td><td>1</td><td>90 degree current commands.</td></tr>
</table></td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | | |
|---|---|---|---|---|---|---|---|
| | | | | | | 2 | Angle/Magnitude form.<br>U input gives magnitude.<br>V gives angle. |
| | | | | | 10-15 | Reserved. | |
| | | | | | 16 | If set, value of Motor Hall Offset (0x4F) is added to UV angle | |
| | | | | | 17 | If set, drive will use field oriented control. Normally FOC is disabled in UV mode due to ambiguity of phase angle with zero inputs. This is best used when running in angle/magnitude format. | |
| 0x181 | 0x581 | 0x2327 | R | INT16 | U Input when running in UV mode.<br>This parameter can be used to read calculated U value or to set U value when UV inputs are being directly set over serial/network interface. | | |
| 0x182 | 0x582 | 0x2328 | R | INT16 | V Input when running in UV mode.<br>Same as 0x181 but for V Input. | | |
| 0x183 | 0x583 | 0x2329 | R | INT16 | Raw Counter Value From Pulse & Direction Input. This can be read when running in any mode, not<br>just pulse & direction modes.<br>This parameter can be written also, but should not be written when drive is being controlled by pulse<br>& direction inputs. Writing in that mode will<br>cause drive to treat change in counter as real pulse inputs resulting in possible unexpected motion. | | |
| 0x184 | 0x584 | 0x2254 | RF | 8 to 40 | Input Shaping Filter.<br>This filter is used to modify trajectory before it is input into position loop. This can be used to compensate for low frequency resonances in loads.<br>Parameter is an array of 32-bit values. First four values are used to store information about input shaping filter (filter type, frequency, etc.) and are mostly unused by firmware. The only exception is that most significant bit of first word should not be set to ensure compatibility with future firmware versions.<br>The remaining 32-bit values are pairs of IEEE floating point values. Each pair defines a time (first value) and an impulse amplitude (second value). Up to eight pairs may be passed for up to 8 impulses in input shaping filter. Time values are specified in seconds and must be >= 0.0. Impulse values are unit-less and must have an absolute magnitude of < 16.0. | | |
| 0x185 | 0x585 | 0x2160 | R | U32 | Output Compare Configuration. For detailed description of output compare function, see Trigger Outputs at Position application note describing it. | | |
| 0x186 | 0x586 | 0x2161 | R | U32 | Output Compare Status. | | |
| 0x187 | 0x587 | 0x2162 | R | INT32 | Output Compare Value 1. | | |
| 0x188 | 0x588 | 0x2163 | R | INT32 | Output Compare Value 2. | | |
| 0x189 | 0x589 | 0x2164 | R | INT32 | Output Compare Increment. | | |
| 0x18A | 0x58A | 0x2165 | R | INT32 | Output Compare Pulse Width. | | |
| 0x18B | 0x58B | 0x2255 | RF | INT32 | Trajectory options.<br>This parameter is used to modify behavior of some | | |

<table>
<tr><th>ASCII</th><th>MACRO</th><th>CAN/ECAT<br>IDX: SUB</th><th>Mem</th><th>Type</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="2">trajectory modes. Interpretation depends on trajectory mode being used.<br>Following trajectory modes currently make use of this parameter:</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="2">EtherCAT CSP mode:</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Bits</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>0-7</td><td>Number of extra loop cycles to extrapolate trajectories if input<br>data from master is not received.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>8-15</td><td>Reserved.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>16</td><td>If set, jump to quick stop mode if master data is not received within number of cycles set in bits 0-7.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>17</td><td>If set, and Interpolation Time object (0x60C2) is non-zero, then calculated velocity will be filtered, and trajectory acceleration will also be calculated.<br>If clear, velocity is unfiltered, and acceleration is not calculated (zero).</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>18-31</td><td>Reserved.</td></tr>
<tr><td>0x18C</td><td>0x58C</td><td>0x21A1</td><td>RF</td><td>U32</td><td colspan="2">I/O Extension Configuration for Plus Modules.<br>This parameter is used to configure I/O extension feature on Plus Modules which support it.<br>For detailed description of this I/O extension feature, see Extending Plus Module I/O application note.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>Bits</td><td>Description</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>0-7</td><td>Number of bits to transfer less 1 (e.g., set to 19 to transfer 20 bits).</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>8</td><td>Reserved</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>9</td><td>If set, automatically restart transmission.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>10</td><td>If set, leave CS line low after transfer.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>11</td><td>Status bit indicating new receive data is available. Auto-cleared when data is read via parameter 0x18E</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>12</td><td>Clock polarity setting.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>13</td><td>Data phase setting.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>14-15</td><td>Reserved.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>16-23</td><td>Clock period. Units: 100 ns.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>24-27</td><td>Reserved.</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>28</td><td>If set, enable SPI I/O extension feature. If clear, enable LED/Switch interface</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>29-31</td><td>Reserved</td></tr>
<tr><td>0x18D</td><td>0x58D</td><td>0x21A2</td><td>R</td><td>INT32*</td><td colspan="2">I/O Extension Transmit Data.<br>Data to be transferred over SPI port is sent immediately after being written here.<br><br>*Data size is variable dependent on drive configuration. Refer to Extending Plus Module I/O application note.</td></tr>
<tr><td>0x18E</td><td>0x58E</td><td>0x21A3</td><td>R</td><td>INT32*</td><td colspan="2">I/O Extension Receive Data.<br>Data received from SPI port can be read here.<br><br>*Data size is variable dependent on drive configuration. Refer to Extending Plus Module I/O application note.</td></tr>
</table>

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0x18F | 0x58F | 0x220B | RF | INT16 | Encoder Sine Offset. This is set in A/D units and only used with resolvers and servo-tube motors. It gives an offset which is added to encoder sine signal before calculating position. |
| 0x190 | 0x590 | 0x220C | RF | INT16 | Encoder Cosine Offset. Like 0x18F, but for encoder cosine signal. |
| 0x191 | 0x591 | 0x220D | RF | U16 | Encoder Cosine Scaling Factor.<br>Used by resolver & Servotube encoder calculations. This scaling factor is used to adjust cosine signal amplitude so it is same as sine signal amplitude.<br>If set to zero, both Encoder Sine Offset (0x18F) and Encoder Cosine Offset (0x190) will be ignored.<br>If non-zero the cosine is scaled by N/32768 where N is the value of this parameter. |
| 0x192 | 0x592 | 0x2226 | RF | U32 | Motor Encoder Calibration settings. The meaning of this value is dependent on encoder type. |
| 0x193 | 0x593 | 0x2227 | RF | U32 | Load Encoder Calibration settings. Same as 0x192, but applied to load encoder. |
| 0x194 | 0x594 | 0x232A | R* | INT16 | PWM Input Duty Cycle.<br>This can be used to read duty cycle of PWM input. Returned 16-bit value gives duty cycle in range<br>+/-32767. Digital Input Command Configuration (0xA8) is used to configure PWM input. |
| 0x195 | 0x595 | 0x2123 | RF | INT32 | Jerk Value. Units: 100 counts/$s^3$.<br>Value to use during trajectory aborts.<br>If this is zero, abort will be calculated without any jerk limits. |
| 0x196 | 0x596 | 0x220E | R* | INT32 | Returns magnitude squared of analog encoder signals (sin*sin + cos*cos) |
| 0x197 | 0x597 | 0x2378 | RF | INT16 | Cross Coupling XPp Gain.<br>On dual axis drives this gain is applied to difference in position error of two axes. |
| 0x198 | 0x598 | 0x2379 | RF | INT16 | Cross Coupling XPi Gain. |
| 0x199 | 0x599 | 0x237A | RF | INT16 | Cross Coupling XPd Gain. |
| 0x19A | 0x59A | 0x220F | RF | 5 words | Steinhart Constants, Motor Analog Temperature Sensor.<br>This parameter is only used on drives that include motor temperature sensor analog input. For such drives, this parameter can be used to define type of NTC thermistor connected to analog input.<br>If nonzero, motor temperature (in degrees C) will be read from Present Voltage at Analog Motor Temperature (0x13A) rather than analog voltage. Parameter uses same format as an output pin configuration, 16-bit integer followed by two 32-bit integers. Three integer values contain A, B and C Steinhart- Hart coefficients for motor thermistor. Three coefficients are scaled by following constants:<br>A: 1.0e6<br>B : 1.0e7<br>C: 1.0e10<br>For example, A thermistor with coefficients 1.4626e3, 2.4024e4 and 8.0353e8 would be configured with the three integer values: 1463, 2402 and 804. |
| 0x19B | 0x59B | 0x2384:30 | F* | INT16 | Current at which minimum PWM deadtime is used. |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description |
|---|---|---|---|---|---|
| 0x19C | 0x59C | 0x2406 | R* | INT32 | High Speed Capture, Passive Load Encoder. |
| 0x19D | 0x59D | 0x2142 | RF | INT16 | Open Motor Wiring Check Current.<br>If Motor Brake Enable Delay Time (0x15B) is greater than zero, then during that time period on enable this current will be applied to motor wiring to check that motor is connected. If programmed current cannot be applied to motor, then motor disconnected fault will be flagged. |
| 0x19E | 0x59E | 0x6066 | RF | U16 | Tracking Window Warning Time. Units: ms. |
| 0x19F | 0x59F | 0x2264 | RF | INT16 | Phase Advance. Scaled so 32767 is 180 degrees. |
| 0x1A0 | 0x5A0 | 0x2193:9 | RF | 3-5 | Output 9 Configuration. See Output 0 Configuraton (0x70). |
| 0x1A1 | 0x5A1 | 0x2193:10 | RF | 3-5 | Output 10 Configuration. See Output 0 Configuraton (0x70). |
| 0x1A2 | 0x5A2 | 0x2193:11 | RF | 3-5 | Output 11 Configuration. See Output 0 Configuraton (0x70). |
| 0x1A3 | 0x5A3 | 0x2193:12 | RF | 3-5 | Output 12 Configuration. See Output 0 Configuraton (0x70). |
| 0x1A8 | 0x5A8 | 0x2228 | RF | INT16 | Motor Encoder Downshift.<br>This parameter is useful when using very high resolution encoders that would otherwise have limited speed and travel distance due to range of position and velocity parameters. Setting downshift causes position read from encoder to be right-shifted before being used. For example, setting this parameter to value of 2 effectively cuts the encoder resolution by a factor of 4.<br>If set, servo loops use fractional encoder counts, therefore encoder resolution is not completely lost. |
| 0x1A9 | 0x5A9 | 0x2229 | RF | INT16 | Load Encoder Downshift. Same as Motor Encoder Downshift (0x1A8), but for load encoder. |
| 0x1AA | 0x5AA | 0x21E2 | RF | INT16 | Fan Turn On Temperature. Units: Degrees C.<br>For products with software controlled internal fan, this value is temperature when fan will first turn on. |
| 0x1AB | 0x5AB | 0x21E3 | RF | INT16 | Fan Max Speed Temperature. Units: Degrees C.<br>For products with software controlled internal fan, this value is temperature when fan will run at top speed. Must be >= Fan Turn On Temperature (0x1AA) or value will be ignored |
| 0x1AD | 0x5AD | 0x21E4 | RF | INT16 | Encoder Cosine Angular Offset. Units: 0.1 degree<br>This parameter gives angular error of encoder cosine signal. Used to compensate for imperfections in encoder signals. This adjustment is only used if<br>Encoder Cosine Scaling Factor (0x191) is nonzero. |
| 0x1AE | 0x5AE | 0x21A4 | RF | U32 | Inter-drive communication configuration.<br>This parameter is only used on drives that support the IDC bus. |
| | | | | | Bit: Description |
| | | | | | 0: Set for IDC master. Clear for IDC slave devices |
| | | | | | 1: Set to disable serial command forwarding via IDC |
| | | | | | 8-10: Address of partner axis for cross coupling |
| 0X1AF | 0x5AF | 0x21A5 | R | U32 | Inter-drive communication status. |
| | | | | | Bit: Description |
| | | | | | 0: Synchronized to IDC bus if set |

| ASCII | MACRO | CAN/ECAT IDX: SUB | Mem | Type | Description | |
|---|---|---|---|---|---|---|
| | | | | | 1 | Address assignment complete if set |
| | | | | | 2 | IDC running normally if set |
| | | | | | 8 | Set if IDC is reset |
| | | | | | 16-18 | Assigned IDC address |

# 4 FILTER COEFFICIENTS

There are several drive parameters which are used to define filters. These filters are implemented as generic biquadratic filter structures. Filters of this type implement the following formula to transform the input parameter x(n) at time n to an output parameter y(n):

$y(n) = b0 \cdot x(n) + b1 \cdot x(n-1) + b2 \cdot x(n-2) + a1 \cdot y(n-1) + a2 \cdot y(n-2)$

Values a1, a2, b0, b1, b2 are constants known as filter coefficients. They define the type of filter being implemented.

Values passed to these drive filter parameters are used to define filter coefficients. Formatting of these parameters varies depending on drive family being interfaced to.

All first-generation Copley drives use 16-bit integer math to implement their filters internally. Filter coefficients are given as 16-bit signed integer values. To increase resolution of these coefficients, an additional unsigned scaling coefficient (k) is also specified. Actual filter formula used within these drives is as follows:

$y(n) = K \cdot (b0 \cdot x(n) + b1 \cdot x(n-1) + b2 \cdot x(n-2) + a1 \cdot y(n-1) + a2 \cdot y(n-2)) / 32768 / 4096$

To set filter coefficients on drives of this category, 9 words of parameter data are passed. First three words of data are informational parameters which are used by CME software to describe filter. If upper 3 bits of first word are all set, then filter will be disabled. Other than that, first three words of data are not used in any way by firmware. These three words are reserved for CME use.

| Word | Description |
|---|---|
| 1 | Filter info. Set to 0xFFFF to disable filter. Otherwise, reserved for CME use. |
| 2 | Filter info. Reserved for CME use. |
| 3 | Filter info. Reserved for CME use. |
| 4 | b2 coefficient. |
| 5 | b1 coefficient. |
| 6 | b0 coefficient. |
| 7 | a2 coefficient. |
| 8 | a1 coefficient. |
| 9 | K scaler. |

For Plus family of drives (Accelnet Plus, Stepnet Plus, Xenus Plus), a new format is used to describe Biquad filter coefficients. These drives include ability to design filters in firmware using Cephes filter design library.

Filters on these families of drives are calculated internally using 32-bit IEEE floating point coefficients. Format of parameter information passed when setting filter parameters on these drives consists of an array of up to fourteen 16-bit words. First 4 words describe filter and remaining 10 words give filter coefficients as 32-bit IEEE floating point values. Filter coefficient words are optional and are only necessary if firmware is not calculating coefficients internally.

| Word | Description | |
|---|---|---|
| 1 | Bits | Usage |
| | 0-3 | Filter family. |
| | 4 | If set, filter will not be designed. Always set by firmware after successfully designing filter. This prevents filter from being redesigned when copied from flash at startup. |
| | 5-7 | Reserved. |
| | 8 | Number of poles – 1 (i.e. 0 for single pole, 1 for two pole). |
| | 9-12 | Reserved. |
| | 13-15 | Filter type. |
| | All reserved bits should be set to zero.<br>Filter family should be one of following values: | |
| | 0 | Custom Bi-quad filter. Coefficients must be passed; firmware will not design filter. |
| | 1 | Butterworth filter. |
| | 2 | Chebychev filter. |
| | 3 | Elliptic filter. |
| | 4-15 | Reserved. |
| | Filter type should be one of the following: | |
| | 0 | Custom Bi-quad filter. Coefficients must be passed; firmware will not design filter. |
| | 1 | Low pass. |
| | 2 | High pass. |
| | 3 | Band reject (notch). |
| | 4 | Band pass. |
| | 5-6 | Reserved |
| | 7 | Disabled. The filter will have no effect in system. |
| | If legal values are passed for filter type and family, firmware will attempt to design specified filter and fill in coefficient values itself. Firmware can calculate 1- or 2-pole low pass or high pass filters. For notch and band pass filters firmware can only calculate 2-pole filter. For these filter types, bit 8 must be set. | |
| 2 | This word gives cut off frequency for low pass and high pass filters. Units: Hz.<br>For notch and band pass filters this gives first filter frequency. | |
| 3 | This word gives second filter frequency for notch and band pass filters. Units: Hz. | |
| 4 | Bits | Usage |
| | 0-7 | Rp. Units: 0.1 dB. |
| | 8-15 | Rs. Units: dB. |
| | Rp is pass band ripple. This parameter is only used for Chebychev and Elliptic filters.<br>Rs used only with elliptic filters. Defines stop band as Rs dB down from peak value in pass band. | |
| 5-6 | Coefficient a1. All filter coefficients are passed as 32-bit IEEE floating point numbers.<br>The upper 32-bits should be passed first.<br>If firmware designs filter then coefficients will be filled in by firmware and need not be passed. | |
| 7-8 | Coefficient a2. | |

| Word | Description |
|---|---|
| 9-10 | Coefficient b0. |
| 11-12 | Coefficient b1. |
| 13-14 | Coefficient b2. |

Parameter Dictionary

P/N 16-01091

Revision 02

March 2019


Copley Controls
20 Dan Road
Canton, MA 02021 USA