MACRO Network User Guide

copley controls

P/N 16-01052
Revision 00
February 2013

This page for notes

TABLE OF CONTENTS

About This Manual

1.1.1: Overview and Scope

This manual describes the use of the MACRO network with Copley Controls drives and controllers. MACRO is an acronym for *Motion and Control Ring Optical*. It is a non-proprietary fiber optic network developed by Delta Tau for use in controlling devices such as servo drives.

1.1.2: Related Documentation

MACRO-related documents:

- 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 Amplifier Parameter Dictionary (describes the parameters used to program and operate Copley Controls drives)
- CME 2 User Guide (describes the installation and use of Copley Controls CME 2 software)

Copley ASCII Interface Programmer's Guide and Copley Amplifier Parameter Dictionary can be found here: http://www.copleycontrols.com/Motion/Downloads/protocols.html.

CME 2 User Guide can be found here:
http://www.copleycontrols.com/Motion/Downloads/software.html.

1.1.3: Comments

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

1.1.4: Copyrights

1.1.5: 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.

1.1.6: Product Warnings

Observe all relevant state, regional and local safety regulations when installing and using Copley Controls servo drives. For safety and to assure compliance with documented system data, only Copley Controls should perform repairs to servo 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 the *CME 2 User Guide.*

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.

1.1.7: Revision History

Revision	Date	Comments
00	February 2013	Initial publication.

.

CHAPTER

1: INTRODUCTION

This chapter discusses how Copley Controls supports the use of the MACRO network to provide distributed motion control.

1.1: The MACRO Network

MACRO was developed by Delta Tau for a single cable connection between multi-axis motion controllers, drives, and I/O through a fiber optic ring or twisted copper pair. Copley drives connect to a MACRO ring via SC-type fiber optic connectors. Copley's MACRO drives may be configured using Copley's CME 2 software or by using I-variables with a communications protocol. This manual describes the most useful I-variables. For a comprehensive list see the Copley Amplifier Parameter Dictionary (http://www.copleycontrols.com/Motion/Downloads/protocols.html).

The MACRO network consists of one or more master controllers (typically a Delta-Tau PMAC card), and a number of slave devices. The master controller sends out messages to each of the slaves on the ring and each slave passes the messages on to the next device on the ring until they return to the master.

Each MACRO message passed on the ring is 12 bytes long. It includes a ring command byte, an ID byte (containing the master/slave address), a checksum byte (used to verify data integrity), and 9 data bytes. The 9 data bytes in a MACRO message are grouped into a single 24-bit register and three 16-bit registers (see Cyclic Registers, p. 25).

Each master on the ring is assigned an identifying number in the range 0 to 3. Each slave device on the ring is assigned both a master number and a slave identifier (see Slave ID Assignment, p. 11).

1.1.1: Transfer Rate

MACRO uses a 125 Mbits/sec transfer rate which will close the servo loops across the MACRO ring, allowing the flexibility to choose distributed intelligence or centralized control.

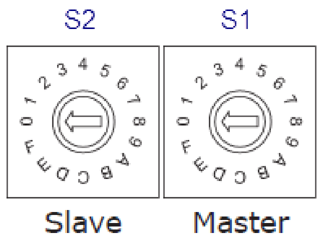
CHAPTER

2: SLAVE ID ASSIGNMENT

This chapter describes how to set slave ID values for Copley Controls MACRO compatible drives.

2.1: Introduction

Copley's MACRO drives have two rotary switches used for MACRO slave identification located on the outside frame. S1 is used to select the master ID value that the slave device is associated with. S2 is used to select the slave ID value. Drives must have both switches set to match the master's message to which it intends to respond.

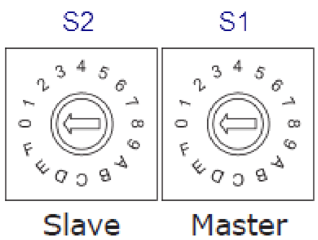


As messages are passed around the ring, each slave device evaluates the message to determine if it is addressed to that slave. If either the master ID or slave ID contained in the message does not match the slave's ID values, the message will be passed to the next device on the ring, with no modification. If both the master ID and slave ID values match the slave's, a response message will be passed on to the next device on the network.

Note: It is unnecessary to use MACRO ASCII to set slave IDs since the switches on the Copley drive will override any settings. In addition, setting slave IDs using the *ring order method* is unnecessary. Delta Tau also refers to a Station ID, or Number (this is in addition to the Node Number).The Station ID (or Number), is used by Copley Drives only for ASCII communication.

2.2: Setting the Node ID Switches

The table below shows the available selections for S1 & S2. Boxes greyed-out are invalid selections and have no function. The switch positions are numbered in hexadecimal. The table shows these positions with the master and slave addresses in decimal.



Switch	S2	S1
Address	SLAVE	MASTER
HEX	DEC	
0	0	0
1	1	1
2		2
3		3
4	4	
5	5	
6		
7		
8	8	
9	9	
A		
B		
C	10	
D	11	
E		
F		

CHAPTER

3: CONFIGURING USING CME2

This chapter describes using CME 2 to configure Copley drives and the MACRO network. It is suggested that Torque (current) mode be used for drives on a MACRO ring.

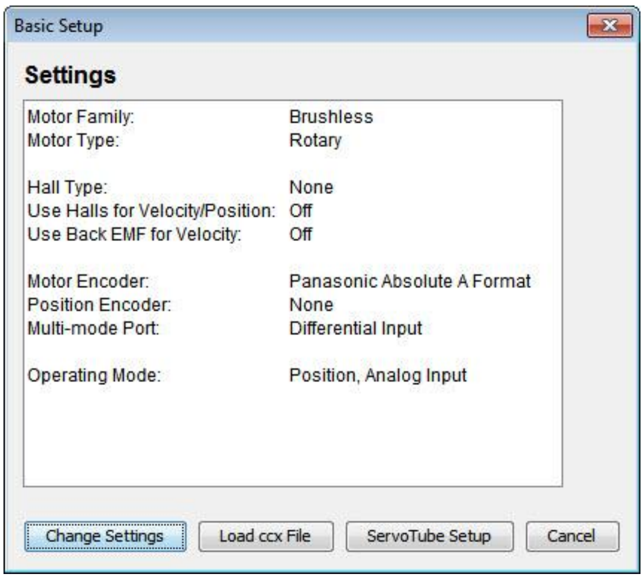
> Note: Before using CME 2’s Control Panel to operate motors, the MACRO network must be disabled.

3.1: Setting a Drive to Current Mode

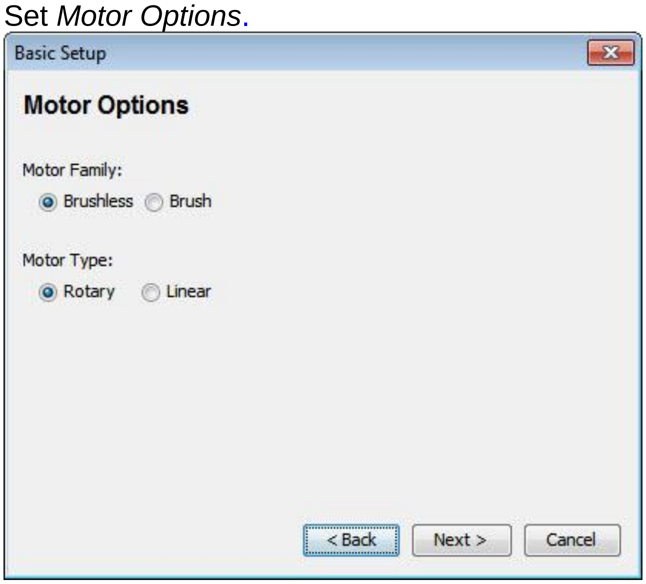
This section describes setting a drive to current mode using the Basic Setup in CME 2. It is essential that the Command Source be **Software Programmed** (step 5). Perform the steps outlined below.

1 In the CME 2 main screen Click  to open the *Basic Setup* screen.

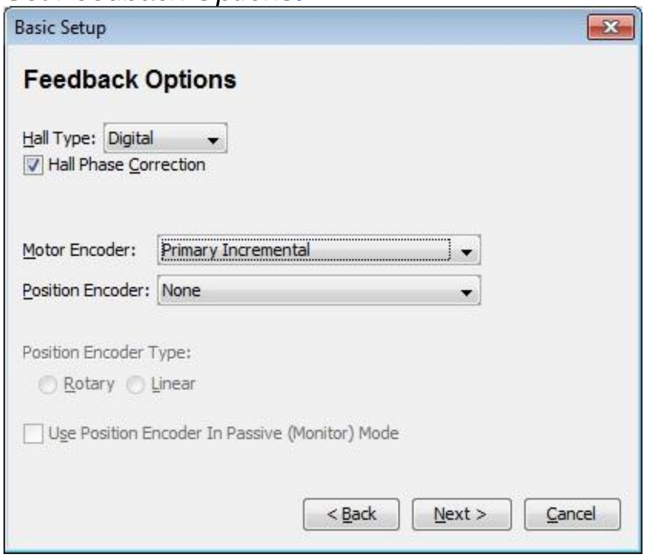
2 On the *Basic Setup* screen, click **Change Settings** to start the Basic Setup wizard. Screen details vary depending on amplifier model and mode selection.



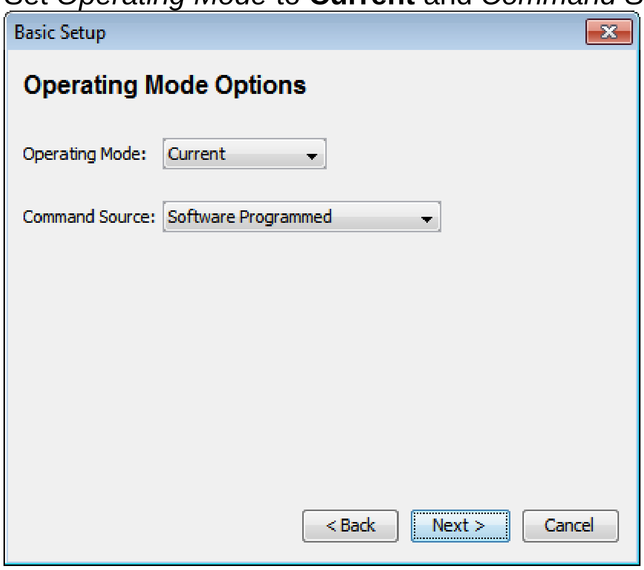
3 Set *Motor Options*.



4 Set *Feedback Options*.



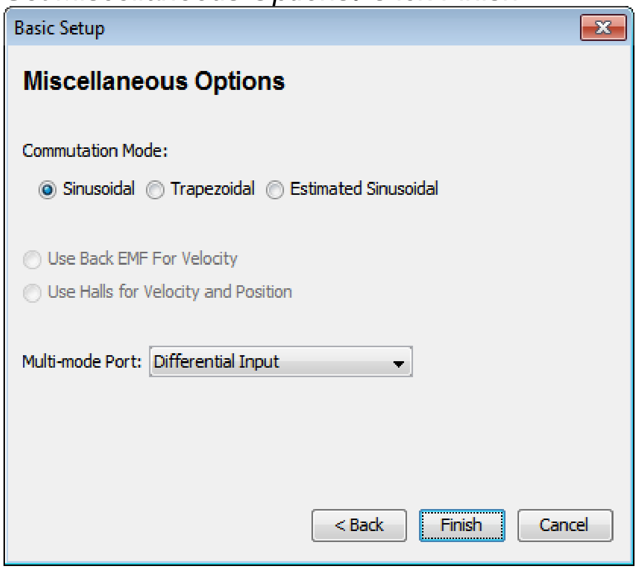
5 Set *Operating Mode* to **Current** and *Command Source* to **Software Programmed.**



Continued…

…Setting a Drive to Current Mode, continued:

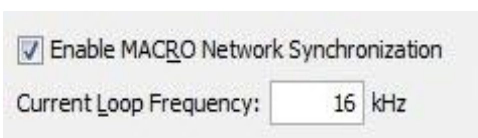
6 Set *Miscellaneous Options*. Click Finish



3.2: Sync Function

The MACRO ring update frequency is the phase clock frequency of the ring master controller. It must function at a multiple of the Copley drive's current loop update frequency. This is critical to establish synchronous operation. Copley drives' current loop frequency may be adjusted in CME 2's MACRO configuration window (see Configuring the MACRO Network, p. 19).

In CME 2's MACRO configuration screen there is a check box to enable current loop frequency adjustment.



Note: Copley's Accelnet MACRO (AMP) drive comes loaded with a current loop frequency of 15 kHz. This should usually be set to 16 kHz using CME 2. Copley's Xenus Plus for MACRO (XML) drive comes with a default current loop frequency of 16 kHz and may usually be left alone.

If there is more than one PMAC controller on the ring, only one of them can be the ring master controller. If there is an excess of nodes on the MACRO ring, the PMAC ring update frequency may have to be lowered by a multiple of the drive's current loop frequency for smooth operation.

Note: Changing the PWM Frequency will affect the current loop tuning. Therefore, current loop tuning will need to be checked. See the CME 2 User Guide for details.

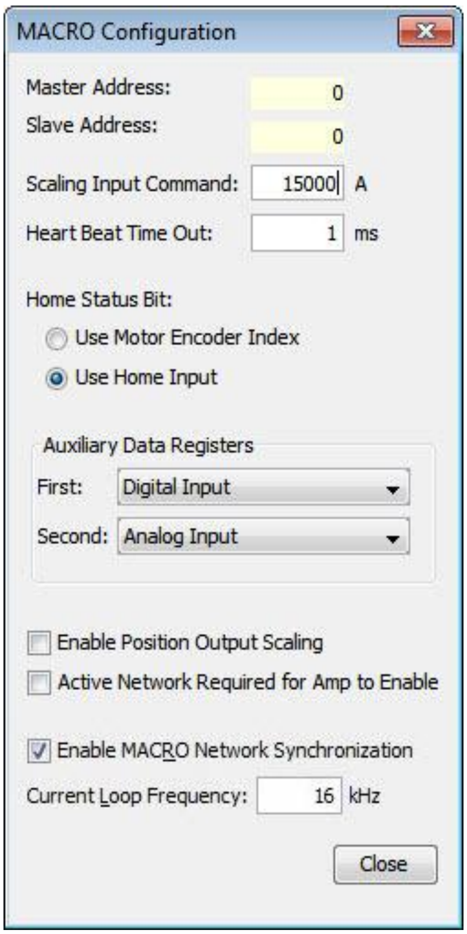
3.3: Configuring the MACRO Network

Accelnet MACRO drives can be configured over the serial port using CME 2 software.

To configure a MACRO interface using CME 2

1 Verify the S1 and S2 switch settings (see Slave ID Assignment, p. 11).

2 Select **Amplifier→Network Configuration** from the CME 2 Main Screen to open the MACRO Configuration screen as shown in the following example.



Continued…

…Configuring the MACRO Network, continued:

3 Verify or adjust the following parameters.

Parameter	Description
Scaling Input Command	Current mode: output current produced by +10 Vdc of input. Range: 0 to 10,000,000 A. Default: Peak Current value. Velocity mode: output velocity produced by +10 Vdc of input. Range: 0 to 100,000 rpm (mm/sec). Default: Maximum Velocity value.
Heart Beat Time Out	The frequency at which the drive will produce heartbeat messages. It is suggested this be set to 1 ms. The default is 0 ms, but this will disable heartbeat production, will not detect a ring break and may cause run away conditions.
Home Status Bit	Use Motor Encoder index: Return the primary encoder index state in the home status bit of the MACRO status word. Use Home Input: The state of any general purpose input configured as a home input will be returned in the home status bit of the MACRO status word.
Auxiliary Data Registers	Defines what type of additional data is transmitted in the Auxiliary data registers of every MACRO response message. First Register: Digital input Value, Secondary Analog Input Value. Second Register: Analog Input, Motor Encoder, Position Encoder.
Enable Position Output Scaling	When selected, position data sent over the MACRO network is shifted up 5 bits to be compatible with Delta-Tau controllers.
Enable MACRO Network Synchronization	Allows the drive's PWM frequency to be adjusted to allow synchronous operation with the MACRO ring. Note that changing the PWM Frequency will affect the current loop tuning. Therefore, current loop tuning will need to be checked.
Current Loop Frequency	
Active Network Required for Drive to Enable	If selected, drive will not enable if network is not active.

4 Click **Close** to close the screen and save the settings.

CHAPTER

4: PMAC AND DRIVE SYNC FREQUENCIES

This chapter describes how to setup the Master controller (PMAC) with a motor, and how to sync drives on the MACRO ring. It is imperative that the PMAC update rate is matched to the MACRO masters or slaves current loop frequency (see Sync Function, p. 18). The tables in this section are suggested setups.

4.1: Examples Table

To configure the PMAC ring update frequency and desired servo drive frequency, set the following PMAC I-variables to the suggested values indicated.

Desired Servo Frequency	MACRO Comm. Freq.(PhaseFreq.)	Ultralite / UMAC Settings	Notes
4 kHz	16 kHz	I6800=3684 I6801=0 I6802=3 I10=2096640	**Ring Bandwidth limited to ~48 active Servo and I/O nodes
4 kHz	8 kHz	I6800=7371 I6801=0 I6802=1 I10=2097066	
2 kHz	8 kHz	I6800=737 I6801=0 I6802=3 I10=4194133	
8 kHz	16 kHz	I6800=3684 I6801=0 I6802=1 I10=1048320	**Ring Bandwidth limited to ~48 active Servo and I/O nodes

Refer to Delta Tau Turbo SRM and Turbo Users Manuals for I70-I82 and MACRO IC 1-3 setups.

4.2: PMAC Communication Setup Example

Below is an example of PMAC I-variables set to establish a 16 kHz MACRO ring frequency and a 4kHz servo loop (using an Ultralite or UMAC PMAC), for the first 4 servo axes on a MACRO ring (Nodes 0,1,4,5).

	Enter	Description
1	I6800=3684	Max phase clock frequency.
2	I6801=0	Frequency of the phase clock signal for the system — controlling its division from the max phase clock frequency.
3	I10=2096640	Servo update time.
4	I6840=$4030	MACRO IC 0 master configuration.
5	I6841=$0FC033	MACRO IC 0 Node activation control.
6	I70=$0033	Node auxiliary function enable.
7	I71=$0033	Node protocol type control.
8	I78=128	Master/Slave Auxiliary Communications Timeout.

4.3: PMAC Motor Setup Example

Below is an example of commands for configuring a Delta Tau PMAC controller to work with a motor connected to a Copley drive.

	Enter	Description
1	I100=1	Activate motor.
2	I101=0	Disable commutation, Copley drive does commutation.
3	I102=$078420	Command Output Address, dependent on mode.
4	I103=@$I8001	Position loop feedback address, assigned the value to the address of the conversion table I-variable.
5	I104=@$8001	Velocity loop feedback address, assigned the value to the address of the conversion table I-variable.
6	I111=16*25000	Set the following error limit.
7	I119=1	Set the acceleration limit.
8	I122=500	Set the jog speed.
9	I125=$003440	Flag address.
10	I124=$840001	Set flag modes. High true amp state, capture flags through macro and PMAC 2 style bits are set.
11	I130=7172	I-variables 130-135 are the servo loop gains, specific to this example only.
12	I131=1934	Servo loop gains.
13	I132=1934	Servo loop gains.
14	I133=197466	Servo loop gains.
15	I134=0	Servo loop gains.
16	I135=15714	Servo loop gains.
17	I8000=$2F8420	Encoder conversion table value.
18	I8001=$018000	Encoder conversion table value.

CHAPTER

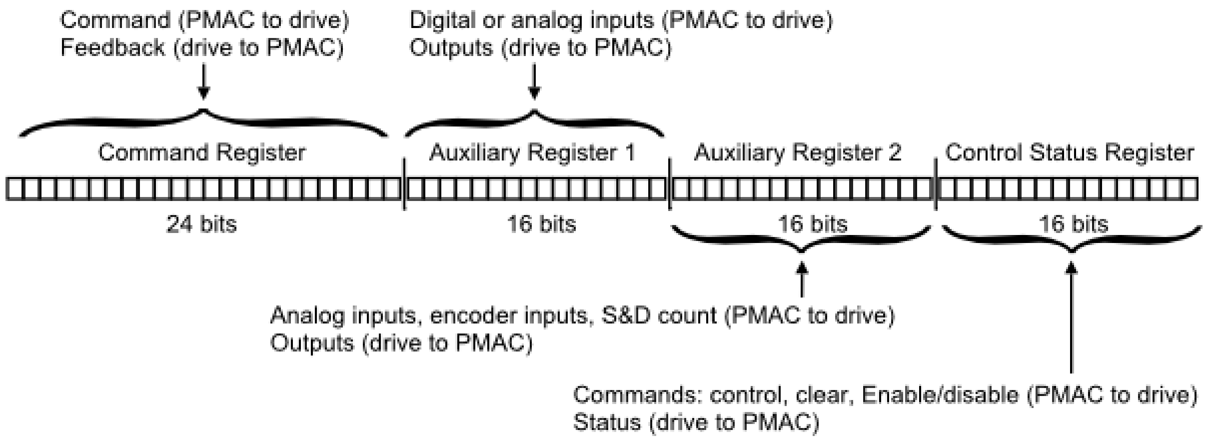
5: CYCLIC REGISTERS

This chapter describes the cyclic registers on the MACRO ring.

5.1: The MACRO Message

Each MACRO message passed on the MACRO ring is 12 bytes long. The message includes a ring command byte, an ID byte (containing the master/slave address), a checksum byte (used to verify data integrity), and 9 data bytes.

The 9 data bytes in a MACRO message are grouped into one 24-bit register, and three 16-bit registers, for a total of 72 bits that cycle through each node on the MACRO ring.



As messages are passed around the ring, each slave device evaluates the message to determine if the message is addressed to that slave. If either the master or slave identifier contained in the message does not match the slave's ID values, the message will be passed to the next node on the ring with no modification. If both the master and slave ID values match the slave, the slave will pass a response message to the next device on the network.

For PMAC register details see Delta Tau's Macro Drive User Manual.

5.2: Command Register

The 24-bit register in the MACRO message is used to pass a command value to the drive, and to pass a motor position back from the drive to the master. The command value sent can either be a current command or a velocity command depending on the mode of operation of the drive (current command is suggested). The response message from the drive will always use this 24-bit register to send the primary encoder position back to the master.

When the drive is configured to run in current mode, a 24-bit current command is passed in the command register. The scaling (transconductance) of this current command is programmable through the drive's I-variable 1193. This I-variable gives the actual current (in 0.01 Amp units) that corresponds to the maximum positive 24-bit input value.

When the drive is configured to run in velocity mode, the value passed in the command register is treated as a velocity command. This command is also scaled by the system transconductance which is programmed as a maximum velocity in units of 0.1 encoder counts/second.

In both current and velocity mode the 24-bit command register location is used to return the current encoder position back to the MACRO master controller. The units that the position is reported in can be configured as either encoder counts, or 1/32 encoder counts. The latter selection is only provided to improve compatibility with the Delta Tau master software. However, the drive itself does not resolve fractional encoder counts. So, if 1/32 encoder count units are selected, the lower 5 bits of the position feedback will always be zero.

5.3: Auxiliary Register 1

The first 16-bit register in the MACRO message is used to send the state of the general purpose digital input and output pins between the master and the drive.

Copley drives have several general purpose output pins which can be programmed to function in a number of different ways. These pins may be configured to be controlled by the drive itself (for example, a brake or fault output), or may be configured to be manually controlled by the MACRO master.

Any pins which are configured to be manually controlled by the MACRO master may be set using the value in the digital I/O register. Each time a MACRO update is received by the drive, each manually controlled digital output pin will be updated based on the corresponding bit in this register.

> Note: The MACRO status word returned during every cycle has bits for positive and negative limit. If an input pin is configured in CME as a positive limit, then when it goes active, it will cause the appropriate status bit to be set. This is true whether or not the input pin values are returned in aux register 1.

Output pin 1 will be set based on the value of bit 0, output pin 2 will be set based on the value of bit 1, etc. On the response messages sent from the drive back to the master, the value of this register can take one of several programmable values. Bits 4-7 of the MACRO network configuration parameter (drive parameter 0x121, or I-variable 0x521) are used to select the data stored here.

The following values are currently supported:

Value	Data
0	Send digital input value.
1	Send secondary analog reference value.

5.4: Auxiliary Register 2

The second 16-bit register value can be programmed to transfer various types of data between the master and slave device.

For messages sent from the master to the slave, this register passes a 16-bit value that will be written to the XML model drive’s analog output. For AMP model drives this value is reserved. In order for the analog output value to have any effect, the general purpose analog output of the XML must be configured for manual control. Manual mode can be configured by setting drive parameter 0x134 (or I-variable 0x534) to zero.

On response messages sent from the slave device to the master this register's contents are configurable. Bit 8-11 of the MACRO network configuration parameter is used to select the data to be sent back to the master in this register location:

Value	Data
0	Drive's analog reference input in units of millivolts
1	Lowest 16-bits of the drive's motor encoder input.
2	Lowest 16-bits of the drive’s load encoder input (passive or active).
3	Raw 16-bit value of pulse & direction counter.

5.5: Control Status Register

The third 16-bit register is a non-programmable Control Status Register. It is a command register with the following bits mapped.

5.5.1: PMAC to Drive

The message from the PMAC to the drive is a control word, bit mapped a follows:

Value	Data
3	Trigger latch enable. Enables position capture when set. Clears captured position on the 0->edge.
4	Clears encoder position to 0 when set.
6	Enable (1) or Disable (0) the drive. Any latched faults are cleared on the 0->1 transition. Other bits are reserved.

5.5.2: Drive to PMAC

The response is a status word with the following bits mapped:

Value	Data
0	Encoder error.
1	Position compare output set.
2	Set if safety inputs are disabling drive.
3	Position captured flag.
4	Node reset.
5	Ring break detected elsewhere.
6	Drive enabled.
7	Shutdown fault.
8	Home flag.
9	Positive limit.
10	Negative limit.
11	Not used.
12	Not used.
13	Phase initialization.
14	Current limit.
15	Voltage limit.

CHAPTER
6: I-VARIABLES

This chapter discusses how I-variables are used in MACRO compatible Copley drives. It includes tables with commonly used I-variables, and position capture and position compare examples.

6.1: Communicating with the Drive

While configuring drives using Copley's CME 2 software is the easiest way to start, accessing the drives internal parameters is possible using MACRO communication. These internal parameters may be accessed using MACRO node 14 (ASCII) or node 15 communications.

In node 15 communications, the PMAC implements a command with *ms(node)* preceding the I-variable*.* (node) is the slave's ID value. For example, to read the most recently captured index position of slave 0, the I-variable 921 command would be:

```
ms0,i921
```

In node 14 (ASCII) communications, the PMAC implements a command by simply entering an I-variable preceded by *i*. Before using node 14, an ASCII communications window must be opened. This is done by typing *macsta (station number*) in the communications protocol window (I-variable 11 holds the station number). As an example: to read the most recently captured index position using a node 14 communications window, the I-variable 921 command would be:

```
i921
```

The most useful parameters are listed in this chapter. For a full list see the Copley Amplifier Parameter Dictionary (http://www.copleycontrols.com/Motion/Downloads/protocols.html).

> Note: When using ASCII to set parameters of word size 3 or larger, HEX values must be used.

6.2: Accessing Drive Parameters

Each drive parameter, as documented in the Copley ASCII Interface Programmer's Guide (http://www.copleycontrols.com/Motion/Downloads/protocols.html), is identified with an ASCII parameter number in the range of 0 to 511. To access the drive's parameters over MACRO, 1024 is added to the parameter id. For example: I-variable 1025 is the same as parameter id 1.

Due to limitations in node 15 communications, it isn't possible to directly read/write a value greater than 48-bits using this method. Some drives have internal parameters that are larger than this (such as bi-quad filter coefficients). If such a parameter is read using the associated I-variable, the first 48 bits will be returned. Writing such a parameter using this I-variable will only set the first 48 bits, other bits will be set to zero. An extended access command (I-variable 1018) is used to work around this limitation (see Extended Command Access, p. 52).

Note: I-variable tables in this manual have already been interpolated.

6.3: Supported Delta Tau I-variables

Below is a list of useful I-variables used to access Copley drives on the MACRO ring.

<table>
<tr><th>Ixx</th><th>Title</th><th>Description</th></tr>
<tr><td>0</td><td>Firmware version</td><td>Returns the drive's firmware version number. Read only.</td></tr>
<tr><td>1</td><td>Firmware date</td><td>Returns the firmware build date in the format of MM/DD/YYYY>.</td></tr>
<tr><td>2</td><td>Station ID and User Config word</td><td>Read/Write. Saved to flash.</td></tr>
<tr><td>3</td><td>Switch values</td><td>Reports the values of the two hardware switches.</td></tr>
<tr><td>4</td><td>Drive Status</td><td>Returns a 32-bit status register value. This status is bit-mapped as follows:
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>Set if a current limit has been reached.</td></tr>
<tr><td>2</td><td>Set if a short circuit has been detected.</td></tr>
<tr><td>4</td><td>Set on motor over temperature.</td></tr>
<tr><td>6</td><td>Set on drive over temperature.</td></tr>
<tr><td>8</td><td>Set on encoder feedback error detection.</td></tr>
<tr><td>20</td><td>Set on bus under voltage condition.</td></tr>
<tr><td>21</td><td>Set on bus over voltage condition</td></tr>
<tr><td>24</td><td>Set if amp fault occurs.</td></tr>
<tr><td>25</td><td>Set if MACRO network fault occurs. (Set on break detected).</td></tr>
</table>
All other bits are currently reserved and should be ignored.</td></tr>
<tr><td>5</td><td>Ring error counter</td><td>The number of MACRO errors since power-up. Read only.</td></tr>
<tr><td>8</td><td>Macro ring check period</td><td>The MACRO ring check period. Represents one watchdog period for a Copley drive. Read/Write. Units: ms.</td></tr>
<tr><td>11</td><td>Station number</td><td>Identical to the STN ASCII command. Read/Write. Saved to flash.</td></tr>
<tr><td>12</td><td>Device ID</td><td>This returns a Copley Controls’ hardware type device ID value. Values currently supported are:
<table>
<tr><th>Value</th><th>Type</th></tr>
<tr><td>$0390</td><td>AMP panel drive rev 0</td></tr>
<tr><td>$0391</td><td>AMP panel drive rev 1</td></tr>
<tr><td>$0392</td><td>AMP panel drive rev 2</td></tr>
<tr><td>$1010</td><td>XML A/C powered drive</td></tr>
</table></td></tr>
<tr><td>910</td><td>Sets encoder direction</td><td>Normal direction is used if bit 2 is cleared. Reversed direction is used if bit 2 is set. All other bits are ignored.</td></tr>
<tr><td>920</td><td>Absolute position at power up</td><td>The absolute position for the power-on read of position. A read only parameter.</td></tr>
<tr><td>921</td><td>Get captured position</td><td>Returns the most recently captured index position (32-bits). Writes are ignored.</td></tr>
<tr><td>923</td><td>Compare auto increment value</td><td>Output compare increment value.</td></tr>
<tr><td>925</td><td>Compare A position value</td><td>Output compare position A.</td></tr>
<tr><td>926</td><td>Compare B position value</td><td>Output compare position B.</td></tr>
<tr><td>928</td><td>Compare state write enable</td><td>Forces compare output to the disabled state on write (reads ignored).</td></tr>
</table>

<table>
<tr><td>974</td><td>Display code</td><td>Gets the standard station display code.
The following code values are supported:
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Disabled</td></tr>
<tr><td>1</td><td>Enabled</td></tr>
<tr><td>10</td><td>Drive fault</td></tr>
<tr><td>14</td><td>Encoder feedback fault</td></tr>
</table></td></tr>
<tr><td>992</td><td>Max phase frequency control</td><td>Reports the PWM frequency in 10 ns units. Read/Write. Differs from Delta Tau.</td></tr>
<tr><td>995</td><td>MACRO Ring Configuration/Status</td><td>Bit 7 set to 1 if syncing to the ring is enabled. Read only.</td></tr>
<tr><td>996</td><td>MACRO Node Activate Control</td><td>Node 14 and 15 are always enabled. Bits 0-3 are SW2 value, node number. Bits 24-27 are SW1 value, master address. Read only.</td></tr>
<tr><td>1018</td><td>Extended command access</td><td>See Extended Command Access (p. 52).</td></tr>
<tr><td>1020</td><td>Output compare control register</td><td>Compare module configuration.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0</td><td>Set to enable module.</td></tr>
<tr><td>1</td><td>Set to invert active state of output.</td></tr>
<tr><td>2</td><td>If set, toggle output on compare match. If clear, pulse output for programmable time.</td></tr>
<tr><td>3-4</td><td>Define mode of compare module.</td></tr>
<tr><td>5-31</td><td>Reserved for future use. Should be set to zero.</td></tr>
</table></td></tr>
<tr><td>1021</td><td>Output compare status register</td><td>Compare module status register.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0</td><td>Current value of compare output (read only).</td></tr>
<tr><td>1</td><td>Set when position matches compare register 0. Write 1 to clear.</td></tr>
<tr><td>2</td><td>Set when position matches compare register 1. Write 1 to clear.</td></tr>
<tr><td>3-31</td><td>Reserved.</td></tr>
</table></td></tr>
<tr><td>1022</td><td>Output compare period register</td><td>Read/Write compare period value.</td></tr>
<tr><td>1023</td><td>Output compare position register.</td><td>Read/Write compare position value.</td></tr>
</table>

6.4: Copley Drive Specific I-variables

The table below shows Copley specific I-variables used to access Copley drives on a MACRO ring. Use these decimal I-variable values to set Copley drives. Some of the most useful parameters are listed in this section. For a full list see the Copley Amplifier Parameter Dictionary (http://www.copleycontrols.com/Motion/Downloads/protocols.html).

Dec Ixx	Hex I$xx	Title	Description and Notes
1024	0x400	Current loop proportional gain	Current loop KP value (proportional gain).
1025	0x401	Current loop integral gain	Current loop KI value (integral gain).
1026	0x402	Current loop programmed value	This current will be used to command the amplifier when in state 1.Units: 0.01 A.
1027	0x403	Winding A current	Actual current measured at winding A. Units: 0.01 A.
1028	0x404	Winding B current	Actual current measured at winding B. Units: 0.01 A.

Dec Ixx	Hex I$xx	Title	Description and Notes
1045	0x415	Command motor current	This is the value that is presently being sent to the current loop. It may come from the programmed value, analog reference, velocity loop, etc. depending on the drive's state. Units: 0.01 A.
1053	0x41d	A/D reference input voltage	The analog command voltage after offset and deadband have been applied. Units: mV.
1054	0x41e	High voltage A/D reference	The voltage that is present on the high-voltage bus. Units: 0.1 V.
1056	0x420	Drive temperature A/D reading	Units: degrees C.
1057	0x421	Peak current limit	This value cannot exceed the peak current rating of the drive. Units: 0.01 A.
1058	0x422	Continuous current limit	This value should be less than the Peak Current Limit. Units: 0.01 A.
1059	0x423	Time at peak current limit	Units: ms.
1060	0x424	Desired drive state	**Value** – **Description** 0 – Disabled. 1 – The current loop is driven by the programmed current value. 2 – The current loop is driven by the analog command input. 3 – The current loop is driven by the PWM. 4 – The current loop is driven by the internal function generator. 5 – The current loop is driven by UV commands via PWM inputs.
1061	0x425	Limited motor current command	Output of current limiter (input to the current loop). Units: 0.01 A.
1136	0x470	Output pin configuration (OUT 1)	Data type is dependent on configuration and uses 1 to 5 words. The first word is a bit-mapped configuration value. The remaining words give additional parameter data used by the output pin. Typically the second and third words are used as a 32-bit bit mask to identify which bit(s) in the status register the output should follow. If any of the selected bits in the status register are set, then the output will go active. If none of the selected bits are set then the output will be inactive. Output pin 0 to 7 (OUT1 to 8) may be programmed as a sync output for use in synchronizing multiple amplifiers. In this configuration, the first word of this variable should be set to 0x0200 (i.e. only bit 9 is set), and the remaining words should be set to zero. Note that only output pin #0 has this feature. Attempting to program any other output pin as a sync output will have no effect. Here is the bit mapping of the first word: **Bits** – **Configuration** 0-4 – Define which internal register drives the output. The acceptable values for these bits are as follows: Value – Description 0 – Words 2 and 3 are used as a mask of the drive vent status register. When any bit set in the mask is also set in the drive vent status register, the output goes active.

Dec Ixx	Hex I$xx	Title	Description and Notes		
				1	Words 2 and 3 are used as a mask of the amplifier's latched event status register. When any bit set in the mask is also set in the latched event status register the output goes active and remains active until the necessary bit in the latched event status register is cleared.
				2	Puts the output in manual mode. Additional words are not used in this mode, and the output's state follows the value programmed in the parameter output states and program control.
				3	Words 2 and 3 are used as a mask of the trajectory status register. When any bit set in the mask is also set in the trajectory status register the output goes active.
				4	Output goes active if the axis position is between the low position specified in words 2 and 3 (bits 16-47) and the high position specified in words 4 and 5 (bits 48-80).
				5	Output goes active if the actual axis position crosses, with a low to high transition; the position specified in words 2 and 3 (bits 16-47). The output will stay active for number of ms specified in words 4 and 5 (bits 48-80).
				6	Same as 5 but for a high to low crossing.
				7	Same as 5 but for any crossing.
				8	Go active if motor phase angle (plus offset) is between 0 and 180 degrees. The offset is set using the first word of extra data in units of 32k/180 degrees.
			3-7	Reserved for future use.	
			8	If set, the output is active high. If clear, the output is active low.	
			9	If set, program the output as a sync output. This bit is reserved for all output pins except pin 0.	
			10-11	Reserved for future use.	
			12-13	Axis number for multi-axis amplifiers.	
			14-15	Usage depends on output function selected.	
1137	0x471	Output pin configuration (OUT 2)	Output pin #1 (output 2). See I-variable 1137, Output Pin Configuration, p. 30 for details.		
1138	0x472	Output pin configuration (OUT 3)	Output pin #2 (output 3). See I-variable 1137, Output Pin Configuration, p. 30 for details.		
1139	0x473	Output pin configuration (OUT 4)	Output pin #3 (output 4). See I-variable 1137, Output Pin Configuration, p. 30 for details.		
1140	0x474	Output pin configuration (OUT 5)	Output pin #4 (output 5). See I-variable 1137, Output Pin Configuration, p. 30 for details.		
1141	0x475	Output pin configuration (OUT 6)	Output pin #5 (output 6). See I-variable 1137, Output Pin Configuration, p. 30 for details.		
1142	0x476	Output pin configuration (OUT 7)	Output pin #6 (output 7). See I-variable 1137, Output Pin Configuration, p. 30 for details.		
1143	0x477	Output pin configuration (OUT 8)	Output pin #7 (output 8). See I-variable 1137, Output Pin Configuration, p. 30 for details.		

Dec Ixx	Hex I$xx	Title	Description and Notes
1144	0x478	Input pin configuration (IN 1)	Input pin #0 (input 1). Assigns a function to the input pin. All values not listed below are reserved for future use. Bits 8-11 may be used to pass parameters to the input pin functions. Bits 12-13 are used to select the axis on multi-axis drives.

Value	Function
0	No function.
1	Reserved for future use (no function).
2	Reset the drive on the rising edge of the input.
3	Reset the drive on the falling edge of the input.
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 on rising edge, disable while high.
11	Clear on falling edge, disable while low.
12	Reset on rising edge. Disable drive when high.
13	Reset on falling edge. Disable drive when low.
14	Home switch. Active high.
15	Home switch. Active low.
16	Disable drive when high.
17	Disable drive when low.
19	Synch input on falling edge.
20	Halt motor and prevent a new trajectory when high.
21	Halt motor and prevent a new trajectory when low.
22	Scale analog input when high.
23	Scale analog input when 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).
27	Count falling edges of input to indexer register.
28-35	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	Amp disabled hi with AC removed.
39	Amp disabled lo with AC removed.
40	Update trajectory on rising/falling edge.
41	Update trajectory on rising/falling edge.

Dec Ixx	Hex I$xx	Title	Description and Notes
			42 — Clear faults & event latch on rising/falling edge. 43 — Clear faults & event latch on rising/falling edge. All other values are reserved for future use. Bits 8-11 may be used to pass parameters to the input pin functions if necessary. Bits 12-13 are used to select the axis number on multi-axis amplifiers.
1145	0x479	Input pin configuration (IN 2)	See Input pin configuration (IN 1), p. 38 for details.
1146	0x47A	Input pin configuration (IN 3)	
1147	0x47B	Input pin configuration (IN 4)	
1148	0x47C	Input pin configuration (IN 5)	
1149	0x47D	Input pin configuration (IN 6)	
1150	0x47E	Input pin configuration (IN 7)	
1151	0x47F	Input pin configuration (IN 8)	
1152	0x480	Drive model number	
1153	0x481	Drive serial number	
1154	0x482	Drive peak current	Units: 0.01 A.
1155	0x483	Drive continuous current	Units: 0.01 A.
1156	0x484	Drive current corresponding to max A/D Reading	Units: 0.01 A.
1157	0x485	Drive PWM period	Units: 10 ns.
1158	0x486	Drive servo period	Servo loop update period as a multiple of the current loop period.
1160	0x488	Drive time at peak current	The maximum time for which the drive is rated to output peak current. Units: ms.
1172	0x494	Firmware version number	The version number consists of a major and a minor version number. The minor number is passed in bits 0-7; the major number is in bits 8-15. For example, the version 1.12 would be encoded 0x010C.

Dec Ixx	Hex I$xx	Title	Description and Notes
1184	0x4a0	Drive event status	This is a bit-mapped value:

Bits	Description
0	Short circuit detected.
1	Drive over temperature.
2	Over voltage.
3	Under voltage.
4	Motor temperature sensor active.
5	Encoder feedback 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	Tracking error.
19	Tracking warning.
20	Drive is currently in a reset condition.
21	Position has wrapped. The 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. A drive fault that was configured as latching has occurred. For information on latching faults, see the CME 2 User Guide.
23	Velocity limit has been reached.
24	Acceleration limit has been reached.
25	Tracking Window.
26	Home switch is active.
27	Set if trajectory is running or motor has not yet settled into position at the end of the move. Once the position has settled, the in motion bit won't be set until the next move starts.
28	Velocity window. Set if the absolute velocity error exceeds the velocity window value.
29	Phase not yet initialized. If the drive is phasing with no Halls, this bit is set until the drive has initialized its phase.
30	Command fault. PWM or other command signal not present. If ***Allow 100% Output*** option is enabled, by setting Bit 3 of Digital Input Command Configuration this fault will not detect a missing PWM command.

<table>
<tr><th>Dec
Ixx</th><th>Hex
I$xx</th><th>Title</th><th>Description and Notes</th></tr>
<tr><td>1185</td><td>0x4a1</td><td>Latched Event Status</td><td>This is a latched version of the event status word. Bits are set by the amplifier, but only cleared by a set command.
When writing to this value, any bit set in the passed data will cause the corresponding bit in the latched event flag to be cleared. For example, write a 1 to clear the short circuit detected bit in the latch.</td></tr>
<tr><td>1188</td><td>0x4a4</td><td>Drive fault latch</td><td>Bit-mapped to show which latching faults have occurred in the drive. When a latching fault has occurred, the fault bit (bit 22) of the Amplifier Event Status Register (see the Parameter Dictionary), is set. The cause of the fault can be read from this register.
To clear a fault condition, write a 1 to the associated bit in this register. The events that cause the drive to latch a fault are programmable.
<table>
<tr><th>Bits</th><th>Error Condition</th></tr>
<tr><td>0</td><td>Data flash CRC failure.</td></tr>
<tr><td>1</td><td>A/D offset out of range.</td></tr>
<tr><td>2</td><td>Short circuit detection.</td></tr>
<tr><td>3</td><td>Drive over temperature.</td></tr>
<tr><td>4</td><td>Motor over temperature.</td></tr>
<tr><td>5</td><td>Over voltage.</td></tr>
<tr><td>6</td><td>Under voltage.</td></tr>
<tr><td>7</td><td>Encoder power error.</td></tr>
<tr><td>8</td><td>Phasing error.</td></tr>
<tr><td>9</td><td>Tracking error.</td></tr>
<tr><td>10</td><td>Current limited by i^2t algorithm.</td></tr>
<tr><td>11</td><td>Unable to initialize internal amplifier hardware (FPGA).</td></tr>
<tr><td>12</td><td>Loss of command input.</td></tr>
<tr><td>13</td><td>Unable to initialize internal amplifier hardware (Co processor).</td></tr>
<tr><td>14</td><td>Safety circuit consistency check failure.</td></tr>
<tr><td>15</td><td>Unable to control motor current.</td></tr>
</table></td></tr>
<tr><td>1190</td><td>0x4a6</td><td>Input pin state</td><td>The 16-bit value returned by this command gives the current state (high/low) of the drive’s input pins after debouncing. The inputs are returned one per bit as mapped below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0</td><td>Programmable input pin 0 (In 1).</td></tr>
<tr><td>1</td><td>Programmable input pin 1 (IN 2).</td></tr>
<tr><td>2</td><td>Programmable input pin 2 (IN 3)..</td></tr>
<tr><td>3</td><td>Programmable input pin 3 (IN 4).</td></tr>
<tr><td>4</td><td>Programmable input pin 4 (IN 5).</td></tr>
<tr><td>5</td><td>Programmable input pin 5 (IN 6).</td></tr>
<tr><td>6</td><td>Programmable input pin 6 (IN 7).</td></tr>
<tr><td>7</td><td>Programmable input pin 7 (IN 8).</td></tr>
<tr><td>8</td><td>Programmable input pin 8 (IN 9).</td></tr>
<tr><td>9</td><td>Programmable input pin 9 (IN 10).</td></tr>
<tr><td>10</td><td>Programmable input pin 10 (IN 11).</td></tr>
<tr><td>11</td><td>Programmable input pin 11 (IN 12).</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Dec
Ixx</th><th>Hex
I$xx</th><th>Title</th><th>Description and Notes</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>12</td><td>Programmable input pin 12 (IN 13).</td></tr>
<tr><td>13</td><td>Programmable input pin 13 (IN 14).</td></tr>
<tr><td>14</td><td>Programmable input pin 14 (IN 15).</td></tr>
<tr><td>15</td><td>Programmable input pin 15 (IN 16).</td></tr>
</table>
</td></tr>
<tr><td>1192</td><td>0x4a8</td><td>Configuration word for digital control input pins</td><td>
Bits 0-7 of this value are used to configure the PWM inputs when PWM is used to drive the current or velocity loops:
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0</td><td>Signed/magnitude mode if set. 50% duty cycle if clear.</td></tr>
<tr><td>1</td><td>Invert PWM input signal if set</td></tr>
<tr><td>2</td><td>Invert sign input signal if set.</td></tr>
<tr><td>3</td><td>Allow 100% duty cycle if set.</td></tr>
<tr><td>4</td><td>If set, use parameter 0xB6 as deadband for PWM input.</td></tr>
<tr><td>5</td><td>If set, allow longer PWM periods (up to 50ms).</td></tr>
<tr><td>6</td><td>For drives based on the 8367 processor only; setting this bit will cause the hall offset (0x4f) to be added to the angle calculated in UV mode. For newer FPGA based products.</td></tr>
</table>
Bits 8-15 are used to configure the digital inputs for driving the position loop:
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>8 - 9</td><td>Specify the type of input signals. These bits should hold one of the following values.
<table>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Pulse and direction.</td></tr>
<tr><td>1</td><td>Up/down counters.</td></tr>
<tr><td>2</td><td>Master encoder inputs.</td></tr>
<tr><td>3</td><td>PWM input commands absolute position.</td></tr>
</table></td></tr>
<tr><td>12</td><td>Pulses are counted on their falling edge if this bit is clear, rising edge if set. This bit has no effect when the inputs are configured as encoder inputs.</td></tr>
<tr><td>13</td><td>Causes the direction of the input to be reversed. Works for all three modes.</td></tr>
<tr><td>14 - 15</td><td>Identify which input pins to use. Note that not all choices are valid for all drives.
<table>
<tr><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 (useful for Stepnet mostly).</td></tr>
</table></td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Dec Ixx</th><th>Hex I$xx</th><th>Title</th><th>Description and Notes</th></tr>
<tr><td>1193</td><td>0x4a9</td><td>Digital control input scaling factor</td><td>This value gives the amount of current to command at 100% PWM input. The scaling depends on what the PWM input is driving:
Current mode: 0.01 amps
Velocity (Junus): 0.01 RPM
Velocity (Accelus): 0.1 encoder counts/second
In position mode the scaling factor is a ratio of two 16-bit values. The first word passed gives the numerator and the second word gives the denominator. This ratio determines the number of encoder units moved for each pulse (or encoder count) input.
For example, a ratio of 1/3 would cause the motor to move 1 encoder unit for every three input steps.
When running in PWM position mode, the scaling factor is a single 32-bit integer which gives the range of commanded position in encoder counts. The minimum PWM duty cycle (MACRO parameter 0x53C) corresponds to an absolute position of 0, the maximum duty cycle (MACRO parameter 0x53D) corresponds to an absolute position equal to this scaling factor. Additionally, an offset may be added using MACRO parameter 0x50F. Current mode: 0.01 A</td></tr>
<tr><td>1196</td><td>0x4ac</td><td>‘Sticky’ version of event status register</td><td>This read-only variable is bit-mapped in exactly the same way as the event status, however instead of giving the present status of the amplifier, it indicates any bits in the event status that have been set since the last reading.
This is similar to the latched event status (variable 0x4A1), however it is not necessary to explicitly clear this register it is automatically cleared on every read.</td></tr>
<tr><td>1282</td><td>0x502</td><td>Network status word</td><td>This register gives a status of the network.
Bit mapped as follows.
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>0</td><td>Set if the MACRO network is detected,</td></tr>
<tr><td>1</td><td>Set if the drive is being disabled by the MACRO master.</td></tr>
<tr><td>2</td><td>Set if the MACRO network has been broken (i.e. once detected but now gone).</td></tr>
<tr><td>3</td><td>Set on heartbeat error.</td></tr>
<tr><td>4-15</td><td>Reserved.</td></tr>
</table></td></tr>
<tr><td>1313</td><td>0x521</td><td>Network options</td><td>This bit-mapped parameter is used to configure the amplifier's network.
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>0</td><td>If set, position data sent over the MACRO network is shifted up 5 bits for compatibility with Delta-Tau controllers.</td></tr>
<tr><td>1</td><td>If set, the drive will be disabled on startup until it is enabled through the MACRO interface. If clear, the drive can be used without the MACRO interface connected until it starts receiving MACRO messages.</td></tr>
<tr><td>2</td><td>If set, return the primary encoder index state (high/low) in the home status bit of the MACRO status word. If clear, the state of any general purpose input configured as a home input will be used.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Dec Ixx</th><th>Hex I$xx</th><th>Title</th><th>Description and Notes</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>3</td><td>If set, the drive will attempt to synchronize its current loop to the MACRO ring. The ring period must be an integer multiple of the drive's PWM period (parameter 0x85).</td></tr>
<tr><td>4-7</td><td>Identify the type of data sent from the drive in the first 16-bit MACRO register. 0=digital inputs, 1=secondary analog input 1, 2-15=reserved</td></tr>
<tr><td>8-11</td><td>Identify the type of data sent in the second 16-bit MACRO register:
0 – primary analog input.
1 – motor encoder position.
2 – load encoder position.
3 – reserved.</td></tr>
<tr><td>12-15</td><td>Reserved.</td></tr>
</table>
</td></tr>
<tr><td>1317</td><td>0x525</td><td>MACRO specific capture configuration</td><td>Configures the MACRO drive’s encoder capture circuit for accurate homing or position, as follows:
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>0-3</td><td>Type of capture to use.
<table>
<tr><td>Value</td><td>Description</td></tr>
<tr><td>0</td><td>Capture on edge of encoder index.</td></tr>
<tr><td>1</td><td>Capture using a general purpose input pin.</td></tr>
<tr><td>2-15</td><td>Reserved.</td></tr>
</table>
</td></tr>
<tr><td>4-7</td><td>Input pin number to use if using capture type 1.</td></tr>
<tr><td>8</td><td>Active level; high if clear (0), low if set (1).</td></tr>
<tr><td>9</td><td>If set, capture is re-enabled immediately when the capture position is read (using I-variable 921). If clear, capture is only re-enabled on an explicit clear instruction.</td></tr>
<tr><td>10</td><td>If set, the passive load encoder (if configured) will be captured. Passive load encoder currently only supports capture type 1 (general purpose input).</td></tr>
<tr><td>11-15</td><td>Reserved.</td></tr>
</table>
</td></tr>
<tr><td>1322</td><td>0x52a</td><td>Motor encoder options</td><td>Specifies various configuration options for the motor encoder. The mapping of option bits to function depends on the encoder type.
<table>
<tr><th colspan="2">Quadrature</th></tr>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0</td><td>If set, ignore differential signal errors (if detected in hardware.</td></tr>
<tr><td>1</td><td>If set, select single ended encoder inputs (if available in hardware).</td></tr>
<tr><td>2</td><td>If set, ignore differential signals errors on encoder index only (if supported by hardware).</td></tr>
<tr><th colspan="2">EnDat (type 11)</th></tr>
<tr><td>0 - 5</td><td>Number of bits of single turn data available from encoder.</td></tr>
<tr><td>8 - 12</td><td>Number of bits of multi-turn data available from encoder.</td></tr>
<tr><td>16</td><td>Set if analog inputs are supplied by encoder.</td></tr>
</table>
</td></tr>
</table>

Dec Ixx	Hex I$xx	Title	Description and Notes

17	If set, use multi-mode port.
18	If set, read position using EnDat 2.2 style commands rather than the default 2.1 style.
20 -23	Number of least significant bits of the encoder reading to discard.
SSI (type 12)	
0 - 5	Number of bits of position data available.
8 - 10	Number of extra status bits sent after position data.
12	If set, ignore the first bit of data sent by the 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 LSB first.
16 - 21	Encoder bit rate in 100 kHz units. If zero, default to 1MHz.
22	If set, use setting of encoder counts/rev to determine how many data bits to use.
24	If set, first bit sent is 'data valid' bit.
Encoder type 14	
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).
28	Bit rate (set for 4 Mbit, clear for 2.5 Mbit).
30	If set, treat encoder battery errors as warnings.
BiSS (type 13)	
0 - 5	Number of bits of single turn data.
8 - 12	Number of bits of multi-turn data.
15	If set, ignore the multi-turn data from the encoder. Useful when the encoder sends zero bits before the reading
16	Set for mode-C encoder format.
20	Set if 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	Set if encoder error bit is transmitted before warning bit. Clear for warning bit sent first.
24 - 26	Number of alignment bits (reserved bits sent before position info).
28	Use multi-mode port if set. If clear use primary encoder interface.
30	Set for 2.5MHz baud rate, clear for 4MHz baud rate.

<table>
<tr><th>Dec
Ixx</th><th>Hex
I$xx</th><th>Title</th><th>Description and Notes</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th colspan="2">Gurley virtual absolute (type 17)</th></tr>
<tr><td>0</td><td>Invert sine/cosine signals if set.</td></tr>
<tr><td>1</td><td>Invert virtual absolute signal if set.</td></tr>
<tr><td>2</td><td>If set, use custom interface board (customer specific).</td></tr>
<tr><td>3</td><td>If set, use the encoder digital index input for VABS. If clear the encoder analog index (if available) is used.</td></tr>
<tr><td>8</td><td>If set, switch from algorithmic phase init to encoder based phasing as soon as absolute position is found.</td></tr>
<tr><td>9</td><td>If set, treat any VABS warnings as an encoder fault. If clear
these warnings set status bits but aren’t treated as encoder
errors.</td></tr>
<tr><th colspan="2">Kawasaki absolute (type 18)</th></tr>
<tr><td>28</td><td>Use multi-mode port if set. If clear use primary encoder interface.</td></tr>
<tr><th colspan="2">S2 custom</th></tr>
<tr><td>0 - 4</td><td>Number of bits of single turn position data / rev.</td></tr>
<tr><td>8</td><td>Set for incremental encoders, clear for absolute.</td></tr>
<tr><td>9</td><td>Use multi-mode port if set. If clear use primary encoder interface.</td></tr>
<tr><td>10</td><td>If set, treat encoder battery errors as warnings.</td></tr>
</table>
</td></tr>
<tr><td>1325</td><td>0x52b</td><td>Load encoder options</td><td>Same as 1322 (0x52a), but affects load or position encoder.</td></tr>
</table>

Dec Ixx	Hex I$xx	Title	Description and Notes
1326	0x52e	Motor encoder status	This parameter gives additional status information for the encoder. Bits set in the status word are latched and cleared when the status value is read. The format of this status word is dependent on the encoder type. Many error bits are taken directly from encoder data stream. For a full description of what these error bits mean, please consult the encoder manufacturer.

Bits	Description
Quadrature	
0	Only used for Yaskawa Sigma-I wire saving incremental encoders. Set on startup if the encoder did not transmit hall information successfully.
1	Set on bad differential signal levels on any of the encoder inputs.
BiSS (type 13)	
0	CRC error on data received from encoder.
1	Encoder failed to transmit data to amp.
2	Error bit on encoder stream is active.
3	Warning bit on encoder stream is active.
4	Encoder transmission delay is too long.
EnDAT (type 11)	
0	CRC error on data received from encoder.
1	Failed to detect encoder connected to amplifier.
2	Error bit on encoder stream is active.
3	Encoder failed to respond to request for position.
SSI (type 12)	
0 - 6	Fault flags returned from encoder.
15	Encoder data invalid bit set.
Tamagawa & Panasonic (type 14)	
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.

<table>
<tr><th>Dec
Ixx</th><th>Hex
I$xx</th><th>Title</th><th>Description and Notes</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th colspan="2">Sanyo Denki & Harmonic Drives (encoder type 14)</th></tr>
<tr><td>0</td><td>Battery warning reported by encoder.</td></tr>
<tr><td>1</td><td>Battery error reported by encoder.</td></tr>
<tr><td>3</td><td>Over speed reported by encoder.</td></tr>
<tr><td>4</td><td>Memory error reported by encoder.</td></tr>
<tr><td>5</td><td>STERR reported by encoder.</td></tr>
<tr><td>6</td><td>PSERR reported by encoder.</td></tr>
<tr><td>7</td><td>Busy error reported by encoder.</td></tr>
<tr><td>8</td><td>Memory busy reported by encoder.</td></tr>
<tr><td>9</td><td>Over temperature reported by encoder.</td></tr>
<tr><td>15</td><td>CRC error on data received from encoder.</td></tr>
<tr><th colspan="2">Harmonic Drive (type 15)</th></tr>
<tr><td>0</td><td>System error reported by encoder.</td></tr>
<tr><td>1</td><td>Overflow error reported by encoder.</td></tr>
<tr><td>2</td><td>Mode error reported by encoder.</td></tr>
<tr><td>3</td><td>Battery error reported by encoder</td></tr>
<tr><td>4</td><td>CRC error on data received from encoder</td></tr>
<tr><td>5</td><td>No data received from encoder on read.</td></tr>
<tr><th colspan="2">Gurley virtual absolute (type 17)</th></tr>
<tr><td>0</td><td>Amplitude of sine/cosine signals is out of range.</td></tr>
<tr><td>1</td><td>Encoder power current limited.</td></tr>
<tr><td>2</td><td>Encoder moving too fast during initialization.</td></tr>
<tr><td>3</td><td>Missing trigger signal (only occurs when using custom interface hardware).</td></tr>
<tr><td>4</td><td>Virtual absolute signal changed state at incorrect time.</td></tr>
<tr><td>5</td><td>Invalid virtual absolute data received.</td></tr>
<tr><td>6</td><td>Encoder has not finished initializing position.</td></tr>
<tr><th colspan="2">Kawasaki absolute (type 18)</th></tr>
<tr><td>0</td><td>Busy bit from encoder set.</td></tr>
<tr><td>1</td><td>ABSALM bit from encoder set.</td></tr>
<tr><td>2</td><td>INPALM bit from encoder set.</td></tr>
<tr><td>8</td><td>CRC error on data received from encoder.</td></tr>
<tr><th colspan="2">S2 custom (type 19)</th></tr>
<tr><td></td><td>Battery error alarm bit from encoder.</td></tr>
<tr><td></td><td>Encoder error alarm bit from encoder.</td></tr>
<tr><td></td><td>Battery warning alarm bit from encoder.</td></tr>
<tr><td></td><td>Absolute error alarm bit from encoder.</td></tr>
<tr><td></td><td>Over speed error alarm bit from encoder.</td></tr>
<tr><td></td><td>Over heat error alarm bit from encoder.</td></tr>
<tr><td></td><td>CRC error on data received from encoder</td></tr>
<tr><td></td><td>Encoder not responding to queries from drive.</td></tr>
</table>
</td></tr>
<tr><td>1327</td><td>0x52f</td><td>Load encoder status</td><td>Same as 1326 (0x52e), but for a load encoder.</td></tr>
</table>

<table>
<tr><th>Dec Ixx</th><th>Hex I$xx</th><th>Title</th><th>Description and Notes</th></tr>
<tr><td>1328</td><td>0x530</td><td>RMS current calculation period</td><td>This sets the period (in milliseconds) over which the RMS current is calculated. If this value is set to zero, then the RMS current will be updated each time it is read for the period since the last read. In this case, the RMS current must be read at least once every 65536 current loop periods (about every 4 seconds) for the returned RMS values to be accurate. Units: milliseconds.</td></tr>
<tr><td>1329</td><td>0x531</td><td>RMS current</td><td>RMS current over the period set in I-variable 0x530. Units: 0.01 A.</td></tr>
<tr><td>1330</td><td>0x532</td><td>User current limit running sum</td><td>In 0.01% units. (i.e. 0 to 10000).</td></tr>
<tr><td>1331</td><td>0x533</td><td>Drive current limit running sum</td><td>In 0.01% units. (i.e. 0 to 10000).</td></tr>
<tr><td>1332</td><td>0x534</td><td>D/A converter configuration</td><td>This parameter sets the mode for the D/A converter on drives so equipped.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0-3</td><td>Define the mode of the D/A converter.</td></tr>
<tr><td>16-17</td><td>Identify the axis associated with the D/A converter.</td></tr>
<tr><th>Mode</th><th>Description</th></tr>
<tr><td>0</td><td>Manual configuration (set using parameter 0x535).</td></tr>
<tr><td>1</td><td>Actual current of configured axis.</td></tr>
</table></td></tr>
<tr><td>1333</td><td>0x535</td><td>D/A converter output value</td><td>For drives that support an auxiliary D/A converter, this parameter sets the output value in mV units when the D/A is in manual mode. In other modes, the current value being output on the D/A can be read here.</td></tr>
<tr><td>1337</td><td>0x539</td><td>Safety circuit control/status</td><td>Status of a drive’s safety circuit. This parameter allows the status of the safety circuit built into some drives to be queried. For drives without a safety circuit, this parameter is reserved.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0</td><td>Set when safety input 0 is preventing the drive from enabling.</td></tr>
<tr><td>1</td><td>Set when safety input 1 is preventing the drive from enabling.</td></tr>
<tr><td>8</td><td>This read/write bit can be used to force the ‘drive is unsafe’ output of the safety circuit to go active for testing purposes. Write 1 to force.</td></tr>
</table></td></tr>
<tr><td>1413</td><td>0x585</td><td>Compare module config</td><td>This parameter is bit-mapped as follows:
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0</td><td>Set to enable module.</td></tr>
<tr><td>1</td><td>Set to invert active state of output.</td></tr>
<tr><td>2</td><td>If set, toggle output on compare match. If clear, pulse output for programmable time.</td></tr>
<tr><td>3-4</td><td>Define mode of compare module.</td></tr>
<tr><td>5-31</td><td>Reserved for future use. Should be set to zero.</td></tr>
</table>
See Compare Position Window Example (p. 57).</td></tr>
</table>

<table>
<tr><th>Dec
Ixx</th><th>Hex
I$xx</th><th>Title</th><th>Description and Notes</th></tr>
<tr><td>1414</td><td>0x586</td><td>Compare status register</td><td>This parameter is bit-mapped as follows:
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0</td><td>Current value of compare output (read only).</td></tr>
<tr><td>1</td><td>Set when position matches compare register 0. Write 1 to clear.</td></tr>
<tr><td>2</td><td>Set when position matches compare register 1. Write 1 to clear.</td></tr>
<tr><td>3-31</td><td>Reserved.</td></tr>
</table>
See Compare Position Window Example (p. 57).</td></tr>
<tr><td>1415</td><td>0x587</td><td>Compare value A (0)</td><td>See Compare Position Window Example (p. 57).</td></tr>
<tr><td>1416</td><td>0x588</td><td>Set compare value B (1)</td><td>See Compare Position Window Example (p. 57).</td></tr>
</table>

6.5: Node 14 (ASCII) Specific Commands

These ASCII commands may be used with node 14 communications.

<table>
<tr><th>Ixx</th><th>Title</th><th>Description</th></tr>
<tr><td>?</td><td>Station global status</td><td><table><tr><th>Bits</th><th>Description</th></tr><tr><td>0</td><td>Set when a current limit event occurs.</td></tr><tr><td>2</td><td>Set when a short circuit event occurs.</td></tr><tr><td>4</td><td>Set when a motor over temp event occurs.</td></tr><tr><td>6</td><td>Set when a drive over temp event occurs.</td></tr><tr><td>8</td><td>Set when a feedback error event occurs.</td></tr><tr><td>20</td><td>Set when an under voltage event occurs.</td></tr><tr><td>21</td><td>Set when an over voltage event occurs.</td></tr><tr><td>24</td><td>Set if amp fault occurs.</td></tr><tr><td>25</td><td>Set if MACRO network fault occurs. (Set on break detected).</td></tr></table></td></tr>
<tr><td>$$$</td><td>Station reset to saved parameters</td><td>The drive software disables, clears all faults and restores parameters from flash.</td></tr>
<tr><td>$$$**</td><td>Station re-initialize to default parameters</td><td>The drive software disables, clears all faults and restores parameters from flash. (Same as $$$).</td></tr>
<tr><td>BKUP</td><td>Report saved I-variables</td><td>Backs up all parameters that are saved in flash memory.</td></tr>
<tr><td>CID</td><td>Report card ID number</td><td>Reports the hardware type number.</td></tr>
<tr><td>CLRF</td><td>Clear station faults</td><td>Clears all faults on the drive.</td></tr>
<tr><td>STN</td><td>Station number</td><td>Read/Write, saved to flash. This parameter behaves as outlined by Delta Tau. It is used for ASCII communications.</td></tr>
<tr><td>DATE</td><td>Report firmware date</td><td>Reports firmware build date in format of MM/DD/YYYY.</td></tr>
<tr><td>MACSTAN</td><td>Station Initialization Variable</td><td>Reports the value of the initialization variable n.
n is the station number being initialized. For example to talk to macro station number 1, enter MACSTA1.</td></tr>
<tr><td>SAVE</td><td>Save station I-variables</td><td>The drive will save all parameters settings in volatile RAM into non-volatile flash memory.</td></tr>
<tr><td>SID</td><td>Reports serial ID number</td><td>Reports the amp serial number.</td></tr>
<tr><td>TYPE</td><td>Report MACRO station type</td><td>Reports “Copley XML” or “Copley AMP”, depending on the drive.</td></tr>
<tr><td>VERS</td><td>Report firmware version</td><td>Reports the drive's firmware build version number.</td></tr>
<tr><td>VID</td><td>Report vendor ID number</td><td>Reports “7 Copley”.</td></tr>
</table>

6.6: Extended Command Access

Copley drives implement a special I-variable, 1018, which can be used to access parameters that are longer than 48 bits.

When I-variable 1018 is written, the upper 16 bits of the value written to it are taken as a command code, and the lower 32-bits of data are treated as data. Through the use of multiple such writes, data can be written to an internal buffer and then this data can be used to set long drive parameters. Similarly, long parameters can be read out of the drive by first issuing a command which causes the parameters contents to be copied into the internal buffer, then read using I-variable 1018 (which allows the contents of this buffer to be read out).

6.6.1: Writing to I-Variable 1018

When writing to I-variable 1018, the upper 16 bits of the value written are taken to be a command code. The following commands are currently supported:

Code	Description
0	Clears the internal buffer and sets the internal pointer to the first buffer location. Bits 0-31 of the value written to I-1018 are ignored for this command. This command also clears the internal error code from the last executed extended command.
1	Sets an internal pointer to the location passed in bits 16-31. If the value held in bits 16-31 is greater then the length of data currently written to the buffer, then the pointer is set to the end of the buffer.
2	First clears the internal buffer, then writes the value passed in bits 16-31 to the first buffer location, and the value passed in bits 0-15 in the second buffer location. After this command finishes, the buffer will hold two words of data, and the internal pointer will point to the end of the buffer.
3	Appends two words of data to the end of the buffer. The data value passed in bits 16-31 will be stored in the buffer position currently addressed by the internal pointer, and the value passed in bits 0-15 will be stored in the following position. After the command completes, the buffer length will be set so that the two new words are at the end of the buffer, and the buffer position pointer will be set to the end of the buffer.
4	Set a drive parameter to equal the value currently held in the internal buffer. If the number of words written to the buffer are greater then required for this parameter, then any extra data will be ignored. If there is not enough data in the buffer to set the parameter, then an error will result. The ID number of the parameter to be set should be passed in bits 16-31 of the value written to I-1018. Note that this is the native parameter ID number, not the I-variable number (which is 1024 larger than the normal parameter ID number). Parameters on Copley drives exist in either RAM or Flash memory, or both. The parameter number used to access the parameter value uses one bit to identify which memory space is being referenced. When using the I-1018 technique to read/write parameters over MACRO, the value of bit 28 identifies which page of memory should be accessed. If bit 28 is clear, then the RAM version of the parameter will be set. If bit 28 is set, then the flash version of the parameter is set. For example, drive parameter number 0x006B contains a set of bi-quad filter coefficients which control a filter used in the drive's velocity loop. This parameter exists both in working RAM and also in flash memory. To set this parameter, one would first upload the new value for the parameter to the internal buffer using the commands described above, then the parameter can be set using one of the following commands: ms0,i1018 = $0004006B0000 ; set the RAM version ms0,i1018 = $0004106B0000 ; set the Flash version
5	Read the value of an drive parameter and store the result to the internal buffer. The contents of this buffer can then be read back by reading I-1018. Bits 16-31 of the value written to I-1018 should hold the parameter number to read. This parameter number is formatted the same as for command code 4 described above.

Code	Description
6	Execute a trajectory command. Trajectory commands are used when the drive is running in position mode using its internal trajectory generator. There are a number of different trajectory commands which allow a new move to be started, a homing routine to be started and a move in progress to be aborted. The details of the various commands are given in the programmer's guide. Bits 16-31 of the value passed to I-1018 identify the trajectory command to execute (move, abort, home, etc.).
7	Execute a generic serial port command. This allows nearly any command that can be executed over the binary serial interface to be used with MACRO. The command opcode is passed in bits 16-31. Any data in the internal buffer will be treated as though it were data passed with the serial port command. On return, the internal buffer will be filled with any response data returned by the command.

6.6.2: Reading from I-Variable 1018

When reading from I-variable 1018, the value returned will either contain a negative error code, or will return data from the internal buffer.

If an error occurred on the last extended command executed, the value returned when reading I-variable 1018 will contain this error code multiplied by -1.

If no error occurred on the last command, the value returned when reading I-variable 1018 will be positive. The 48-bit response read from I-variable 1018 will contain the following information:

Bits	Contents
47	Always clear (positive value).
46-40	Buffer position of first returned word.
39-32	Number of words of data currently stored in internal buffer.
31-16	First word of data returned from the internal buffer
15-0	Second word of data returned from the internal buffer

Each time I-variable 1018 is read, the internal buffer pointer is incremented by two positions until the end of the internal buffer is reached. This allows the entire contents of the internal buffer to be easily read out by repeatedly reading I-variable 1018.

For example, drive parameter 0x0092 is called the 'axis name' parameter. This parameter is 40 bytes long and normally contains an ASCII string which can be used to describe the axis. Like most string type parameters on the Copley drive, this parameter is stored only in flash memory. There is no version of the parameter stored in RAM. Since the parameter is stored in flash, it will need to be accessed using parameter number 0x1092 (bit 12 is set indicating the flash page).

Reading the Parameter

To read this parameter, first send a read command to i1018:

ms0,i1018 = $000510920000

This sets a command code 5 (read parameter), with a parameter ID of 0x1092 (axis name in flash).

Then the returned value can be read by reading i1018 repeatedly:

ms0,i1018

$0014582D6178

Here, the value returned gave the buffer position (0x00) in the upper 8 bits, the total number of words of data in the buffer (0x14 = 20) in the next 8 bits, and 32 bits of data from the buffer (0x58 0x2D 0x61 0x78) in the remaining bits.

ms0,i1018

$021469730000

Continuing to read from the buffer causes the buffer position to increase by 2 words on each read. More ASCII data is returned from each read of the buffer.

ms0,i1018

$041400000000

ms0,i1018

$061400000000

ms0,i1018

$081400000000

ms0,i1018

$0A1400000000

ms0,i1018

$0C1400000000

ms0,i1018

$0E1400000000

ms0,i1018

$101400000000

ms0,i1018

$121400000000

ms0,i1018

$141400000000

The complete axis name that was set is available once all the data from the buffer has been read.

In hex this is 0x58 0x2D 0x61 0x78 0x69 0x73 0x00 0x00...

In ASCII, this name is “X-axis”

Example of Setting an Axis Name to ABCDEF

To set the axis name to a value of “ABCDEF” the following set of commands may be used:

	Enter	Description
1	ms0,i1018=$000241424344	Write the first 4 characters (in hex they are 0x41 0x42 0x43 0x44). Command code 2 is used to do this since it resets the buffer before adding the passed data.
2	ms0,i1018=$000345460000	Write the next 4 characters using command code 3 (which appends data to the end of the buffer).
3	ms0,i1018=$000410920000	For string parameters we don't have to append all the extra zeros. If less than 40 bytes is written, the missing bytes are automatically assumed to be zero. I'll therefore now write this value to the axis name parameter using command code 4. After this command executes, the axis name parameter will be set to the string “ABCDEF”

6.7: Position Capture Examples

This section shows three examples of setting I-variables for position captures.

6.7.1: Example 1

The example below shows how to set a capture position once from input 1 (to auto re-enable capture set bit 9 of I-variable 1317). See I-variable 1317 (p.44), for bit information.

	Command	Description
1	ms1,i1317=$331	Set I-variable 1317 to capture the input position.
2	#2j+	Jog motor 2.
	When input 4 transitions from Hi to Lo, a capture occurs.	
3	ms1,i921	Read I-variable 921 for the captured value.

6.7.2: Example 2

The example below shows how to set a capture position once from input 1, using a home command. See I-variable 1317 (p.44), for bit information.

	Command	Description
1	ms1,i1317=$331	Set I-variable 1317 to capture the input position.
2	#2hm	Start home.
	When input 4 transitions from Hi to Lo, a capture occurs.	
3	ms1,i921	Read I-variable 921 for the captured value.

6.7.3: Example 3

The example below shows how to set a capture position once from input 1, using a home to index pulse command. See I-variable 1317 (p.44), for bit information.

	Command	Description
1	ms1,i1317=0	Set I-variable 1317 to capture the input position.
2	#2hm	Home motor 2 to index pulse
	When input 4 transitions from Hi to Lo, a capture occurs.	
3	ms1,i921	Read I-variable 921 for the captured value.

6.8: Compare Position Window Example

This section shows an example of setting I-variables for a compare position window on output 2, with a position range of 100 to 1000 counts.

6.8.1: Example

	Command	Description
1	ms0,i1137=$1000000000	Set output two to compare mode.
2	ms0,i1413=$11	Set compare module configuration.
3	ms0,i1415=100	Set compare value A (Value 0).
4	ms0,i1416=1000	Set compare value B (Value 1).
5	ms0,i1414	Read the compare status register and check that the 0 bit value is 0.
6	Move the motor to a position that falls between 100 and 1000 counts.	
7	ms0,i1414	Read the compare status register and check that bit 0 was correctly set to 1.
8	ms0,i920	Read absolute position.
	Additionally check that the logic level of output 2 has toggled.	

MACRO Network User Guide

P/N 16-01052

Revision 00

February 2013