

USER MANUAL

Accessory 68M

MACRO UR Protected/OPTO (Sinking 24in/24out)

3Ax-603747-xUxx

December 9, 2009



DELTA TAU
Data Systems, Inc.

NEW IDEAS IN MOTION ...

Copyright Information

© 2009 Delta Tau Data Systems, Inc. All rights reserved.

This document is furnished for the customers of Delta Tau Data Systems, Inc. Other uses are unauthorized without written permission of Delta Tau Data Systems, Inc. Information contained in this manual may be updated from time-to-time due to product improvements, etc., and may not conform in every respect to former issues.

To report errors or inconsistencies, call or email:

Delta Tau Data Systems, Inc. Technical Support

Phone: (818) 717-5656

Fax: (818) 998-7807

Email: support@deltatau.com

Website: <http://www.deltatau.com>

Operating Conditions

All Delta Tau Data Systems, Inc. motion controller products, accessories, and amplifiers contain static sensitive components that can be damaged by incorrect handling. When installing or handling Delta Tau Data Systems, Inc. products, avoid contact with highly insulated materials. Only qualified personnel should be allowed to handle this equipment.

In the case of industrial applications, we expect our products to be protected from hazardous or conductive materials and/or environments that could cause harm to the controller by damaging components or causing electrical shorts. When our products are used in an industrial environment, install them into an industrial electrical cabinet or industrial PC to protect them from excessive or corrosive moisture, abnormal ambient temperatures, and conductive materials. If Delta Tau Data Systems, Inc. products are directly exposed to hazardous or conductive materials and/or environments, we cannot guarantee their operation.

REVISION HISTORY

REV.	DESCRIPTION	DATE	CHG	APPVD
1	UPDATED MANUAL FOR RELEASE. NEW 24V CONNECTOR	2/20/2007	C. PERRY	R.NADDAF
2	UPDATED 16-BIT ADC OPTION	12/08/2009	C.PERRY	S.FIERRO

Table of Contents

INTRODUCTION	1
Options	1
HARDWARE REFERENCE SUMMARY	3
Product Layout	3
Connectors and Indicators	4
Inputs and Outputs LED Indicators	4
Status LED Indicators.....	4
7-segment LED Indicator.....	4
Relay Status LED Indicators.....	4
USB Connector	4
24V Input Connector.....	4
MACRO Link Connectors	4
DB-15 Option-1 Connector	4
Inputs and Outputs Terminal Blocks.....	4
Connections Example: Sourcing Inputs and Sinking Outputs.....	5
Connections Example: Sinking Inputs and Sinking Outputs.....	5
Connections Example: OPT-1 Amplifier Signals	5
JUMPERS DESCRIPTION.....	7
E1: Watchdog Timer Disable	7
E2: CPU Mode Operation	7
E3: USB Port Serial Baud Rate	7
E4: MACRO Type Connection	7
JP1: T	7
JP2 – JP6: Reserved for Future Use	7
JP7: Re-initialization	7
CONNECTORS DESCRIPTION	9
USB Universal Serial Bus Port	9
24VDC Input Edge Connector	9
MACRO RJ-45 Copper Connectors	9
Inputs Connector: 30-pin Terminal Block.....	10
Outputs Connector: 30-pin Terminal Block	11
OPT-1: DB-15 Connector	12
SOFTWARE SETUP	13
ASCII Ring Order Initial Binding of the ACC-68M Station.....	13
Establishing Communications with the ACC-68M Station	14
Using ACC-68M Inputs and Outputs.....	15
Using the ACC-68M ADC	16
Using the ACC-68M DAC Output	17
Using the ACC-68M Amplifier Enable Outputs	18
FLAGs Command Register	19
FLAGs Status Register.....	19
MACRO ASCII Communication Reference:	19
Firmware Updates	21
ACC-68M MACRO STATION MI-VARIABLE REFERENCE	22
Global MI-Variables	22
MS{anynode},MI0 Station Firmware Version (Read Only).....	22
MS{anynode},MI1 Station Firmware Date (Read Only)	22
MS{anynode},MI2 Station ID and User Configuration Word	22
MS{anynode},MI3 (Reserved for Future Use).....	23
MS{anynode},MI4 Station Status Word (Read Only)	23

<i>MS{anynode},MI5</i> Ring Error Counter	23
<i>MS{anynode},MI6</i> Maximum Permitted Ring Errors in One Second	24
<i>MS{anynode},MI7</i> (Reserved for Future Use).....	24
<i>MS{anynode},MI8</i> MACRO Ring Check Period.....	24
<i>MS{anynode},MI9</i> MACRO Ring Error Shutdown Count	24
<i>MS{anynode},MI10</i> MACRO Sync Packet Shutdown Count	25
<i>MS{anynode},MI11</i> Station Order Number	25
<i>MS{anynode},MI12</i> Card Identification	25
<i>MS{anynode},MI13</i> (Reserved for Future Use).....	25
<i>MS{anynode},MI14</i> (Reserved for Future Use).....	26
<i>MS{anynode},MI15</i> Enable MACRO PLCC.....	26
MACRO IC MI-Variables.....	26
<i>MS{anynode},MI176</i> MACRO IC Base Address.....	26
<i>MS{anynode},MI177</i> MACRO IC Address for Node 14	26
<i>MS{anynode},MI178</i> MACRO IC Address for Node 15	26
<i>MS{anynode},MI181–MI188</i> MACRO Channels 1 - 8 Address	26
MACRO IC I/O Transfer MI-Variables	26
<i>MS{anynode},MI198</i> Direct Read/Write Format and Address.....	27
<i>MI198</i> Format Digits	28
<i>MS{anynode},MI199</i> Direct Read/Write Variable.....	30
Global MACRO Status MI-Variables	30
<i>MS{anynode}, MI203</i> Phase Period	30
<i>MS{anynode}, MI204</i> Phase Execution Time	30
<i>MS{anynode}, MI205</i> Background Cycle Time	30
<i>MS{anynode}, MI206</i> Maximum Background Cycle Time	30
<i>MS{anynode}, MI208</i> User Ram Start.....	30
MACRO IC MI-Variables.....	31
<i>MS{anynode},MI942</i> ADC Strobe Word for ADC1 and ADC2 Inputs	31
MACRO IC Setup MI-Variables.....	31
<i>MS{anynode},MI970–MI973</i> (Reserved for Future Use)	31
<i>MS{anynode},MI974</i> Station Display Status (Read Only).....	31
<i>MS{anynode},MI977</i> Motor Nodes Reporting Ring Break.....	31
<i>MS{anynode},MI978–MI986</i> (Reserved for Future Use).....	31
MACRO IC I/O Control and Initialize	32
<i>MS{anynode},MI987</i> TBD	32
<i>MS{anynode},MI988</i> TBD	32
<i>MS{anynode},MI989</i> TBD	32
MACRO IC MI-Variables.....	32
<i>MS{anynode},MI992</i> MaxPhase Frequency Control.....	32
<i>MS{anynode},MI993</i> Hardware Clock Control Handwheel Channels.....	32
<i>MS{anynode},MI994</i> PWM Deadtime / PFM Pulse Width Control for Handwheel.....	34
<i>MS{anynode},MI995</i> MACRO Ring Configuration/Status	35
<i>MS{anynode},MI996</i> MACRO Node Activate Control	35
<i>MS{anynode},MI997</i> Phase Clock Frequency Control	36
<i>MS{anynode},MI998</i> Servo Clock Frequency Control.....	37
<i>MS{anynode},MI999</i> Handwheel DAC Strobe Word (Not used).....	37
Other Acc-68M MACRO Station Mm & MP-Variables.....	37
ACC-68M MACRO STATION MACPLCCS.....	38
Requirements.....	38
Arithmetic Data Types	38
MACRO MI Integer Variables (n = 0 – 1099)	38
MACRO MM and MP Integer Variables (n = 0 – 511)	38
<i>MACROPlcc Ln</i> Integer Variables (n = 0 – 511)	38
Direct Memory Addressing for Integer Ln & Ln[] Variable Definitions	38
Standard MACRO Program Commands	38

Valid Math, Assignment and Conditional Operators	38
Valid Expressions and Arrays	39
Ln Arrays Definition Examples.....	39
<i>Example Program</i>	39
MACRO PLCC Code Memory	39
MAC PLCC Related ASCII Commands	39
ACC-68M MACRO STATION SERIAL COMMANDS	40
Serial Commands	40
<i>\$\$\$ Station Reset</i>	40
<i>\$\$\$*** Station Re-initialize</i>	40
<i>CHN Reports Channel Number</i>	40
<i>CID Reports Card ID Number</i>	40
<i>CLRF Clears Station Faults</i>	40
<i>DATE Reports Firmware Date</i>	40
<i>DISABLE PLCC or CNTRL D Disables PLCC</i>	40
<i>ENABLE PLCC Enables PLCC</i>	40
<i>MI{constant} Reports Station MI-Variable Value</i>	40
<i>MI{constant}={constant} Sets Station MI-Variable Value</i>	40
<i>MM{constant} Report Station MM-Variable Value</i>	41
<i>MM{constant}={constant} Sets Station MM-Variable Value</i>	41
<i>MP{constant} Reports Station MP-Variable Value</i>	41
<i>MP{constant}={constant} Sets Station MP-Variable Value</i>	41
<i>MM{constant}-> Reports Station MM-Variable Definition</i>	41
<i>MM{constant}-> {X/Y:offset,width,format} Sets Station MM-Variable Definition</i>	41
<i>R{address} Reads Station Address</i>	41
<i>SAVE Saves Station MI-Variables</i>	41
<i>SID Reports Serial Identification Number</i>	41
<i>VERS Reports Firmware Version</i>	41
<i>VID Reports Vendor ID Number</i>	42
<i>W{address},{value} Writes Value to Station Address</i>	42
PMAC TYPE 1 ACC-68M MACRO STATION COMMANDS	44
On-Line Commands	44
<i>MS Command</i>	44
<i>MS Variable Read</i>	44
<i>MS Variable Write</i>	45
<i>MS Variable Read Copy</i>	45
<i>MS Variable Write Copy</i>	46
Turbo PMAC PLC Commands for Type 1 Acc-68M MACRO Stations	46
<i>MS Variable Read Copy</i>	46
<i>MS Variable Write Copy</i>	47
ACC-68M MACRO STATION MEMORY AND I/O MAP	48
Global Servo Calculation Registers.....	48
Open Memory	48
DSPGATE2 Registers	48
DSPGATE2 Channel 1* and Channel 2*.....	53

INTRODUCTION

The Acc-68M is a boxed accessory with 24 isolated self-protected digital inputs and 24 isolated self-protected digital outputs. The inputs and outputs are controlled through a MACRO link either with fiber optic or copper RJ-45 connector. The inputs are either sinking or sourcing (user configurable, by wiring) at 12 to 24 Volt levels. The outputs are sinking, each at up to 24VDC with 600mA continuous and 1.2A peak for up to two seconds. Optional sets of 2 analog inputs, 2 analog outputs and 2 relay contacts are available (i.e. control one or two inverter drives through the MACRO link). This accessory works only with Turbo PMAC2 system, either in Ultralite or in UMAC MACRO versions.

Options

- **OPT-A (30A-603747-OPT): Fiber Optic MACRO connectors**
- **OPT-C (30C-603747-OPT): RJ45 MACRO connectors**
- **OPT-1 (301-603747-OPT): This option includes:**
 - Two relay contact outputs
 - Two 12-bit bipolar DAC outputs (± 10 Volts)
 - Two 16-bit bipolar ADC inputs (± 32767 Counts)

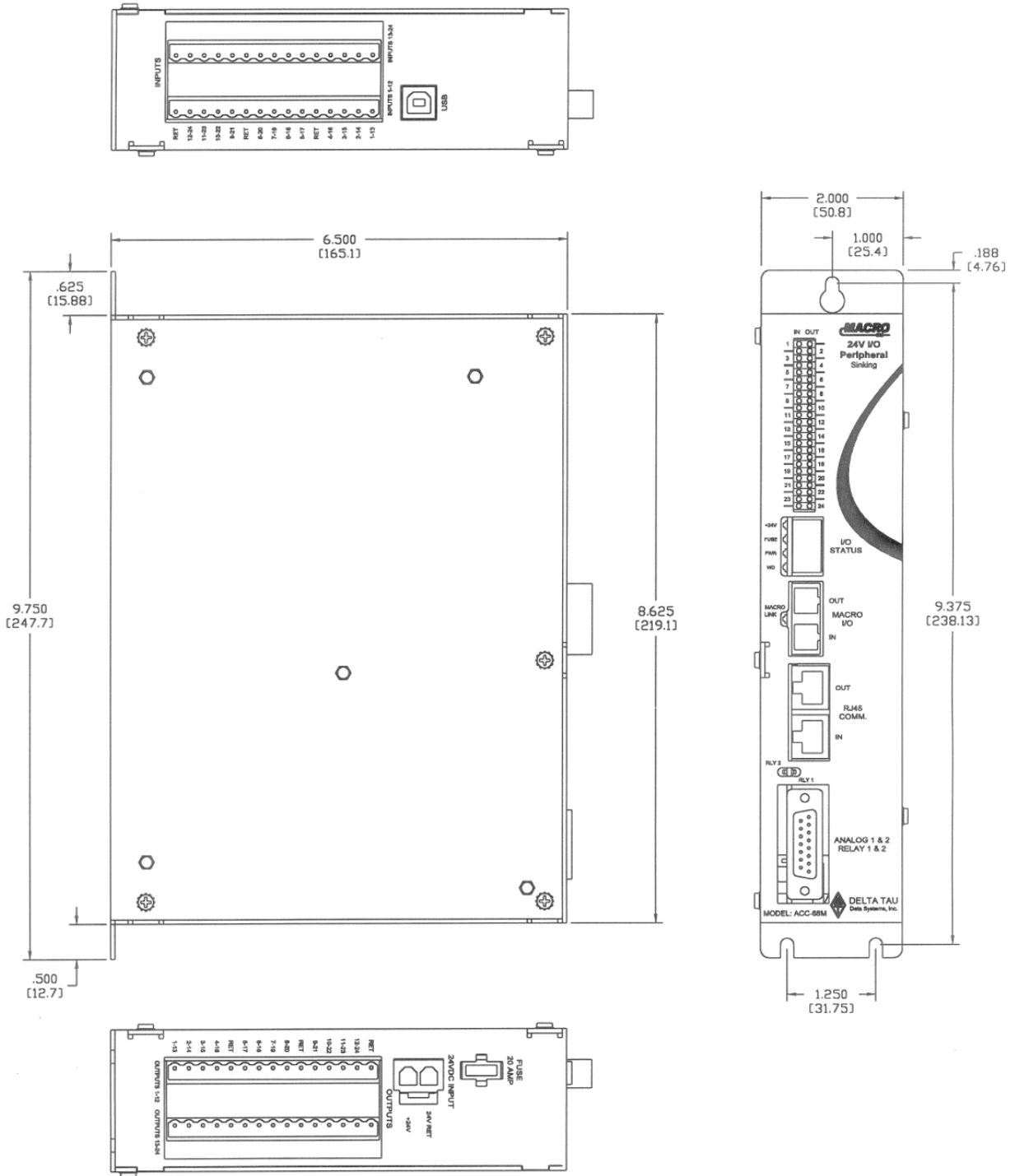


Note

Older revisions (rev 100, 101, and 102) could only support the 12-bit ADC inputs which allowed the user to have ± 2047 counts of resolution. All examples in this manual refer to the 16-bit ADC setup.

HARDWARE REFERENCE SUMMARY

Product Layout



Connectors and Indicators

Inputs and Outputs LED Indicators

Each of the 24 input and 24 output lines has an associated LED on the front panel of the unit that displays its current state; either active or inactive.

Status LED Indicators

+24V: when lit, this LED indicates that 24V is applied to the unit

Fuse: when lit, this LED indicates that the internal fuse protecting the external 24V is properly functional

PWR: when lit, this LED indicates that proper power is applied to the logic circuits

WD: when lit, this LED indicates that the watchdog safety circuit is activated indicating a failure condition

7-segment LED Indicator

This indicator reports the status of the unit with respect to the MACRO link:

- 0: Ring Active with no errors
- 1-9: NA
- A: 24V Input fault
- B: Ring-break fault
- C: Configuration change fault
- D: Ring data-error fault
- E: NA
- F: Momentary ring fault

Relay Status LED Indicators

RLY1: when lit, this LED indicates that the first amplifier enable relay is activated

RLY2: when lit, this LED indicates that the second amplifier enable relay is activated

USB Connector

This connector is used to perform some software diagnostic procedures, or to download the operational firmware. This connector is used in conjunction with the PEWIN32PRO2 or equivalent software package.

24V Input Connector

the power is applied to the unit through this connector. The power requirements are 25A @ 24VDC.

MACRO Link Connectors

The ACC-68M could be ordered with either RJ45 (twisted pair copper wires) or fiber optic connectors. In either case, one input and one output connectors are present to insert the unit in the MACRO ring. The input connector is tied to the MACRO output connector of the previous device on the ring. The output connector connects to the input MACRO connector of the next device on the ring.

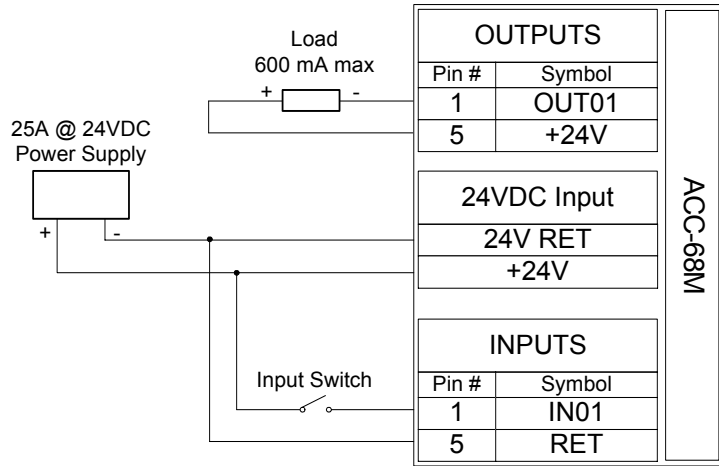
DB-15 Option-1 Connector

When OPT-1 is ordered, this connector provides the lines for two relay contact outputs, two 12-bit $\pm 10V$ DAC outputs and two bipolar 16-bit ADC inputs (producing ± 32767 Counts).

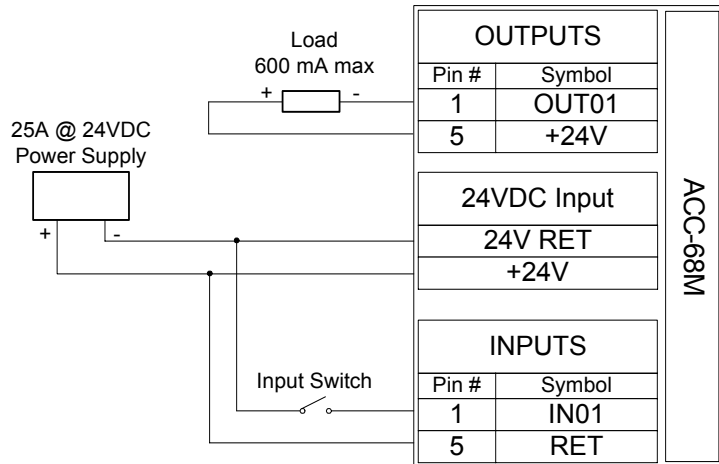
Inputs and Outputs Terminal Blocks

The digital inputs are connected through a 30-pin terminal block on the top panel of the unit, and the digital outputs are connected through a 30-pin terminal block on the bottom panel of the unit

Connections Example: Sourcing Inputs and Sinking Outputs

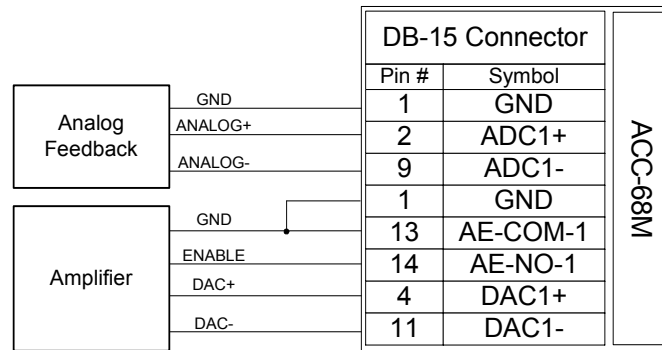


Connections Example: Sinking Inputs and Sinking Outputs



Connections Example: OPT-1 Amplifier Signals

In this example, the amplifier is enabled with a ground connection. If single-ended DAC or ADC signals are used, leave the negative side of the differential lines floating.



JUMPERS DESCRIPTION

E1: Watchdog Timer Disable

Jumper Type	Description	Default
2-Pin	Remove jumper to enable Watchdog Timer. Jump pins 1 and 2 to disable Watchdog Timer (for test purposes only).	Not jumpered

E2: CPU Mode Operation

Jumper Type	Description	Default
3-Pin	Jump pins 1 and 2 for firmware download through USB port. Jump pins 2 and 3 for normal operation.	Pin 2-3

E3: USB Port Serial Baud Rate

Jumper Type	Description	Default
2-Pin	Jump pins 1 and 2 for 9600-baud serial port operation. Remove jumper for 38400-baud serial port operation.	Not jumpered

E4: MACRO Type Connection

Jumper Type	Description	Default
2-Pin	Jump pins 1 and 2 for RJ-45 connection. Jump pins 2 and 3 for fiber optic connection.	Factory set

JP1: T

Jumper Type	Description	Default
2-Pin	Remove jumper for Acc-68M. Install jumper for Acc-68M.	Factory set

JP2 – JP6: Reserved for Future Use

JP7: Re-initialization

Jumper Type	Description	Default
2-Pin	Jump pins 1 and 2 for re-initialization on power-up/reset Remove jumper for normal operation	Not jumpered

CONNECTORS DESCRIPTION

USB Universal Serial Bus Port

Pin #	Symbol	Function
1	VCC	N.C.
2	D-	DATA-
3	D+	DATA+
4	GND	GND
5	SHELL	SHIELD
6	SHELL	SHIELD

This connector is used only to change the operational firmware, or to perform basic software diagnostic operations. The user can use a serial port terminal window such as Microsoft® HyperTerminal to communicate MACRO Device. Set the serial port communication settings as follows:

Baud Rate: 38400 for E3 not jumpered or 9600 for E3 jumpered

Data Bits: 8

Parity: None

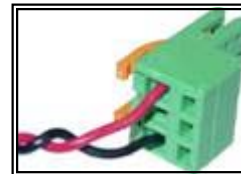
Stop Bits: 1

Flow Control: Xon/Xoff

If PEWIN PRO2 software is installed on the pc, then the USB device should be recognized by the operating system. If the device is not recognized, contact technical support for assistance.

24VDC Input Edge Connector

PIN #	Symbol	Function
1	24 V RET	GND
2	+24 V	+24 V CONTROL
3	+ 24 V	+24 V PWR



This 3-pin spring cage connector is used to power the unit with a 24VDC power supply at 25A.

Note: Delta Tau Part No: 014-188305-001

Phoenix Contact Part No: 1883051

MACRO RJ-45 Copper Connectors

				Front View
Pin #	Symbol	Function	Description	
1	DATA+	Data +	Differential MACRO Signal.	
2	DATA-	Data -	Differential MACRO Signal	
3	Unused		Unused terminated pin	
4	Unused		Unused terminated pin	
5	Unused		Unused terminated pin	
6	Unused		Unused terminated pin	
7	Unused		Unused terminated pin	
8	Unused		Unused terminated pin	

The cable used for MACRO wired connections is CAT5 verified straight-through 8 conductor. The input connector is tied to the MACRO output connector of the previous device on the link. The output connector connects to the input MACRO connector of the next device on the link.

Inputs Connector: 30-pin Terminal Block

Pin #	Symbol	Function
1	IN01	INPUT 1
2	IN02	INPUT 2
3	IN03	INPUT 3
4	IN04	INPUT 4
5	RET	RETURN FOR INPUTS 1-8
6	IN05	INPUT 5
7	IN06	INPUT 6
8	IN07	INPUT 7
9	IN08	INPUT 8
10	RET	RETURN FOR INPUTS 1-8
11	IN09	INPUT 9
12	IN10	INPUT 10
13	IN11	INPUT 11
14	IN12	INPUT 12
15	RET	RETURN FOR INPUTS 9-16
16	IN13	INPUT 13
17	IN14	INPUT 14
18	IN15	INPUT 15
19	IN16	INPUT 16
20	RET	RETURN FOR INPUTS 9-16
21	IN17	INPUT 17
22	IN18	INPUT 18
23	IN19	INPUT 19
24	IN20	INPUT 20
25	RET	RETURN FOR INPUTS 17-24
26	IN21	INPUT 21
27	IN22	INPUT 22
28	IN23	INPUT 23
29	IN24	INPUT 24
30	RET	RETURN FOR INPUTS 17-24

The inputs are sinking or sourcing by user wiring. For sinking inputs, connect the +24V side of the power supply to the corresponding return line. For sourcing inputs, connect the GND side of the power supply to the corresponding return line. See the connections example diagram in this manual for details.

Outputs Connector: 30-pin Terminal Block

Pin #	Symbol	Function
1	OUT01	OUTPUT 1
2	OUT02	OUTPUT 2
3	OUT03	OUTPUT 3
4	OUT04	OUTPUT 4
5	+24V	OUTPUTS RETURN
6	OUT05	OUTPUT 5
7	OUT06	OUTPUT 6
8	OUT07	OUTPUT 7
9	OUT08	OUTPUT 8
10	+24V	OUTPUTS RETURN
11	OUT09	OUTPUT 9
12	OUT10	OUTPUT 10
13	OUT11	OUTPUT 11
14	OUT12	OUTPUT 12
15	+24V	OUTPUTS RETURN
16	OUT13	OUTPUT 13
17	OUT14	OUTPUT 14
18	OUT15	OUTPUT 15
19	OUT16	OUTPUT 16
20	+24V	OUTPUTS RETURN
21	OUT17	OUTPUT 17
22	OUT18	OUTPUT 18
23	OUT19	OUTPUT 19
24	OUT20	OUTPUT 20
25	+24V	OUTPUTS RETURN
26	OUT21	OUTPUT 21
27	OUT22	OUTPUT 22
28	OUT23	OUTPUT 23
29	OUT24	OUTPUT 24
30	+24V	OUTPUTS RETURN

The outputs are always sinking type. Pins 5, 10, 15, 20, 25 and 30 are internally connected. See the connections example diagram in this manual for details.

OPT-1: DB-15 Connector

Pin #	Symbol	Function
1	GND	COMMON GROUND
2	ADC1+	ANALOG INPUT 1+
3	ADC2+	ANALOG INPUT 2+
4	DAC1+	ANALOG OUTPUT 1+
5	DAC2+	ANALOG OUTPUT 2+
6	AE-NC-1	NORMALLY CLOSE RELAY 1
7	AE-COM-2	COMMON RELAY 2
8	AE-NO-2	NORMALLY OPEN RELAY 2
9	ADC1-	ANALOG INPUT 1-
10	ADC2-	ANALOG INPUT 2-
11	DAC1-	ANALOG OUTPUT 1-
12	DAC2-	ANALOG OUTPUT 2-
13	AE-COM-1	COMMON RELAY 1
14	AE-NO-1	NORMALLY OPEN RELAY 1
15	AE-NC-2	NORMALLY CLOSE RELAY 2

When OPT-1 is ordered, this connector provides the lines for two relay contact outputs , two 12-bit bipolar DAC outputs (± 10 Volts), two 16-bit bipolar ADC inputs (± 32767 Counts).

SOFTWARE SETUP

Typically, the MACRO Slave Device and MACRO Master IC (Ultralite) can support to eight axis nodes (0, 1, 4, 5, 8, 9, 12, and 13) and up to six I/O transfer nodes (2, 3, 6, 7, 10, and 11). This data exchange goes through a MACRO IC at both points (master and slave) on the MACRO Ring. There are three types of I/O transfers allowed to send the information between the Turbo Ultralite and a MACRO Device: 48-bit I/O background data transfer, 72-bit phase rate I/O node transfer, and 48-bit ASCII transfer. The 48-bit I/O transfer occurs on node 15 and the 48-bit ASCII transfer occurs on node 14 using the broadcast feature of MACRO. The ACC-68M station uses the three data type transfers. The 72-bit node transfer is used to exchange all hardware I/O on the card, the 48-bit I/O transfer for MI variables and the 48-bit ASCII for Ring Order setup of the Station. (The ACC-68M does not have switches that bind it to a certain Master and Node so it uses Ring Order for initial binding to a Master and Node.) The Turbo PMAC2 Ultralite and the MACRO Station enable transfer of 72 bits per I/O node with the I6841 and MI996 type variables.

The ACC-68M is controlled through a single I/O node from a Turbo PMAC2 Ultralite board or a UMAC Turbo System. Once the node number has been established, it can be used as described in the following table:

Node Breakdown	Read	Write	Notes
24-bit Word 0	Inputs	Outputs	
16-bit Word 1	ADC1	DAC1	DACs are from bits 8-16 signed, ADCs are from bits 8-23 signed
16-bit Word 2	ADC2	DAC2	DACs are from bits 8-16 signed, ADCs are from bits 8-23 signed
16-bit Word 3	-	AENA1 & 2	AENA1 is at bit 19 and AENA2 is at bit 20

To use this card, establish communications using the ring order method to activate the nodes at the MACRO Slave Device (ACC-68M). Also activate the nodes at the MACRO Master Device (typically an Ultralite) to allow communications from the Master to the Slave. Once communications is working properly, then set up the inputs and outputs, ADCs (if ordered), DACs (if ordered), and amplifier enable outputs (if ordered).

The following sections will show the setup:

- Ring Order Method of Communications
- Inputs and Outputs on ACC-68M
- ADC's on ACC-68M
- DAC's, on ACC-68M
- Amplifier Enables on ACC-68M

ASCII Ring Order Initial Binding of the ACC-68M Station

To initially bind the ACC-68M to a MACRO Master, the Ring Order method is used. A command is sent out on the Ring by the Ring Controller in the ASCII communication protocol asking to talk to the first MACRO Device that does not have a Station Number (MI11=0 or STN=0). When this communication state is entered, the Ring Controller is now talking to the MACRO Device in an ASCII data exchange mode. That MACRO Device can be either another MACRO CPU, MACRO Amplifier or Slave Station like the ACC-68M. Once communication is established, the developer at the Ring Controller binds the Station to a Master and Node (It sets the Slave Station's MI996). It is now setup for the normal 72-bit and 48-bit I/O exchange between the Master and Slave Station (the ACC-68M). To come back and communicate with this Station in the ASCII data exchange, its station number (STN) is set normally to its order on the Ring. Once this is done, the Ring Order attempts to find the next station on the Ring that has not been setup for Ring Order (STN=0). Control T is entered. It terminates the ASCII communication transfer between the Ring Controller and the Station and returns to normal communication with the Ring Controller.

At a minimum, the following I-variables must be set to enable MACRO ASCII mode communications.

I6840=\$4030 ;to enable MACRO IC0 as sync-master and node 14 for auxiliary communications

I6841=\$0FCxxx ;to enable node 15 and 14. If activating nodes 0,1,4,5 we would set I6841=\$0FC033
 I79=32 ;Timeout value for Node 14 Auxiliary communications

If using more than one MACRO IC set up I6890, I6891, I6940, I6941, I6990, and I6991 appropriately. Once the communication variables are modified, they must be saved to the memory of the controller with the **SAVE** command and then reset the controller with either a **\$\$\$** command or power cycle the controller.

Note:

The PMAC Controller will be able to communicate to the MACRO Device in MACRO ASCII communication mode after the unit has been restarted with the changes saved to its memory.

Establishing Communications with the ACC-68M Station

After hooking up the Ring and 24VDC power, try to read and write to the IO Device.

1. Ring Order (at the Ring Controller), enter **MACSTA255**. Now a Station number can be assigned by entering **STN=n** where **n** is the Station number. If a Macro I/O error is received, make sure I6840, I6841 and I79 are set correctly. Also make sure that the unit has not been assigned a Station number already.

If the station has already been assigned a station number, there are two options:

- a) Find out the station number **n** and enter **MACSTA<n>**, where **n** is the station number, to initiate MACRO ASCII communication with the Station.
- b) Reset the station number of all the stations by entering **MACSTA0** and then enter **STN=0**

Note:

This will not reset all the parameters in the MACRO Stations or **\$\$\$****

Note:

This will reset all the parameters in the MACRO Stations.

Next enter **^T** to exit MACRO ASCII communications. Then enter **MACSTA255** to access the first Station. Now assign it a station number by entering **STN=n** where **n** is the station number. Enter **^T** to exit MACRO ASCII Communications. Enter **MACSTA255** again to access the next station and repeat this process until a MACRO I/O error is received stating that there are no further unassigned stations.

2. Enter **MACSTA<n>** where **n** is the Station number. Enter **I996=\$F4004**. (Binds to Ring Controller 0 and Node 2)
3. Enter **^T**. (Control-T terminates MACRO ASCII Communications.)
4. Enter **MSCLRF2**. (Clears any faults)
5. Enter **I6841=I6841 | \$0FC004** (Enable Node 2).

6. Set up M-Variables for I/O as follows:

```
M0->X:$78420,24 ; 24 bit I/O
M1->X:$78421,8,16,S ; DAC_1 Output
M2->X:$78422,8,16,S ; DAC_2 Output
M3->X:$78423,0,24 ; AENA_1/2 Output command
M4->X:$78421,8,16,S ; ADC_1 Input
M5->X:$78422,8,16,S ; ADC_2 Input
```

7. Test with the I/O (if I/O is powered properly and not connected to machine devices)

```
M0=$555555 ;The Output LEDs in the 55 pattern.
M0=$AAAAAA ;The Output LEDs in the AA pattern.
M3=$80000 ;The AENA_1 LED on and the relay closed.
M3=$100000 ;The AENA_2 LED on and the relay closed
M1=653 ;DAC_1 output at 1V
M2=653 ;DAC_2 output at 1V
M4 & M5 ;ADC_1/2 inputs
```

Using ACC-68M Inputs and Outputs

The MACRO Peripheral Devices such as the ACC-65M and the ACC-68M allow reading and writing to 24 inputs and 24 outputs. These MACRO Peripheral Devices use the 24-bit node register of the activated node. Using the IO is accomplished by writing to a node register to activate the desired outputs and reading the same node register to read the status of the inputs. In other words the one 24-bit node register is used for both inputs and outputs. This is efficient because it allows the 48-bits of information to be processed using one 24-bit word and minimizes the number of nodes needed for the IO data transfers for each MACRO Device. The only drawback to this technique is that the user will have to keep track of the status of their outputs (see example).

Example: If node 2 is activated at both the Master and MACRO Device, make the following definitions to read and write to the inputs and outputs.

```

M3000->X:$78420,0,24           ;Actual Input/Output Word for node2
M4000->X:$10F0,0,24           ;Input Image Word
M4001->X:$10F1,0,24           ;Output Image word

Open PLC1 Clear
M4000=M3000                     ;Input Image Word equals Actual Input Word
}
} Process Inputs and build image
} output word (M4001)
M3000=M4001                     ;Set Actual Output word to Output Image word
Close

```

If the user is using another node they can be accessed at the following locations:

User Node	IO Word Address
2	X:\$078420,0,24
3	X:\$078424,0,24
6	X:\$078428,0,24
7	X:\$07842C,0,24
10	X:\$078430,0,24
11	X:\$078434,0,24
18	X:\$079420,0,24
19	X:\$079424,0,24
22	X:\$079428,0,24
23	X:\$07942C,0,24
26	X:\$079430,0,24
27	X:\$079434,0,24
34	X:\$078420,0,24
35	X:\$07A424,0,24
38	X:\$07A428,0,24
39	X:\$07A42C,0,24
42	X:\$07A430,0,24
43	X:\$07A434,0,24
50	X:\$07B420,0,24
51	X:\$07B424,0,24
54	X:\$07B428,0,24
55	X:\$07B42C,0,24
58	X:\$07B430,0,24
59	X:\$07B434,0,24

Using the ACC-68M ADC

The MACRO Peripheral Accessories can be ordered with two analog to digital converters. These A/D converters are 16-bit devices that are ready to be used without any software setup. Delta Tau uses the Burr Brown ADS8361E for this circuit. This 16-bit option can be ordered on the ACC-68M (603747-103) revision 2 and greater.

To read the A/D data from the MACRO device, the user must create the M-variable definitions to the node associated with the MACRO device. The data received is a signed 16-bit number scaled from -10V to +10V. The data is transferred into the upper 16-bits of the MACRO IO node registers. For example, if the ACC-68M is associated with node 2, then the following M-variable assignment can be created:

```
M5000->X:$78421,8,16,S      ;ADC0 upper 16 bits of IO Node 2 word1
M5001->X:$78422,8,16,S      ;ADC1 upper 16 bits of IO Node 2 word2
```

Example Data Read: If you read a value of 20480 in M5000, then that would be approximately 6.25V. 16-bit Voltage Conversion = $\text{Data} \times 10\text{V} / 32767$ or for this example, $6.25\text{V} \approx 20480 \times 10\text{V} / 32767$.

Earlier revisions of the ACC-68M used the 12-bit Burr Brown ADS7861E for this circuit. For proper operation of the 12-bit ADC or if the users application requires 12-bit data reads then they can make the following definitions (12-bit Voltage Conversion = $\text{Data} \times 10\text{V} / 2047$):

```
M5000->X:$78421,12,12,S     ;ADC0 upper 12 bits of IO Node 2 word1
M5001->X:$78422,12,12,S     ;ADC1 upper 12 bits of IO Node 2 word2
```

These examples also assume that the IO nodes are activated at both the MACRO Peripheral Device (Slave) and at the Ultralite (Master). The following table lists the locations of the ADCs if using other node locations.

User Node	ADC0	ADC1
2	X:\$078421,8,16,S	X:\$078422,8,16,S
3	X:\$078425,8,16,S	X:\$078426,8,16,S
6	X:\$078429,8,16,S	X:\$07842A,8,16,S
7	X:\$07842D,8,16,S	X:\$07842E,8,16,S
10	X:\$078431,8,16,S	X:\$078432,8,16,S
11	X:\$078435,8,16,S	X:\$078436,8,16,S
18	X:\$079421,8,16,S	X:\$079422,8,16,S
19	X:\$079425,8,16,S	X:\$079426,8,16,S
22	X:\$079429,8,16,S	X:\$07942A,8,16,S
23	X:\$07942D,8,16,S	X:\$07942E,8,16,S
26	X:\$079431,8,16,S	X:\$079432,8,16,S
27	X:\$079435,8,16,S	X:\$079436,8,16,S
34	X:\$07A421,8,16,S	X:\$07A422,8,16,S
35	X:\$07A425,8,16,S	X:\$07A426,8,16,S
38	X:\$07A429,8,16,S	X:\$07A42A,8,16,S
39	X:\$07A42D,8,16,S	X:\$07A42E,8,16,S
42	X:\$07A431,8,16,S	X:\$07A432,8,16,S
43	X:\$07A435,8,16,S	X:\$07A436,8,16,S
50	X:\$07B421,8,16,S	X:\$07B422,8,16,S
51	X:\$07B425,8,16,S	X:\$07B426,8,16,S
54	X:\$07B429,8,16,S	X:\$07B42A,8,16,S
55	X:\$07B42D,8,16,S	X:\$07B42E,8,16,S
58	X:\$07B431,8,16,S	X:\$07B432,8,16,S
59	X:\$07B435,8,16,S	X:\$07B436,8,16,S

Using the ACC-68M DAC Output

The MACRO Peripheral Accessories can be ordered with two +/-10V outputs produced by filtering a PWM signal. This technique has been used for some time now by many of our competitors. Although this technique does not contain the same levels of performance as a true Digital to Analog converter, for most servo applications it is more than adequate.

Both the resolution and the frequency of the Filtered PWM outputs are configured in software on the MACRO Peripheral Accessories through the variable MI992 (MaxPhase Setting). This MI992 variable also effects the phase and servo interrupts. On most MACRO systems, the MI992 value will be set to the same value as the Turbo Ultralite or UMAC ACC-5E equivalent MaxPhase I-variable (I6800/I6850/I6900/I6950). Another key variable to be concerned about at the MACRO Peripheral Device is MI994. MI994 is the PWM dead time specification and it is used to add dead time to the PWM signal as a safety feature for PMAC2 direct PWM commutation.

For the DAC outputs for the MACRO peripheral this variable (MI994) should be set to zero. For more details about these variables, refer to the Software Reference manuals for the respective hardware devices.

To write to the DAC devices at MACRO Peripheral Device, create the M-variable definitions to the node associated with the MACRO device. The data received is a signed 16-bit number scaled from -10V to +10V. The data is transferred into the upper-bits of the MACRO IO node registers. For example, if the ACC-68M is associated with node 2, then the following M-variable assignment can be made:

```
M5000->X:$78421,8,16,S           ;DAC0 bits of IO Node2 word1
M5001->X:$78422,8,16,S           ;DAC1 bits of IO Node 2 word2
MS2,MI992=6527                   ;6527 is default
MS2,MI994=0                       ;set dead-time to zero
```

This example also assumes that the IO node number 2 is activated at both the MACRO Peripheral Device (Slave) and at the Ultralite (Master).

To scale the outputs, the relationship between MI992 and the DAC outputs must be known. If MI994=0, then assume that the maximum voltage output will be scaled relative to MI992. For example, if MI992=6527 (default value), and if M5000=6527 then, measure 10V on DAC1+ relative to AGND or 20V relative to DAC1-. Likewise, if M5000=652.7, measure +1V on DAC1+ relative to AGND or +2V relative to DAC1-.

The following table lists the locations of the DACs if using other node locations.

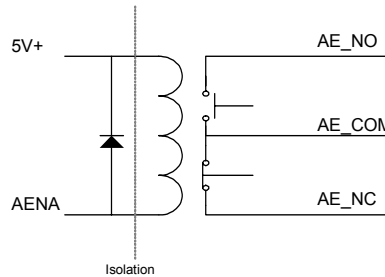
User Node	DAC0	DAC1
2	X:\$078421,8,16,S	X:\$078422,8,16,S
3	X:\$078425,8,16,S	X:\$078426,8,16,S
6	X:\$078429,8,16,S	X:\$07842A,8,16,S
7	X:\$07842D,8,16,S	X:\$07842E,8,16,S
10	X:\$078431,8,16,S	X:\$078432,8,16,S
11	X:\$078435,8,16,S	X:\$078436,8,16,S
18	X:\$079421,8,16,S	X:\$079422,8,16,S
19	X:\$079425,8,16,S	X:\$079426,8,16,S
22	X:\$079429,8,16,S	X:\$07942A,8,16,S
23	X:\$07942D,8,16,S	X:\$07942E,8,16,S
26	X:\$079431,8,16,S	X:\$079432,8,16,S
27	X:\$079435,8,16,S	X:\$079436,8,16,S
34	X:\$07A421,8,16,S	X:\$07A422,8,16,S
35	X:\$07A425,8,16,S	X:\$07A426,8,16,S
38	X:\$07A429,8,16,S	X:\$07A42A,8,16,S
39	X:\$07A42D,8,16,S	X:\$07A42E,8,16,S
42	X:\$07A431,8,16,S	X:\$07A432,8,16,S
43	X:\$07A435,8,16,S	X:\$07A436,8,16,S
50	X:\$07B421,8,16,S	X:\$07B422,8,16,S

51	X:\$07B425,8,16,S	X:\$07B426,8,16,S
54	X:\$07B429,8,16,S	X:\$07B42A,8,16,S
55	X:\$07B42D,8,16,S	X:\$07B42E,8,16,S
58	X:\$07B431,8,16,S	X:\$07B432,8,16,S
59	X:\$07B435,8,16,S	X:\$07B436,8,16,S

Using the ACC-68M Amplifier Enable Outputs

The MACRO Peripheral amplifier enable outputs are very similar to the amplifier enable circuits used on other Delta Tau products. These outputs allow the user to send outputs that can be configured as either normally open or normally closed. The amplifier enable outputs are memory mapped to the third 16-bit node address or at the base address of the activated node +3. The AENA1 signal is at bit 19 and AENA2 signal is at bit 20. The following M-variable definitions can be made for the AENA signals at node 2 (assuming node 2 is activated at both the Master and MACRO Device).

```
M3003->X:$78423,8,16 ;AENA Outputs at bits 19 and 20
M3010->X:$78423,19 ;AENA1
M3011->X:$78423,20 ;AENA2
```



The following table lists the AENA output locations for all nodes.

User Node	AENA1	AENA2
2	X:\$078423,19	X:\$078423,20
3	X:\$078427,19	X:\$078427,20
6	X:\$07842B,19	X:\$07842B,20
7	X:\$07842F,19	X:\$07842F,20
10	X:\$078433,19	X:\$078433,20
11	X:\$078437,19	X:\$078437,20
18	X:\$079423,19	X:\$079423,20
19	X:\$079427,19	X:\$079427,20
22	X:\$07942B,19	X:\$07942B,20
23	X:\$07942F,19	X:\$07942F,20
26	X:\$079433,19	X:\$079433,20
27	X:\$079437,19	X:\$079437,20
34	X:\$07A423,19	X:\$07A423,20
35	X:\$07A427,19	X:\$07A427,20
38	X:\$07A42B,19	X:\$07A42B,20
39	X:\$07A42F,19	X:\$07A42F,20
42	X:\$07A433,19	X:\$07A433,20
43	X:\$07A437,19	X:\$07A437,20
50	X:\$07B423,19	X:\$07B423,20
51	X:\$07B427,19	X:\$07B427,20
54	X:\$07B42B,19	X:\$07B42B,20
55	X:\$07B42F,19	X:\$07B42F,20
58	X:\$07B433,19	X:\$07B433,20
59	X:\$07B437,19	X:\$07B437,20

FLAGS Command Register

- B09 - Reserved for future ring protocol control
- B10 - Reserved for future ring protocol control
- B11 - Reserved
- B12 - Reserved
- B13 - The Slave detected a MACRO Ring Break MRB & became a Ring master (Note 3)
- B14 - Reserved
- B15 - When B13 = 1 then B15 = 1 is a Station Fault.
- B16 - Reserved for future ring protocol control
- B17 - Reserved for future ring protocol control
- B18 - Reserved for future ring protocol control
- B19 - Fast User Defined Command Flag (UserCmd1) **AENA_1 Output**
- B20 - Fast User Defined Command Flag (UserCmd2) **AENA_2 Output**
- B21 - Fast User Defined Command Flag (UserCmd3)
- B22 - Fast User Defined Command Flag (UserCmd4)
- B23 - Fast User Defined Command Flag (UserCmd5)

FLAGS Status Register

- B08 - Reserved for future ring protocol status
- B09 - Reserved for future ring protocol status
- B10 - Reserved for future ring protocol status
- B11 - Reserved
- B12 - Reserved
- B13 - This Node detected a MACRO Ring Break MRB (Note 3)
- B14 - Reserved
- B15 - Station or Real Time Data Node Fault
- B16 - Reserved for future ring protocol status
- B17 - Reserved for future ring protocol status
- B18 - Reserved for future ring protocol status
- B19 - Fast User Defined Status Flag (UserSatus1) **AENA_1 Output**
- B20 - Fast User Defined Status Flag (UserSatus2) **AENA_2 Output**
- B21 - Fast User Defined Status Flag (UserSatus3)
- B22 - Fast User Defined Status Flag (UserSatus4)
- B23 - Fast User Status Flag (UserSatus5)

MACRO ASCII Communication Reference:

1. **VID** Vendor ID (Delta Tau = 1, Range=1- 65535)
2. **CID** Vendor Card ID, Part Number, (Range=1- 4,294,967,295) 32 bit unsigned.
 - a) Delta Tau: Turbo PMAC 2 VME = 602413 (MACRO Master)
 - b) Delta Tau: Turbo PMAC 2 Ultralite = 603182 (MACRO Master)
 - c) Delta Tau: Turbo UMAC = 603382 (MACRO Master)
 - d) Delta Tau: UMAC MACRO 8 = 602804 (MACRO Slave)
 - e) Delta Tau: UMAC MACRO 16 = 602???
3. **SID** Serial ID (Range = 64 bit unsigned, 0=Serial ID not available)
4. **\$\$\$**** - Station to reset to default parameter with no station number and ready for Ring location identification. **Note not \$\$\$***.**
5. **SAVe** – Save station number and initialization parameters.
6. **\$\$\$** – Reset Station to saved station number and initialization parameters.
7. **STN=n <n=0-254>** – Assigns the MACRO station number. Normally, this would be its order in the Ring. A STN=0 resets the station number and is reserved for the Ring Controller Master.

8. Commands with **STN=0** is a broadcast to all stations in Ring.
9. Commands with **STN=255** is a request for communication with the 1st station in Ring with its STN=0.
10. Commands with **STN=1-254** is a request for communication with the station in Ring with STN=1-254.
11. **STN** – The addressed MACRO Station responds with its station number (n).
12. **STN=n** where n=0-254 – Set the addressed MACRO Station's STN to n.

Note:

The station will stop responding.

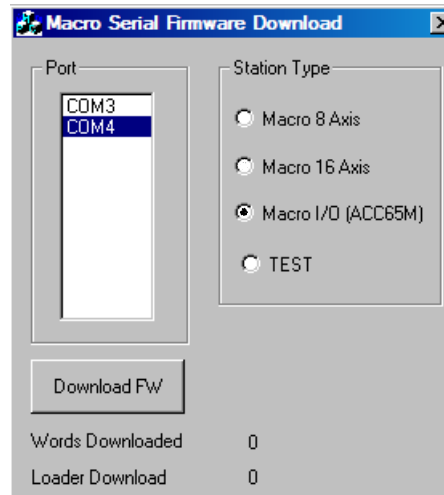
Firmware Updates

Downloading new firmware to the MACRO IO Device is a simple process once the MACRO board is set up properly. To download new firmware to the MACRO IO Device, obtain the following items:

- Two jumpers
- USB cable
- Install drivers for MACRO IO USB Device from <http://www.ftdichip.com>. Usually PEWIN Pro will install the driver and the board should be seen by the operating system automatically.
- MACRO Firmware Download Software (MacroFWDown.exe)
- New firmware file (MACROIO.bin)

To download the software to the MACRO Device:

1. Copy the firmware into a directory (C:\Macro\Firmware).
2. Jumper the E2 (1-2) and E1 (1-2).
3. Place the USB cable to the J2 USB connection on the MACRO Device and place the other end to the USB port on the PC.
4. Power up the MACRO Device and then launch MacroFWDown.exe. Choose the Com port and select **MACRO I/O** and then click the **Download FW** button.



After the download is complete, power down the system and remove jumper E1 and place jumper E2 from 2 to 3.

ACC-68M MACRO STATION MI-VARIABLE REFERENCE

The Acc-68M is set up through its own set of initialization I-variables, which are distinct from the I-variables on a Turbo PMAC2. Usually, they are referenced as MI-variables (e.g. MI900) to distinguish them from PMAC's own I-variables, although they can be referenced just as I-variables.

These MI-variables can be accessed through the on-line **MS{node#},MI{variable#}** read and **MS{node#},MI{variable#}={constant}** write commands, or the **MSR{node#},MI{variable#},{PMAC variable}** read-copy and **MSW{node#},MI{variable#},{PMAC variable}** write-copy commands (either on-line or background PLC), where **{node#}** specifies the MACRO node number (0 to 13), **{variable#}** specifies the number of the Station MI-variable (0 - 1023), **{constant}** represents the numerical value to be written to the Station MI-variable, or **{PMAC variable}** specifies the value to be copied to or from the Station MI-variable.

For most Station MI-variables, the **{node#}** specifier can take the number of any active node on the station (usually the lowest-numbered active node). These variables have **MS{anynode}** in the header of their descriptions below.

Global MI-Variables

MS{anynode},MI0 Station Firmware Version (Read Only)

Range: 1.200 - 1.999
Units: Revision numbers

This variable, when queried, reports the version of the firmware on the Acc-68M MACRO Station.

Example:

```
MS0,MI0
1.200
```

MS{anynode},MI1 Station Firmware Date (Read Only)

Range: 01/01/00 – 12/31/99
Units: MM/DD/YY

This variable, when queried, reports the date of implementation of the firmware on the Acc-68M MACRO Station. The date is reported in the North American style of month/day/year with two decimal digits for each.

Since the year is reported with only two digits, it rolls over at the turn of a century. If the software makes any date comparisons based on this year value, care must be taken to avoid a Y2K error. The earliest firmware date for the ACC-68M MACRO Station is in year 2003. The PMAC command MSDATE, which polls this value, turns the year into a 4-digit value before reporting the value to the host computer.

MS{anynode},MI2 Station ID and User Configuration Word

Range: \$000000 - \$FFFFFF
Units: none
Default: 0

With this variable, a station identification number can be written to the Acc-68M MACRO Station. Typically, when the software setup of a station is complete, a unique value is written to this MI-variable in the station and saved with the other MI-variables. On power-up/reset, the controller can query MI2 as a quick test to see if the station has been set up properly for the application. If it does not report back the expected value, the controller can download and save the setup values.

MS{anynode},MI3 (Reserved for Future Use)

Range: \$00
 Units: none
 Default: 0

MS{anynode},MI4 Station Status Word (Read Only)

Range: \$000000 - \$FFFFFF
 Units: Bits

This variable, when queried, reports the value of the current status word bits for the ACC-68M MACRO Station. The value reported should be broken into bits. Each bit reports the presence or absence of a particular fault on the Station. If the bit is 0, the fault has not occurred since station faults were last cleared. If the bit is 1, the fault has occurred since station faults were last cleared.

BITn	Fault Description
0	CPU – Fault (No MACRO IC #1 detected)
1	Ring Error - Temporary
2	Ring Break
3	Station Fault - Station Shutdown
4	Ring Fault - Any permanent Ring fault
5	Spare
6	24 VDC Output Power Fault
7	Ring Break Received
8	EPROM Saved Variables Checksum Error
9	Spare
10	Spare
11	Spare
12	Ring Active
13	Spare
14	Spare
15	Spare
16	Spare
17	Spare
18	Spare
19	Spare
20	Spare
21	Spare
22	Spare
23	Detected CPU MACRO IC #1 (\$C080)

Any of the fault bits that are set can be cleared with the **MSCLR{ anynode }** (clear fault) command, or the **MS\$\$\$ { anynode }** (Station reset) command.

MS{anynode},MI5 Ring Error Counter

Range: \$000000 - \$FFFFFF
 Units: Error Count

This variable, when queried, reports the number of ring communications errors detected by the Acc-68M MACRO Station since the most recent power-up or reset.

Note:

A value may be written to this variable, but this should not be done if using MI6.

The ring error counter value can be cleared to zero using the **MS\$\$\$ { anynode }** command.

MS{anynode},MI6 Maximum Permitted Ring Errors in One Second

Range: \$0000000 - \$FFFFFFF
 Units: Errors per second

This variable sets the maximum number of ring errors that can be detected by the Acc-68MMACRO Station in a one-second period without causing it to shut down for ring failure.

MS{anynode},MI7 (Reserved for Future Use)

Range: 0
 Units: none
 Default: 0

MS{anynode},MI8 MACRO Ring Check Period

Range: 0 - 255
 Units: Station phase cycles
 Default: 8

MI8 determines the period, in phase cycles, for the Acc-68M MACRO Station to evaluate whether or not there has been a MACRO ring failure. Every phase cycle, the Station checks the ring communications status. In MI8 phase cycles (or MACRO ring cycles), the Station must receive at least MI10 sync packets and detect fewer than MI9 ring communications errors to conclude that the ring is operating correctly. Otherwise, it will conclude that the ring is not operating properly, set its servo command output values to zero, set its amplifier enable outputs to the disable state, and force all of its digital outputs to their shutdown state.

If MI8 is set to 0 at power-on/reset, the Acc-68MMACRO Station will set it to 8 automatically.

MS{anynode},MI9 MACRO Ring Error Shutdown Count

Range: 0 - 255
 Units: none
 Default: 4

MI9 determines the number of MACRO communications errors detected that will cause a shutdown fault of the Acc-68M MACRO Station. If the station detects MI9 or greater MACRO communications errors in MI8 phase (MACRO ring) cycles, it will shut down on a MACRO communications fault, turning off all outputs.

The station can detect one ring communications error per phase cycle (even if more than one error has occurred). Setting MI9 greater than MI8 means that the Station will shut down for ring communications error.

The station can detect four types of communications errors: byte violation errors, packet checksum errors, packet overrun errors, and packet under run errors. If MI9 errors have occurred in the MI8 check period, and at least half of these errors are byte violation errors, the station will conclude that there is a ring break immediately upstream of it (if there are no ring input communications to the station, there will be continual byte violation errors). In this case, not only will it set its servo command output values to zero, set its amplifier enable outputs to the disable state, and force all of its digital outputs to their shutdown state as defined by I72-I89, also it will turn itself into a master so it can report to other devices downstream on the ring.

If MI9 is set to 0 at power-on/reset, the Acc-68MMACRO Station will set it to 4 automatically.

MS{anynode},MI10 MACRO Sync Packet Shutdown Count

Range: 0 – 65,535
Units: none
Default: 4

MI10 determines the number of MACRO ring sync packets that must be received during a check period for the station to consider the ring to be working properly. If the station detects fewer than MI10 sync packets in MI8 phase (MACRO ring) cycles, it will shut down on a MACRO communications fault, setting its servo command output values to zero, setting its amplifier enable outputs to the disable state, and forcing all of its digital outputs to their shutdown state as defined by I72-I89.

The node number (0-15) of the sync packet is determined by bits 16-19 of station variable MI996. On the Acc-68M MACRO Station, this is node 15 (\$F) because this node is always active for MACRO Type 1 auxiliary communications.

The station checks each phase cycle to see whether or not a sync packet has been received. Setting MI10 to 0 means the station will not shut down for lack of sync packets. Setting MI10 greater than MI8 means that the Station will shut down for lack of sync packets.

If MI10 is set to 0 at power-on/reset, the Acc-68M MACRO Station will set it to 4 automatically.

MS{anynode},MI11 Station Order Number

Range: 0 – 254
Units: none
Default: 0

MI11 contains the station-order number of the Acc-68M MACRO Station on the ring. This permits it to respond to auxiliary MACROSTASCII commands from a Turbo PMAC ring controller from a power-on default state.

The station ordering scheme permits the ring controller to isolate each master or slave station on the ring in sequence and communicate with it, without knowing in advance how the ring is configured or whether there are any conflicts in the regular addressing scheme. This is useful for the initial setup and debugging of the ring configuration.

Normally, station order numbers of devices on the ring are assigned in numerical order, with the station downstream of the ring controller getting station-order number 1. This does not have to be the case, however.

Unordered stations have the station-order number 0. When the ring controller executes a MACROSTASCII255 command, the first unordered station in the ring will respond.

In addition, MI11 can be set with the ASCII command **STN={constant}** and the value of MI11 can be queried with the ASCII command **STN**.

MS{anynode},MI12 Card Identification

Range: 0 – \$FFFFFF
Units: none
Default: \$9365C(603740)

This returns the card part number. This is the same as the CID ASCII command.

MS{anynode},MI13 (Reserved for Future Use)

Range: 0
Units: none
Default: 0

MS{anynode},MI14 (Reserved for Future Use)

Range: 0
 Units: none
 Default: 0

MS{anynode},MI15 Enable MACRO PLCC

Range: 0 - 1
 Units: none
 Default: 0

MI15 enables and disables the PLCCs running in the Acc-68M MACRO CPU.

MACRO IC MI-Variables

Each MACRO IC (0 & 1) has its own set of these variables; therefore, they are accessed through their MACRO IC. These MI-Variables determine the servo channel transfer addresses. Most are read-only variables and cannot be changed.

MS{anynode},MI176 MACRO IC Base Address

Range: \$000000 - \$00FFFF
 Units: Acc-68MMACRO Station Addresses
 Default: \$C080 for MACRO IC

MS{anynode},MI177 MACRO IC Address for Node 14

Range: \$000000 - \$00FFFF
 Units: Acc-68MMACRO Station Addresses
 Default: \$C0B8 for MACRO IC

MS{anynode},MI178 MACRO IC Address for Node 15

Range: \$000000 - \$00FFFF
 Units: Acc-68MMACRO Station Addresses
 Default: \$C0BC for MACRO IC

MS{anynode},MI181–MI188 MACRO Channels 1 - 8 Address

Range: \$000000000000- \$00FFFF00FFFF
 Units: Acc-68MMACRO Station Addresses
 Default:

These are 48 bit read-only MI variables. The X:part of the MI variable is the MACRO channel address of the Command/Status word. The Y:part is zero.

MACRO IC I/O Transfer MI-Variables

Each MACRO IC (0 & 1) has its own set of these variables. Therefore, they are accessed through their MACRO IC.

MS{anynode},MI198 Direct Read/Write Format and Address

Range: \$000000 - \$FFFFFF
Units: Modified Acc-68MMACRO Station Addresses
Default: \$000000

MI198 controls the address and format of the register to be accessed (read from or written to) with MI199. Any register on the ACC-68M MACRO Station can be accessed by first assigning a value to MI198, and then either reading MI199 or writing to it.

MI198 is a 24-bit variable that can be expressed as six hexadecimal digits. The low 16 bits, represented by the last four hex digits, represent the ACC-68M MACRO Station address of the register. The high eight bits, represented by the first two hex digits, represent the format of that address. The following table shows the legal entries for the first two digits and the format each represents.

For example, for the host computer to read the contents of the DAC1A register as a signed quantity – the high 16 bits of Y:\$C002 – of the Acc-68M MACRO Station through a PMAC board, MI198 would be set to \$6DC002, then MI199 would be read. For an ACC-68M MACRO Station with an active node 0, this could be done with the on-line commands:

```
MS0 , MI198=$6DC002  
MS0 , MI199  
16384
```

In another example, to read the state of Channel 2's encoder A input – bit 12 of X:\$C008 – through a PMAC board, MI198 would be set to \$8CC008, then MI199 would be read.

MI198 Format Digits

MI198 Digits	Address Space	Starting Bit	Bit Width	Format	MI198 Digits	Address Space	Starting Bit	Bit Width	Format
\$00	Y	0	2	U	\$80	X	0	1	U
\$01	Y	2	2	U	\$81	X	1	1	U
\$02	Y	4	2	U	\$82	X	2	1	U
\$03	Y	6	2	U	\$83	X	3	1	U
\$04	Y	8	2	U	\$84	X	4	1	U
\$05	Y	10	2	U	\$85	X	5	1	U
\$06	Y	12	2	U	\$86	X	6	1	U
\$07	Y	14	2	U	\$87	X	7	1	U
\$08	Y	16	2	U	\$88	X	8	1	U
\$09	Y	18	2	U	\$89	X	9	1	U
\$0A	Y	20	2	U	\$8A	X	10	1	U
\$0B	Y	22	2	U	\$8B	X	11	1	U
\$0C	-	-	-	-	\$8C	X	12	1	U
\$0D	-	-	-	-	\$8D	X	13	1	U
\$0E	-	-	-	-	\$8E	X	14	1	U
\$0F	-	-	-	-	\$8F	X	15	1	U
\$10	Y	0	1	U	\$90	X	16	1	U
\$11	Y	1	1	U	\$91	X	17	1	U
\$12	Y	2	1	U	\$92	X	18	1	U
\$13	Y	3	1	U	\$93	X	19	1	U
\$14	Y	4	1	U	\$94	X	20	1	U
\$15	Y	5	1	U	\$95	X	21	1	U
\$16	Y	6	1	U	\$96	X	22	1	U
\$17	Y	7	1	U	\$97	X	23	1	U
\$18	Y	8	1	U	\$98	X	0	4	U
\$19	Y	9	1	U	\$99	X	0	4	S
\$1A	Y	10	1	U	\$9A	-	-	-	-
\$1B	Y	11	1	U	\$9B	-	-	-	-
\$1C	Y	12	1	U	\$9C	X	4	4	U
\$1D	Y	13	1	U	\$9D	X	4	4	S
\$1E	Y	14	1	U	\$9E	-	-	-	-
\$1F	Y	15	1	U	\$9F	-	-	-	-
\$20	Y	16	1	U	\$A0	X	8	4	U
\$21	Y	17	1	U	\$A1	X	8	4	S
\$22	Y	18	1	U	\$A2	-	-	-	-
\$23	Y	19	1	U	\$A3	-	-	-	-
\$24	Y	20	1	U	\$A4	X	12	4	U
\$25	Y	21	1	U	\$A5	X	12	4	S
\$26	Y	22	1	U	\$A6	-	-	-	-
\$27	Y	23	1	U	\$A7	-	-	-	-
\$28	Y	0	4	U	\$A8	X	16	4	U
\$29	Y	0	4	S	\$A9	X	16	4	S
\$2C	Y	4	4	U	\$AC	X	20	4	U
\$2D	Y	4	4	S	\$AD	X	20	4	S
\$30	Y	8	4	U	\$B0	X	0	8	U
\$31	Y	8	4	S	\$B1	X	0	8	S
\$34	Y	12	4	U	\$B4	X	4	8	U

MI198 Format Digits (continued)

MI198 Digits	Address Space	Starting Bit	Bit Width	Format	MI198 Digits	Address Space	Starting Bit	Bit Width	Format
\$35	Y	12	4	S	\$B5	X	4	8	S
\$38	Y	16	4	U	\$B8	X	8	8	U
\$39	Y	16	4	S	\$B9	X	8	8	S
\$3C	Y	20	4	U	\$BC	X	12	8	U
\$3D	Y	20	4	S	\$BD	X	12	8	S
\$40	Y	0	8	U	\$C0	X	16	8	U
\$41	Y	0	8	S	\$C1	X	16	8	S
\$44	Y	4	8	U	\$C4	X	0	12	U
\$45	Y	4	8	S	\$C5	X	0	12	S
\$48	Y	8	8	U	\$C8	X	4	12	U
\$49	Y	8	8	S	\$C9	X	4	12	S
\$4C	Y	12	8	U	\$CC	X	8	12	U
\$4D	Y	12	8	S	\$CD	X	8	12	S
\$50	Y	16	8	U	\$D0	X	12	12	U
\$51	Y	16	8	S	\$D1	X	12	12	S
\$54	Y	0	12	U	\$D4	X	0	16	U
\$55	Y	0	12	S	\$D5	X	0	16	S
\$58	Y	4	12	U	\$D8	X	4	16	U
\$59	Y	4	12	S	\$D9	X	4	16	S
\$5C	Y	8	12	U	\$DC	X	8	16	U
\$5D	Y	8	12	S	\$DD	X	8	16	S
\$60	Y	12	12	U	\$E0	X	0	20	U
\$61	Y	12	12	S	\$E1	X	0	20	S
\$64	Y	0	16	U	\$E4	X	4	20	U
\$65	Y	0	16	S	\$E5	X	4	20	S
\$68	Y	4	16	U	\$E8	X	0	24	U
\$69	Y	4	16	S	\$E9	X	0	24	S
\$6C	Y	8	16	U	\$EC	-	-	-	-
\$6D	Y	8	16	S	\$ED	-	-	-	-
\$70	Y	0	20	U	\$F0	X	0	2	U
\$71	Y	0	20	S	\$F1	X	2	2	U
\$72	-	-	-	-	\$F2	X	4	2	U
\$73	-	-	-	-	\$F3	X	6	2	U
\$74	Y	4	20	U	\$F4	X	8	2	U
\$75	Y	4	20	S	\$F5	X	10	2	U
\$76	-	-	-	-	\$F6	X	12	2	U
\$77	-	-	-	-	\$F7	X	14	2	U
\$78	Y	0	24	U	\$F8	X	16	2	U
\$79	Y	0	24	S	\$F9	X	18	2	U
\$7A	-	-	-	-	\$FA	X	20	2	U
\$7B	-	-	-	-	\$FB	X	22	2	U

MS{anynode},MI199 Direct Read/Write Variable

Range: -8,388,608 – 16,777,215
 Units: (dependent on register addressed)
 Default: none

MI199 is a variable that can be addressed to any register in the Acc-68M MACRO Station's memory and I/O map, in order to read a value directly from that register, or write a value directly to that register. This permits easy access to any register on the Acc-68M MACRO Station.

The address of the register to be accessed, which part of the register, and how the data is to be interpreted is set by MI198. The value of MI198 must be set properly before MI199 can be used to access the register. For repeated access of the same register with MI199, MI198 needs to be set once only.

Example:

```
MS0,MI198=$79C03C      ; Set to Y:$C03C,0,24,S (PFM8 command value)
MS0,MI199              ; Request value of this register
0                      ; PMAC reports this value
MS0,MI199=65536       ; Set to new value through PMAC
```

Global MACRO Status MI-Variables

These variables are used to help identify the MACRO, Servo, and I/O boards located in the UBUS rack. They are similar to the Turbo PMAC I4900 type variables. They are global to the CPU and not a function of the MACRO IC.

MS{anynode}, MI203 Phase Period

Range: 0
 Units: Clock units/2
 Default: \$1540 (9KHZ)

MS{anynode}, MI204 Phase Execution Time

Range: 0
 Units: Clock units/2
 Normal: \$4D - \$E0

Phase duty cycle (%) = (MI204/MI203) * 100. It should be < 50 %

MS{anynode}, MI205 Background Cycle Time

Range: 0
 Units: Clock units/2
 Default: 0

The last background cycle time.

MS{anynode}, MI206 Maximum Background Cycle Time

Range: 0
 Units: Clock units/2
 Default: 0

The maximum background cycle time.

MS{anynode}, MI208 User Ram Start

Range: 0
 Units: N.A.
 Default: \$700

This area is available for scratch pad use in the MACRO PLCCs.

MACRO IC MI-Variables

Each MACRO IC has a set of these variables and they are used to setup each MACRO IC.

MS{anynode},MI942 ADC Strobe Word for ADC1 and ADC2 Inputs

Range: \$000000 - \$FFFFFF
 Units: Individual Bits
 Default: \$FFFFFFE (For DSPGAT -02C)

MI942 specifies the strobe word for the serial A/D'S ADC_1 and ADC_2. The bits of the strobe word are shifted out, one bit per ADC_CLK cycle, MSB first, starting on the rising edge of the phase clock.

MACRO IC Setup MI-Variables

Each MACRO IC (0 & 1) has its own set of these variables and is accessed from each MACRO IC.

MS{anynode},MI970-MI973 (Reserved for Future Use)

MS{anynode},MI1974 Station Display Status (Read Only)

Range: \$0 - \$F
 Units: none

This variable, when queried, reports the hexadecimal digit displayed on the Acc-68MMACRO Station's 7-segment display. The meaning of each digit is:

- 0: Ring Active with no errors
- 1-9:NA
- A: 24V output fault
- B: Ring-break fault
- C: Configuration change fault
- D: Ring data-error fault
- E: NA
- F: Momentary ring fault

Note:

If the display itself is blank, this indicates that ring communications are not active, which means that this value cannot be reported back to the controller.

MS{anynode},MI977 Motor Nodes Reporting Ring Break

Range: \$0000 - \$FFFF
 Units: none (individual bits)
 Default: \$0

MI977 permits the Acc-68M MACRO Station to enable additional motor nodes if it detects a ring break immediately upstream from it, and send out the ring break bit (Bit 13) in the flag word for these nodes. When the station detects a ring break, it turns itself into a ring master, and sets the ring break bit on all active nodes. In this manner, other stations downstream of the break can be notified directly of the ring break, so they can shut down properly.

MI977 is a 16-bit value – bits 0 to 15 – with bit *n* controlling the enabling of MACRO node *n* on a ring break. If the bit is set to 0, the node will not be enabled on a ring break; if the bit is set to 1, the node will be enabled on a ring break. Only bits 0 - 13 of MI975 should be set to 1.

Examples:

MS0 ,MI977=\$3300 ; Enable Motor Nodes 8, 9, 12, & 13 on ring break
MS8 ,MI977=\$0033 ; Enable Motor Nodes 0, 1, 4, & 5 on ring break

MS{anynode},MI978-MI986 (Reserved for Future Use)

MACRO IC I/O Control and Initialize

MS{anynode},MI987 TBD

Range: 0 - 1
 Units: none
 Default: 0

MS{anynode},MI988 TBD

Range: \$00 - \$FF
 Units: none
 Default: \$00

MS{anynode},MI989 TBD

Range:
 Units:
 Default:

MACRO IC MI-Variables

MI-Variables are numbered in the MI990s control hardware aspects of the MACRO IC and the handwheel channels 1 and 2. Each MACRO IC has its own set of these variables.

MS{anynode},MI992 MaxPhase Frequency Control

Range: 0 - 32767
 Units: MaxPhase Frequency = $117,964.8 \text{ kHz} / [2 * \text{MI992} + 3]$
 PWM Frequency = $117,964.8 \text{ kHz} / [4 * \text{MI992} + 6]$
 Default: 6527
 MaxPhase Frequency = $117,964.8 / 13057 = 9.0346 \text{ kHz}$
 PWM Frequency = $117,964.8 / 26114 = 4.5173 \text{ kHz}$

MI992 controls the maximum phase clock frequency for the Acc-68M MACRO Station, and the PWM frequency for the filtered PWMs for supplementary handwheel interface channels 1 and 2. It does this by setting the limits of the PWM up-down counter, which increments and decrements at the PWMCLK frequency of 117,964.8 kHz (117.9648 MHz).

The actual phase clock frequency is divided down from the maximum phase clock according to the setting of MI997. The phase clock frequency must be the same as the ring update frequency as set by the ring controller (usually a PMAC or PMAC2). If the ring controller is a PMAC2 Ultralite, MI992 and MI997 on the Acc-68M MACRO Station should be set to the same values as MI992 and MI997 on the PMAC2 Ultralite.

To set MI992 for a desired maximum phase clock frequency, the following formula can be used:

$$\text{MI992} = (117,964.8 \text{ kHz} / [2 * \text{MaxPhase (kHz)}]) - 1 \text{ (rounded down)}$$

Examples:

To set a PWM frequency of 10 kHz and therefore a MaxPhase clock frequency of 20 kHz:

$$\text{MI992} = (117,964.8 \text{ kHz} / [4 * 10 \text{ kHz}]) - 1 = 2948$$

To set a PWM frequency of 7.5 kHz and therefore a MaxPhase clock frequency of 15 kHz:

$$\text{MI992} = (117,964.8 \text{ kHz} / [4 * 7.5 \text{ kHz}]) - 1 = 3931$$

MS{anynode},MI993 Hardware Clock Control Handwheel Channels

Range: 0 - 4095
 Units: MI993 = Encoder SCLK Divider
 + 8 * PFM_CLK Divider
 + 64 * DAC_CLK Divider

+ 512 * ADC_CLK Divider

where:

Encoder SCLK Frequency = 39.3216 MHz / (2 ^ Encoder SCLK Divider)

PFM_CLK Frequency = 39.3216 MHz / (2 ^ PFM_CLK Divider)

DAC_CLK Frequency = 39.3216 MHz / (2 ^ DAC_CLK Divider)

ADC_CLK Frequency = 39.3216 MHz / (2 ^ ADC_CLK Divider)

Default: 2258 = 2 + (8 * 2) + (64 * 3) + (512 * 4)

Encoder SCLK Frequency = 39.3216 MHz / (2 ^ 2) = 9.8304 MHz

PFM_CLK Frequency = 39.3216 MHz / (2 ^ 2) = 9.8304 MHz

DAC_CLK Frequency = 39.3216 MHz / (2 ^ 3) = 4.9152 MHz

ADC_CLK Frequency = 39.3216 MHz / (2 ^ 4) = 2.4576 MHz

MI993 controls the frequency of three hardware clock frequencies -- SCLK, PFM_CLK, DAC_CLK and ADC_CLK -- for the handwheel interface channels 1 and 2. It is a 12-bit variable consisting of 4 independent 3-bit controls, one for each of the clocks. Each of these clock frequencies can be divided down from a starting 39.3216 MHz frequency by powers of 2, 2^N, from 1 to 128 times (N=0 to 7). This means that the possible frequency settings for each of these clocks are:

Frequency	Divide by	Divider N in 1/2 ^N
39.3216 MHz	1	0
19.6608 MHz	2	1
9.8304 MHz	4	2
4.9152 MHz	8	3
2.4576 MHz	16	4
1.2288 MHz	32	5
611.44 kHz	64	6
305.72 kHz	128	7

Usually the MI993 setting is not changed from the default value.

The encoder sample clock signal SCLK controls how often 2-axis board's digital hardware looks at the encoder inputs. PMAC2 can take at most one count per SCLK cycle, so the SCLK frequency is the absolute maximum encoder count frequency. SCLK also controls the signal propagation through the digital delay filters for the encoders and flags; the lower the SCLK frequency, the greater the noise pulse that can be filtered out. Optimally, the SCLK frequency should be set to the lowest value that can accept encoder counts at the maximum possible rate.

The pulse-frequency-modulation clock PFM_CLK controls the PFM circuitry on the 2-axis board that can create pulse and direction outputs. The maximum pulse frequency possible is 1/4 of the PFM_CLK frequency. Optimally, the PFM_CLK frequency should be set to the lowest value that can generate pulses at the maximum frequency required.

The ADC_CLK controls the serial data frequency from A/D converters, either for digital current loop closure, or from an Acc-28B A/D converter board.

The DAC-CLK controls the serial data frequency to D/A converters for the 2-axis board, either the on-board converters that come with Option A, or the external converters on an Acc-8E board.

To determine the clock frequencies set by a given value of MI993, use the following procedure:

1. Divide MI993 by 512 and round down to the nearest integer. This value N1 is the ADC_CLK divider.
2. Multiply N1 by 512 and subtract the product from MI993 to get MI993'. Divide MI993' by 64 and round down to the nearest integer. This value N2 is the DAC_CLK divider (not relevant here).
3. Multiply N2 by 64 and subtract the product from MI993' to get MI993". Divide MI993" by 8 and round down to the nearest integer. This value N3 is the PFM_CLK divider.
4. Multiply N3 by 8 and subtract the product from MI993". The resulting value N4 is the SCLK divider.

Examples:

The maximum encoder count frequency in the application is 800 kHz, so the 1.2288 MHz SCLK frequency is chosen. A pulse train up to 500 kHz needs to be generated, so the 2.4576 MHz PFM_CLK frequency is chosen. ADCs and DACs are not used, so the default DAC_CLK frequency of 4.9152 MHz and the default ADC_CLK frequency of 2.4576 MHz are chosen. From the table:

SCLK Divider N: 5

PFM_CLK Divider N: 4

DAC_CLK Divider N: 3

ADC_CLK Divider N: 4

$MI993 = 5 + (8 * 4) + (64 * 3) + (512 * 4) = 5 + 32 + 192 + 2048 = 2277$

MI993 has been set to 3429. What clock frequencies does this set?

$N1 = INT(3429/512) = 6$ ADC_CLK = 611.44 kHz

$MI993' = 3429 - (512 * 6) = 357$

$N2 = INT(357/64) = 5$ DAC_CLK = 1.2288 MHz

$MI993'' = 357 - (64 * 5) = 37$

$N3 = INT(37/8) = 4$ PFM_CLK = 2.4576 MHz

$N4 = 37 - (8 * 4) = 5$ SCLK = 1.2288 MHz

MS{anynode}, MI994 PWM Deadtime / PFM Pulse Width Control for Handwheel

Range: 0 - 255

Units: PWM Deadtime = $[16 / PFM_CLK \text{ (MHz)}] * MI994 = 0.135 \mu\text{sec} * MI994$

PFM Pulse Width = $[1 / PFM_CLK \text{ (MHz)}] * MI994$

= $PFM_CLK_period \text{ (}\mu\text{sec)} * MI994$

Default: 15

PWM Deadtime = $0.135 \mu\text{sec} * 15 = 2.03 \mu\text{sec}$

PFM Pulse Width = $[1 / 9.8304 \text{ MHz}] * 15 = 1.526 \mu\text{sec}$ (with default MI993)

MI994 controls the dead time period between top and bottom on-times in the Acc-68M MACRO Station's automatic PWM generation for machine interface handwheel channels 19 and 2. In conjunction with MI993, it also controls the pulse width for PMAC2's automatic pulse-frequency modulation generation for these machine interface channels.

The PWM dead time, which is the delay between the top signal turning off and the bottom signal turning on, and vice versa, is specified in units of 16 PFM_CLK cycles. This means that the dead time can be specified in increments of 0.135 μsec . The equation for MI994 as a function of PWM dead time is:

$$MI994 = \text{Deadtime } (\mu\text{sec}) / 0.135 \mu\text{sec}$$

The PFM pulse width is specified in PFM_CLK cycles, as defined by MI993. The equation for MI994 as a function of PFM pulse width and PFM_CLK frequency is:

$$MI994 = \text{PFM_CLK Freq (MHz)} / \text{PFM pulse width } (\mu\text{sec})$$

In PFM pulse generation, the minimum off time between pulses is equal to the pulse width. This means that the maximum PFM output frequency is:

$$\text{PFM Max_Freq (MHz)} = \text{PFM_CLK Freq} / (2 * MI994)$$

Examples:

A PWM deadtime of approximately one microsecond is desired:

$$MI994 \cong 1 \mu\text{sec} / 0.135 \mu\text{sec} \cong 7$$

With a 2.4576 MHz PFM_CLK frequency, a pulse width of 0.4 μsec is desired:

$$MI994 \cong 2.4576 \text{ MHz} * 0.4 \mu\text{sec} \cong 1$$

MS{anynode},MI995 MACRO Ring Configuration/Status

Range: \$0000 - \$FFFF (0 - 65,535)
 Units: none
 Default: \$4080

MI995 contains configuration and status bits for MACRO ring operation of the Acc-68M MACRO Station. There are 11 configuration bits and 5 status bits, as follows:

Bit #	Value	Type	Function
0	1(\$1)	Status	Data Overrun Error (cleared when read)
1	2(\$2)	Status	Byte Violation Error (cleared when read)
2	4(\$4)	Status	Packet Parity Error (cleared when read)
3	8(\$8)	Status	Packet Under run Error (cleared when read)
4	16(\$10)	Config	Master Station Enable
5	32(\$20)	Config	Synchronizing Master Station Enable
6	64(\$40)	Status	Sync Node Packet Received (cleared when read)
7	128(\$80)	Config	Sync Node Phase Lock Enable
8	256(\$100)	Config	Node 8 Master Address Check Disable
9	512(\$200)	Config	Node 9 Master Address Check Disable
10	1024(\$400)	Config	Node 10 Master Address Check Disable
11	2048(\$800)	Config	Node 11 Master Address Check Disable
12	4096(\$1000)	Config	Node 12 Master Address Check Disable
13	8192(\$2000)	Config	Node 13 Master Address Check Disable
14	16384(\$4000)	Config	Node 14 Master Address Check Disable
15	32768(\$8000)	Config	Node 15 Master Address Check Disable

An ACC-68M MACRO Station is a slave on the ring in all normal operation, so configuration bits 4 and 5 are set to 0. It should synchronize itself to the sync node, so configuration bit 7 should be set to 1. In most applications, it will accept packets only from its own master so bits 8 to 15 are set to 0. All other bits are status bits that are normally 0. This makes the usual setting of MI995 equal to \$0080.

MS{anynode},MI996 MACRO Node Activate Control

Range: \$000000 to \$FFFFFF (0 to 8,388,607)
 Units: none
 Default: \$0F0000 (all nodes de-activated, Synch packet = 15, Master = 0)

MI996 controls which of the 16 MACRO nodes on the Acc-68M MACRO Station are activated. In addition, it controls the master station number and the node number of the packet that creates a synchronization signal.

On a power-up or reset of the Acc-68MMACRO Station, MI996 is set to the saved value or to the default.

The bits of MI996 are arranged as follows:

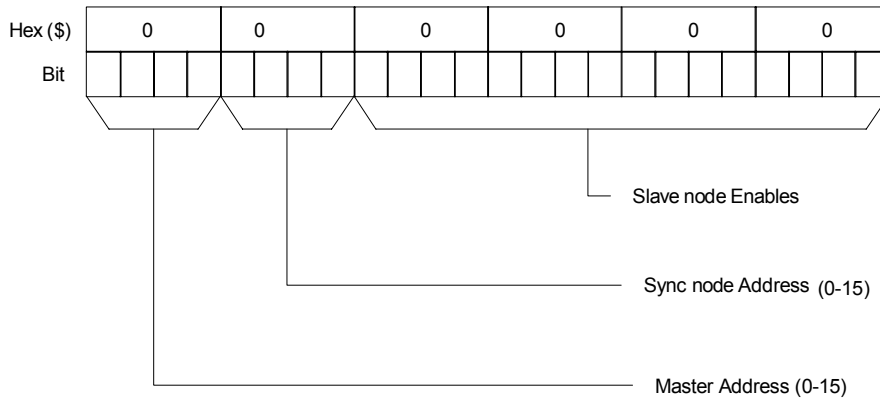
Bit #	Value	Type	Function
0	1(\$1)	Config	Node 0 Activate
1	2(\$2)	Config	Node 1 Activate
2	4(\$4)	Config	Node 2 Activate
3	8(\$8)	Config	Node 3 Activate
4	16(\$10)	Config	Node 4 Activate
5	32(\$20)	Config	Node 5 Activate
6	64(\$40)	Config	Node 6 Activate
7	128(\$80)	Config	Node 7 Activate
8	256(\$100)	Config	Node 8 Activate
9	512(\$200)	Config	Node 9 Activate
10	1024(\$400)	Config	Node 10 Activate

11	2048(\$800)	Config	Node 11 Activate
12	4096(\$1000)	Config	Node 12 Activate
13	8192(\$2000)	Config	Node 13 Activate
14	16384(\$4000)	Config	Node 14 Activate
15	32768(\$8000)	Config	Node 15 Activate
16-19	\$X0000	Config	Packet Sync Node Slave Address (0 – 15)
20-23	\$X00000	Config	Master Station Number (0-15)

Bits 0 to 15 are individual control bits for the matching node number 0 to 15. If the bit is set to 1, the node is activated; if the bit is set to 0, the node is de-activated. On power-up reset, these bits are set as defined by the SW1 setting, with some motor nodes possibly disabled by MI976, and some I/O nodes possibly enabled by MI975. Usually, Node 15 should be activated to support the Type 1 auxiliary communications.

Bits 16-19 specify the slave number of the packet which will generate the sync pulse on the ACC-68M MACRO Station. Usually, this is set to 15 (\$F) on the ACC-68M MACRO Station.

Bits 20-23 specify the master number (0-15) for the ACC-68M MACRO Station.



MS{anynode},MI997 Phase Clock Frequency Control

Range: 0 - 15

Units: Phase Clock Frequency = MaxPhase Frequency / (MI997+1)

Default: 0

Phase Clock Frequency = 9.0346 kHz / 1 = 9.0346 kHz (with default value of MI992)

MI997, in conjunction with MI992, determines the frequency of the Phase clock on an Acc-68M MACRO Station. Each cycle of the Phase clock, a set of MACRO ring information is expected and any data transfers between MACRO nodes and interface circuitry are performed. The Phase clock cycle on the Acc-68M MACRO Station should match as closely as possible that of the PMAC commanding it.

Specifically, MI997 controls how many times the Phase clock frequency is divided down from the maximum phase clock whose frequency is set by MI992. The Phase clock frequency is equal to the maximum phase clock frequency divided by (MI997+1). MI997 has a range of 0 to 15, so the frequency division can be by a factor of 1 to 16. The equation for MI997 is:

$$MI997 = (\text{MaxPhase Freq} / \text{Phase Clock Freq}) - 1$$

The ratio of MaxPhase Frequency to Phase Clock Frequency must be an integer.

Example:

With a 20 kHz MaxPhase Clock frequency established by MI992 and a desired 6.67 kHz Phase clock frequency, the ratio between MaxPhase and Phase is 3:

$$MI997 = (20 / 6.67) - 1 = 3 - 1 = 2$$

MS{anynode},MI998 Servo Clock Frequency Control

Range: 0 - 15
Units: Servo Clock Frequency = Phase Clock Frequency / (MI998+1)
Default: 0
Phase Clock Frequency = 9.0346 kHz / (0+1) = 9.0346 kHz
(with default values of MI992 and MI997)

Note:

Currently, there is no software use of the Servo clock on the Acc-68M MACRO Station. However, it is needed to capture certain encoder values in the DSPGATEX Servo ICs.

MI998, in conjunction with MI997 and MI992, determines the frequency of the Servo clock on the Acc-68M MACRO Station.

Specifically, MI998 controls how many times the Servo clock frequency is divided down from the Phase clock, whose frequency is set by MI992 and MI997. The Servo clock frequency is equal to the Phase clock frequency divided by (MI998+1). MI998 has a range of 0 to 15, so the frequency division can be by a factor of 1 to 16. The equation for MI998 is:

$$\text{MI998} = (\text{Phase Clock Frequency} / \text{Servo Clock Frequency}) - 1$$

The ratio of Phase Clock Frequency to Servo Clock Frequency must be an integer. On the Acc-68M MACRO Station, MI998 should always be set to 0 so the servo clock frequency is equal to the phase clock frequency.

MS{anynode},MI999 Handwheel DAC Strobe Word (Not used)

Range: \$000000 - \$FFFFFF
Units: Serial Data Stream (MSB first, starting on rising edge of phase clock)
Default: \$7FFFC0 (for 18-bit DAC data)

Other Acc-68M MACRO Station Mm & MP-Variables

The Acc-68M MACRO CPU has 512 MM and 512 MP variables. The MM variables are similar to the PMAC X/Y types which are 1 to 24 bit integer data types. The MP variables are general-purpose 24-bit integer data types. The MM types can be used in the MI21-MI68 copy variables and the PLCC. The MP variables can be used only in the PLCC.

ACC-68M MACRO STATION MACPLCCS

The Open MACPLCC compiler in PewinPro is used to compile the MACRO PLCC program that runs in the ACC-68M MACRO CPU. It is designed to handle a limited version of the standard PMAC PLCC programming commands and it will include some new ones. The MACPLCC code is run in the background process of the ACC-68M MACRO CPU. The additions and limitations to the standard PMAC PLC commands are defined below.

Requirements

Turbo PMAC CPU with version 1.939 and greater and the ACC-68M MACRO station restricted to 8K of PLCC memory and from X/Y:\$700 - \$13FF) of data memory.

Arithmetic Data Types

1. Integer – 24-bit signed integer (unsigned is not available)
2. Integer – 1, 4, 8, 12, 16, 20 bit (unsigned or signed)

MACRO MI Integer Variables (n = 0 – 1099)

1. MACRO **MI**n-Variable – Converted to 1 to 24-bit signed/unsigned integer variable. A function of MI-variable.
2. MACRO **MI**[**Index Exp.**]-Array of MI Variables
3. **Indexes into MI**n[] arrays are limited to 0 - 1999. On a read of the index value outside this range, the returned value is zero. On a write of the index value outside this range, no value is written.

MACRO MM and MP Integer Variables (n = 0 – 511)

1. MACRO **MM**n-Variable – Assumed to be defined as **MM**n-><X/Y:Addr,offset,width,SignType>.
2. MACRO **MP**n-Variable – 24-bit signed integer variable
3. MACRO **MM**[**Index Exp.**] – Array to MM Pointer Variables
4. MACRO **MP**[**Index Exp.**] – Array of 24 bit signed Integer MP Variables
5. Index expression into the **MM**n[] and **MP**n[] arrays are forced to a modulo 512

MACRO Plcc Ln Integer Variables (n = 0 – 511)

1. PLCC **Ln**-Variable – Address must be defined. (Accessed with inline code.)
2. PLCC **Ln**[**Index Exp.**]-Array 24 bit signed integer data – Address must be defined.
3. Index expressions into the **Ln**[] arrays are forced to a modulo of the size of the array.

Direct Memory Addressing for Integer Ln & Ln[] Variable Definitions

1. *MACRO Plcc Ln->X/Y:Address[size]* – Accesses entire 24-bit integer data value. The array size range is 2 – 8192 and must be a power of two. If the definition is put after the *OPEN MAC Plcc*, the size range is limited to 2 - 512. This is the recommended limitation.
2. The current PLCC *Ln->* definitions which access portions of the 24-bit word are still available.

Standard MACRO Program Commands

1. **OPEN MACPLCC** – Begins the *MAC Plcc* the program.
2. **CLOSE** – Closes *MAC Plcc* the program.
3. **RETURN** – Returns from PLCC program.
4. **IF, AND, OR, ELSE, ENDIF, WHILE, ENDW**

Valid Math, Assignment and Conditional Operators

1. +, -, *, /, %, &, ^, and |
2. =, >, <, !=, !>, !<

Valid Expressions and Arrays

OpenPlcc Ln integer variable array expression.

Example: $L1 = L2[L3] + L3 + L4[L7+L5*L3]$ or $L1[L1+L3] = L4 + L8$

Note:

The [index] of the array must be an integer and it is limited to the range of the defined Ln array.

Will be run through preprocessor so labels are allowed (#define Mtr1DAC MM1).

Ln Arrays Definition Examples

L5->X:\$600[64]

L6->Y:\$600[64]

MM[MM1 + MP2*MP3] &= MP[MP2 + MP4*MM5]

MP[MM1 + MP2*MP3] = MM[MP2 + MP4*MM5]

The following is allowed for the Ln[array index]:

$L5[L1 + L2*L3] \&= L6[L4(L2) - L4*L5]$

$L5[L1 + L2*L3] \&= L6[L4[L2] - L4*L5]$

Note:

MI[Int. Exp.], MM[Int. Exp.], MP[Int. Exp.] and Lnn[Int. Exp.] must use integer variable indexes.

Example Program

MM1->X:\$00700,24

MM2->Y:\$00700,24

MM3->X:\$00701,24

OPEN MACPLCC

MM3 = MM1*MM2

CLS

MACRO PLCC Code Memory

Memory Location	Data Information
\$4000	End of PLCC Program + 1 (\$\$\$*** is set to \$4005)
\$4001	JMP to location in \$4002
\$4002	Location of start of PLCC (\$\$\$*** is set to \$4003)
\$4003	RTS Instruction
\$4004	Start of PLCC Program (\$\$\$*** set = RTS instruction)

MAC PLCC Related ASCII Commands

Commands	Operation
SAVE	Saves PLCCs from \$4000 to *(\$4000)
DISable PLCC & ^D	Sets *\$4002 = \$4003
ENable PLCC	Sets *\$4002 = \$4004
\$\$\$***	Restores \$4000 - \$4004 to default.
\$\$\$	Restore from SAVEd \$4000 to *(\$4000 end of PLCC). If is SAVEd in DISable state will come up ENabled.
Download	Start of download sends MI15 = 0 and at end of download sets *\$4002 = \$4004
MI15 = 1	Enables the JSR to \$4001 (Runs MacPLCC)

ACC-68M MACRO STATION SERIAL COMMANDS

The ACC-68M MACRO Station can accept ASCII text commands directly through the USB to serial port at connector on the CPU/Interface Board, or in auxiliary mode from a Turbo PMAC over the MACRO ring using MACSTASCII commands. Serial communications is at 38400 (eight bits, one stop bit, no parity). These commands are intended for basic setup and troubleshooting. Usually this port will not be used. Instead, commands will be sent only through the MACRO ring.

The following commands can be sent to the ACC-68M MACRO Station through the USB-serial port or over the MACRO ring.

Serial Commands

\$\$\$ Station Reset

The **\$\$\$** command will reset the ACC-68M MACRO Station and restore all station MI-variables to their last saved values.

\$\$\$*** Station Re-initialize

The **\$\$\$***** command will reset the ACC-68M MACRO Station and restore all station MI-variables to their factory default values.

CHN Reports Channel Number

The **CHN** command causes the ACC-68M MACRO Station to report its present channel number.

CID Reports Card ID Number

The **CID** command causes the ACC-68M MACRO Station CPU to report its part number: 603740.

CLRF Clears Station Faults

The **CLRF** command will clear all faults on the ACC-68M MACRO Station and prepare it for further operation.

DATE Reports Firmware Date

The **DATE** command causes the ACC-68M MACRO Station to report the date of its firmware.

Example:

DATE

12/02/2003

DISABLE PLCC or CNTRL D Disables PLCC

The MACRO PLCC are disabled.

Example:

DIS PLCC

^D

ENABLE PLCC Enables PLCC

The MACRO PLCCs are enabled if MI15 =1.

Example:

ENA PLCC

MI{constant} Reports Station MI-Variable Value

The **MI {constant}** command causes the ACC-68M MACRO Station to report the current value of the specified MI-variable.

MI{constant}={constant} Sets Station MI-Variable Value

The **MI {constant}={constant}** command causes the ACC-68M MACRO Station to set the value of the specified MI-variable to the specified value.

MM{constant} Report Station MM-Variable Value

The **MM{constant}** command causes the ACC-68M MACRO Station to report the current value of the specified MM-variable.

MM{constant}={constant} Sets Station MM-Variable Value

The **MM{constant}={constant}** command causes the ACC-68M MACRO Station to set the value of the specified MM-variable to the specified value.

MP{constant} Reports Station MP-Variable Value

The **MP{constant}** command causes the ACC-68M MACRO Station to report the current value of the specified MP-variable.

MP{constant}={constant} Sets Station MP-Variable Value

The **MP{constant}={constant}** command causes the ACC-68M MACRO Station to set the value of the specified MP-variable to the specified value.

MM{constant}-> Reports Station MM-Variable Definition

The **MM{constant}->** command causes the ACC-68M MACRO Station to report the current definition of the specified MM-variable.

MM{constant}->{X/Y:offset,width,format} Sets Station MM-Variable Definition

The **MM{constant}->{X/Y:offset,width,format}** command causes the Acc-68M MACRO Station to set the value of the specified MM-variable memory location to the specified 1 to 24 bit integer (signed or unsigned) definition.

R{address} Reads Station Address

The **R[H] {address} [, {count}]** command causes the Acc-68M MACRO Station to report the value stored at the specified addresses. If **H** is used, the contents of the register[s] are reported back in hexadecimal; otherwise, they are reported back in decimal form. **{address}** consists of a register type (X, Y, L, or P) and the numerical address of the register. The optional **{count}** value specifies the number of registers to be reported, starting at the specified address and counting up. If no **{count}** value is specified in the command, one register value is reported.

Examples:

```
RX: $20 ; Read X register $20
64 ; CMS responds in decimal
RHX: $20 ; Read X register $20 in hex
40 ; CMS responds in hex
RHY: $FFC0, 3 ; Read Y registers $FFC0, $FFC1, $FFC2
FFFFFFA4 FFFF01 FFFFC7 ; CMS responds in hex
```

SAVE Saves Station MI-Variables

The **SAVE** command causes the ACC-68M MACRO Station to copy its MI-variable values from volatile active memory to the non-volatile flash memory. On the next power-up or reset, these values will be copied back from flash memory to active memory.

SID Reports Serial Identification Number

Reports the SID of the Dallas ID chip.

VERS Reports Firmware Version

The **VERS** command causes the ACC-68M MACRO Station to report its firmware version number.

Example:

```
VERS
1.201
```

VID Reports Vendor ID Number

The **VID** command causes the Acc-68M MACRO Station to report its vendor identification number. For Delta Tau, this number is 1.

W{address},{value} Writes Value to Station Address

The **W{address} [, {value}]** command causes the Acc-68M MACRO Station to write the value to the specified address[es]. **{address}** consists of a register type (X, Y, L, or P) and the numerical address of the register. The optional **{value}** specifies the value to be written to the address.

Examples:

wx: \$20, 5 ; Write X register \$20 = 5

PMAC TYPE 1 ACC-68M MACRO STATION COMMANDS

The following commands from the Turbo PMAC controllers can be used for Type 1 auxiliary communication with the Acc-68M MACRO Station.

On-Line Commands

MS Command

Syntax: **MS {command} {node #}**

where:

{command} is one of the following text strings:

\$\$\$	normal station reset
\$\$\$***	station reset and re-initialize
CLRF	station fault clear for
CONFIG	report station configuration value
DATE	report station firmware date
SAVE	save station setup
VER	report station firmware version

{node #} is a constant representing the number of the node to be commanded (if the command affects the entire station, the number of any active node on the station may be used).

This PMAC command causes PMAC to issue the specified command to a Type 1 MACRO slave station.

The **MS CONFIG** command allows the setting and reporting of a specified configuration value. This provides an easy way to see if the Acc-68M MACRO Station has been already configured to the specifications. The factory default configuration value is 0. It is recommended that after the software configuration of the station is finished, a special number be given to the configuration value with the **MS CONFIG{node #}={constant}** command. This number will be saved to the non-volatile memory with the **MS SAVE** command. Subsequently, when the system is powered up, the station can be polled with the **MS CONFIG {node #}** command. If the expected value is returned, it can be assumed that the station has the proper software setup. If the expected value is not returned (for instance, when a replacement station has just been installed), then the setup must be transmitted to the station.

Examples:

MS \$\$\$0	; Resets Acc-68M MACRO Station which has active node 0.
MS \$\$\$***4	; Reinitializes Acc-68M MACRO Station which has active node 4.
MS CLRF8	; Clears fault on Node 8 of Acc-68M MACRO Station.
MS CONFIG12	; Causes Acc-68M MACRO Station to report its configuration number.
37	; PMAC reports Acc-68M MACRO Station configuration number to host.
MS CONFIG12=37	; Sets Acc-68M MACRO Station configuration number.
MS DATE 0	; Causes Acc-68M MACRO Station to report its firmware date.
03/27/97	; PMAC reports Acc-68M MACRO Station firmware date to host.
MS SAVE 4	; Causes Acc-68M MACRO Station to save setup variables.
MS VER 8	; Causes Acc-68M MACRO Station to report its firmware version.
1.200	; PMAC reports Acc-68M MACRO Station firmware version to host.

MS Variable Read

Syntax: **MACROSLAVE{node #},{slave MI, MM or MP-variable}**

MS{node #},{slave MI-variable}

where:

{node #} is a constant (0-14) representing the number of the node whose variable is to be read (if the variable is not node-specific,

the number of any active node on the station may be used)
{slave MI, MM or MP-variable} is the name of the MI, MM or MP-variable on the slave station whose value is to be reported

This command causes PMAC to query the MACRO slave station at the specified node # and report the value of the specified slave station MI-variable to the host computer.

Examples:

MS0,MI910 ; Causes Acc-68M MACRO Station to report value of Node 0 variable MI910.
 7 ; PMAC reports this value back to host.
MS1,MI997 ; Causes Acc-68M MACRO Station to report value global variable MI997.
 6258 ; PMAC reports this value back to host.

MS Variable Write

Syntax: **MACROSLAVE{node #},{slave variable}={constant}**
MS{node #},{slave variable}={constant}

where:

{node #} is a constant (0-14) representing the number of the node whose variable is to be written to (if the variable is not node-specific, the number of any active node on the station may be used).
{slave MI, MM or MP -variable} is the name of the MI, MM or MP or C-variable on the slave station whose value is to be reported
{constant} is a number representing the value to be written to the specified **MI, MM or MP** -variable.

This command causes PMAC to write the specified constant value to the MACRO slave station MI, MM, MP-variable, or if a **C** command is specified, it causes the station to execute the specified command number (in which case the constant value does not matter).

The valid C-commands are:

- C1** Clear station faults
- C2** Reset station, loading saved station MI-variables
- C3** Re-initialize station, loading default station MI-variables
- C4** Save station MI-variables to non-volatile memory

Examples:

MS0,MI910=7 ;sets Node 0 variable MI910 to 7
MS8,C4=0 ; Clears faults on Acc-68MMACRO Station with active node 8

MS Variable Read Copy

Syntax: **MACROSLVREAD{node #},{slave MI-variable},{PMAC variable}**
MSR{node #},{slave MI, MM or MP -variable},{PMAC variable}

where

{node #} is a constant (0-14) representing the number of the node whose variable is to be read. (If the variable is not node-specific, the number of any active node on the station may be used.)
{slave MI, MM or MP -variable} is the name of MI, MM or MP-variable on the slave station whose value is to be reported.
{PMAC variable} is the name of the variable on the PMAC into which the value of the slave station variable is to be copied.

This command copies the value of the specified MI, MM or MP-variable on the MACRO slave station into the specified variable on PMAC.

The variable on the PMAC or PMAC2 can be any of the I, P, Q, or M-variable on the card.

If this command is issued to a PMAC while a PLC buffer is open, it will be stored in the buffer as a PLC command, not executed as an on-line command.

Examples:

```
MS0 ,MI910 , P1           ; Copies value of Acc-68M MACRO Station Node 0 variable MI910 into
                          ; PMAC variable P1
MS1 ,MM9 , M10           ; Copies value of Acc-68M MACRO Station global variable MM9 into
                          ; PMAC variable M10
```

MS Variable Write Copy

Syntax: **MACROSLVWRITE**{node #},{slave variable},{PMAC variable}
MSW{node #},{slave MI, MM or MP -variable},{PMAC variable}

where:

{node #} is a constant (0-14) representing the number of the node whose variable is to be read. (If the variable is not node-specific, the number of any active node on the station may be used.)

{slave variable} is the name of the MI, MM or MP-variable or C-command on the slave station whose value is to be reported.

{PMAC variable} is the name of the variable on the PMAC into which the value of the slave station variable is to be copied.

This command copies the value of the specified variable on PMAC into the specified MI, MM or MP-variable on the MACRO slave station, or if a slave station C-command number is specified, it executes that command (in which case the PMAC variable value is not used).

The valid C-commands are:

- C1** Clear station faults
- C2** Reset station, loading saved station MI-variables
- C3** Re-initialize station, loading default station MI-variables
- C4** Save station MI-variables to non-volatile memory

The MI-variable on the MACRO slave station can be global to the station, or node-specific.

The variable on the PMAC or PMAC2 can be any of the I, P, Q, or M-variables on the card.

If this command is issued to a PMAC while a PLC buffer is open, it will be stored in the buffer as a PLC command, not executed as an on-line command.

Examples:

```
MSW0 ,MI910 , P35       ; Copies value of PMAC P35 into ACC-68M MACRO Station node 0
                          ; variable MI910
MSW4 , C4 , P0          ; Causes ACC-68M MACRO Station with active node 4 to save its
                          ; MI-variable values to non-volatile memory (P0 is a dummy variable here)
```

Turbo PMAC PLC Commands for Type 1 Acc-68M MACRO Stations**MS Variable Read Copy**

Syntax: **MACROSLVREAD**{node #},{slave MI-variable},{PMAC variable}
MSR{node #},{slave MI, MM or MP -variable},{PMAC variable}

where

{node #} is a constant (0-14) representing the number of the node whose variable is to be read. (If the variable is not node-specific, the number of any active node on the station may be used.)

{slave MI, MM or MP -variable} is the name of the MI-variable on the slave station whose value is to be reported.

{PMAC variable} is the name of the variable on the PMAC into which the value of the slave station variable is to be copied.

This command copies the value of the specified MI-variable on the MACRO slave station into the specified variable on PMAC.

The MI-variable on the MACRO slave station can be global to the station, or node-specific.

The variable on the PMAC or PMAC2 can be any of the I, P, Q, or M-variables on the card.

If this command is issued to a PMAC while no PLC buffer is open, it will be executed as an on-line command, not stored in the buffer as a PLC command.

Examples:

MS0 ,MI910 ,P1 ; Copies value of ACC-68M MACRO Station Node 0 variable MI910 into PMAC variable P1
MS1 ,MI997 ,M10 ; Copies value of ACC-68M MACRO Station global variable MI997 into PMAC variable M10

MS Variable Write Copy

Syntax: **MACROSLVWRITE{node #},{slave variable},{PMAC variable}**
MSW{node #},{slave MI, MM or MP -variable},{PMAC variable}

where:

{node #} is a constant (0-14) representing the number of the node whose variable is to be read. (If the variable is not node-specific, the number of any active node on the station may be used.)

{slave variable} is the name of the MI, MM, MP-variable or C-command on the slave station whose value is to be reported.

{PMAC variable} is the name of the variable on the PMAC into which the value of the slave station variable is to be copied.

This command copies the value of the specified variable on PMAC into the specified MI-Variables on the MACRO slave station, or if a slave station C-command number is specified, it executes that command (in which case the PMAC variable value is not used).

The valid C-commands are:

- C1** Clear station faults
- C2** Reset station, loading saved station MI-variables
- C3** Re-initialize station, loading default station MI-variables
- C4** Save station MI-variables to non-volatile memory

The MI, MM or MP -variable on the MACRO slave station can be global to the station, or node-specific.

The variable on the PMAC or PMAC2 can be any of the I, P, Q, or M-Variables on the card.

If this command is issued to a PMAC while no PLC buffer is open, it will be executed as an on-line command, not stored in the buffer as a PLC command.

Examples:

MSW0 ,MI910 ,P35 ; Copies value of PMAC P35 into Acc-68M MACRO Station node 0 variable MI910
MSW4 ,C4 ,P0 ; Causes Acc-68M MACRO Station with active node 4 to save its MI-variable values to non-volatile memory (P0 is a dummy variable here)

ACC-68M MACRO STATION MEMORY AND I/O MAP

Global Servo Calculation Registers

X/Y: \$0000-\$06FF

Open Memory

X: \$00700 - \$007FF ; Open Memory
 Y: \$00700 - \$007FF ; Open Memory

DSPGATE2 Registers

Y: \$C080 Input 00-23 Data Register

Note:

The pins associated with this register are used for Input 00-23 on the ACC-68M Device.

Bits: 0 Input 00 Input Data Value
 ...
 23 Input 23 Input Data Value

X: \$C080 General I/O Data Direction Control (Do not change.)

Note:

The pins associated with this register are used for other purposes on the ACC-68M Device.

Bits: 0 I/O00 Direction Control
 ...
 23 I/O23 Direction Control
 (All bits: 0=Input; 1=Output)

Y: \$C081 Output 00-07 Data Register
 Bits: 0 Output 00 Data Value
 ...
 7 Output 07 Data Value
 8 I/O24 Latched Data Value
 ...
 15 I/O31 Latched Data Value
 16-23 Not used

X: \$C081 General I/O Direction Control (Do not change.)

Bits: 0 I/O24 (SEL0 pin) Direction Control
 ...
 7 I/O31 (SEL7 pin) Direction Control
 (All bits: 0=Input; 1=Output)
 8-23 Not used

Y: \$C082 Output 08-15 Data Register

Note:

The pins associated with this register are used for other purposes on the ACC-68M Device.

Bits: 0 Output 08 Data Value
 ...
 7 Output 15 Data Value
 8 SEL0 Data Value
 ...
 15 SEL7 Data Value
 16-23 Not used

X: \$C082 General I/O Data Direction Control Register (Do not change.)

Note:

The pins associated with this register are used for other purposes on the ACC-68M Device.

Bits: 0 DAT0 Direction Control
 ...
 7 DAT7 Direction Control
 8 SEL0 Direction Control
 ...
 15 SEL7 Direction Control
 (All bits: 0=Input; 1=Output)
 16-23 Not used

Y: \$C083 Output 15-23 Data Register
 Bits: 0 Output 16 Data Value
 ...
 7 Output 23 Data Value
 8 CTRL0 Data Value
 9 CTRL1 Data Value (Output Enable)
 10 CTRL2 Data Value (AENA_1)
 11 CTRL3 Data Value (AENA_2)
 12-23 Not used

X: \$C083 General I/O Port Data Direction Control Register (Do not change.)

Bits: 0 DISP0 Direction Control
 ...
 7 DISP7 Direction Control
 8 CTRL0 Direction Control
 ...
 11 CTRL3 Direction Control (Do not change.)
 (All bits: 0=Input; 1=Output)
 12-23 Not used

Y: \$C084 Data Type Control Register (Do not change.)

Bits: 0 I/O00 Data Type Control (0=FlagW9; 1=I/O00)
 1 I/O01 Data Type Control (0=FlagV9; 1=I/O01)
 2 I/O02 Data Type Control (0=FlagU9; 1=I/O02)
 3 I/O03 Data Type Control (0=FlagT9; 1=I/O03)
 4 I/O04 Data Type Control (0=USER9; 1=I/O04)
 5 I/O05 Data Type Control (0=MLIM9; 1=I/O05)
 6 I/O06 Data Type Control (0=PLIM9; 1=I/O06)
 7 I/O07 Data Type Control (0=HMFL9; 1=I/O07)
 8 I/O08 Data Type Control (0=PWM_B_BOT9; 1=I/O08)

9 I/O09 Data Type Control (0=PWM_B_TOP9; 1=I/O09)
 10 I/O10 Data Type Control (0=PWM_A_BOT9; 1=I/O10)
 11 I/O11 Data Type Control (0=PWM_A_TOP9; 1=I/O11)
 12 I/O12 Data Type Control (0=PWM_B_BOT10; 1=I/O12)
 13 I/O13 Data Type Control (0=PWM_B_TOP10; 1=I/O13)
 14 I/O14 Data Type Control (0=PWM_A_BOT10; 1=I/O14)
 15 I/O15 Data Type Control (0=PWM_A_TOP10; 1=I/O15)
 16 I/O16 Data Type Control (0=HMFL10; 1=I/O16)
 17 I/O17 Data Type Control (0=PLIM10; 1=I/O17)
 18 I/O18 Data Type Control (0=MLIM10; 1=I/O18)
 19 I/O19 Data Type Control (0=USER10; 1=I/O19)
 20 I/O20 Data Type Control (0=FlagT10; 1=I/O20)
 21 I/O21 Data Type Control (0=FlagU10; 1=I/O21)
 22 I/O22 Data Type Control (0=FlagV10; 1=I/O22)
 23 I/O23 Data Type Control (0=FlagW10; 1=I/O23)
 (All bits: 0=dedicated hardware I/O; 1=general I/O)
 (All bits must be 0 for use with ACC-1E 2-axis piggyback board)

X: \$C084 Data Inversion Control Register (when used as general I/O; see Y:\$C084)

Bits: 0 I/O00 Inversion Control
 ...
 23 I/O23 Inversion Control
 (All bits: 0=Non-inverting; 1=Inverting)

Y: \$C085 General I/O Data Type Control Register (Do not change.)

Bits: 0 I/O24 Data Type Control
 ...
 7 I/O31 Data Type Control
 (These bits are always 1; there is no alternate mode for these lines.)
 8-23 Not used

X: \$C085 General I/O Data Inversion Control

Bits: 0 I/O24 Inversion Control
 ...
 7 I/O31 Inversion Control
 (All bits: 0=Non-inverting; 1=Inverting)
 8-23 Not used

Y: \$C086 Data Type Control Register (Do not change.)

Bits: 0 DAT0 Data Type Control (0=ENCC9; 1 =DAT0)
 1 DAT1 Data Type Control (0=ENCC10; 1 =DAT1)
 2 DAT2 Data Type Control (0=Fault9; 1 =DAT2)
 3 DAT3 Data Type Control (0=Fault10; 1 =DAT3)
 4 DAT4 Data Type Control (0=EQU9; 1 =DAT4)
 5 DAT5 Data Type Control (0=EQU10; 1 =DAT5)
 6 DAT6 Data Type Control (0=AENA9; 1 =DAT6)
 7 DAT7 Data Type Control (0=AENA10; 1 =DAT7)
 8 SEL0 Data Type Control (0=ADC_STROB; 1=SEL0)
 9 SEL1 Data Type Control (0=ADC_CLK; 1=SEL1)
 10 SEL2 Data Type Control (0=ADC_A9; 1=SEL2)
 11 SEL3 Data Type Control (0=ADC_B9; 1=SEL3)
 12 SEL4 Data Type Control (0=ADC_A10; 1=SEL4)

	13	SEL5 Data Type Control (0=ADC_B10; 1=SEL5)
	14	SEL6 Data Type Control (0=SCLK; 1=SEL6)
	15	SEL7 Data Type Control (0=SCLK_DIR*; 1=SEL7) (All bits: 0=dedicated hardware I/O; 1=general I/O) (All bits must be 0 for use with ACC-1E 2-axis piggyback board)
	16-23	Not used
X: \$C086	JTHW Port Data Inversion Control Register (when used as general I/O, see Y:\$C086.)	
Bits:	0	DAT0 Inversion Control
	...	
	7	DAT7 Inversion Control
	8	SEL0 Inversion Control
	...	
	15	SEL7 Inversion Control (All bits: 0=Non-inverting; 1=Inverting) (All bits must be 0 to use standard port accessories)
	16-23	Not used
Y: \$C087	Data Type Control Register (Do not change.)	
Bits:	0	DISP0 Data Type Control
	...	
	7	DISP7 Data Type Control
	8	CTRL0 Data Type Control
	...	
	11	CTRL3 Data Type Control (These bits are always 1; there is no alternate mode for these pins.)
	12-23	Not used
X: \$C087	Data Inversion Control Register	
Bits:	0	DISP0 Inversion Control
	...	
	7	DISP7 Inversion Control
	8	CTRL0 Inversion Control
	9	CTRL1 Inversion Control
	10	CTRL2 Inversion Control (AENA_1 Relay)
	11	CTRL3 Inversion Control (AENA_2 Relay) (All bits: 0=Non-inverting; 1=Inverting) (All bits must be 0 to use standard port accessories.)
	12-23	Not used
Y: \$C088-\$C08B	Not used	
X: \$C088-\$C08B	Not used	
Y: \$C08C	Pure binary conversion from gray code input on I/O00 to I/O23	

Note:

The pins associated with this register are used for other purposes on the ACC-68M Device.

X: \$C08C	DAC Strobe Word, 24 bits (Shifted out MSB first, one bit per DACCLK cycle, starting on rising edge of phase clock)	
Y: \$C08D	Gray-to-binary conversion bit-length control	

Note:

The pins associated with this register are used for other purposes on the ACC-68M Device.

Bits:	0-3	Bit length of less significant word portion (I/O00 - I/Onn)
	4	=1 specifies 16-bit lower/8-bit upper conversion
	5-23	Not used
X:	§C08D	Not used
Y:	§C08E	MACRO Node Enable Control (I996)
Bits:	0	Node 0 enable control
	...	
	15	Node 15 enable control
		(0=node disable; 1=node enable)
	16-19	Sync packet slave node number control
	20-23	Master number control
X:	§C08E	Not used
Y:	§C08F	MACRO Ring Status and Control
Bits:	0	Data overrun error (cleared when read)
	1	Byte violation error (cleared when read)
	2	Packet parity error (cleared when read)
	3	Data under run error (cleared when read)
	4	Master station enable
	5	Synchronizing master station enable
	6	Sync packet received (cleared when read)
	7	Sync packet phase lock enable
	8	Node 8 master address check disable
	9	Node 9 master address check disable
	10	Node 10 master address check disable
	11	Node 11 master address check disable
	12	Node 12 master address check disable
	13	Node 13 master address check disable
	14	Node 14 master address check disable
	15	Node 15 master address check disable
X:	§C08F	DSPGATE2 clock control register
Bits:		(Bits 0-11 comprise I993)
	0-2:	SCLK Frequency Control n ($f=39.3216\text{MHz} / 2^n$, $n=0-7$)
	3-5:	PFM Clock Frequency Control n ($f=39.3216\text{MHz} / 2^n$, $n=0-7$)
	6-8:	DAC Clock Frequency Control n ($f=39.3216\text{MHz} / 2^n$, $n=0-7$)
	9-11:	ADC Clock Frequency Control n ($f=39.3216\text{MHz} / 2^n$, $n=0-7$)
	12:	Phase Clock Direction (0=output, 1=input)
		(This must be 1)
	13:	Servo Clock Direction (0=output, 1=input)
		(This must be 1)
	14-15:	Not used (report as zero)
	16-19:	Phase Clock Frequency Control n (I997)
		($f=\text{MAXPHASE} / [n+1]$, $n=0-15$)
	20-23:	Servo Clock Frequency Control n
		($f=\text{PHASE} / [n+1]$, $n=0-15$)

DSPGATE2 Channel 1* and Channel 2*.

These are the Auxiliary channels that support the ACC-65E ADC's, DAC's, and AENA's

Chan #	1*	2*
Hex	[\$C090]	[\$C098]

Y: \$C09x Channel n Time between last two encoder counts (SCLK cycles)

X: \$C09x Channel n Status Word

- Bits:
- 0-2 Captured Hall Effect Device (UVW) State
 - 3 Invalid demultiplex of C, U, V, and W
 - 4-7 Not used (reports as 0)
 - 8 Encoder Count Error (0 on counter reset, 1 on illegal transition)
 - 9 Position Compare (EQU_n) output value
 - 10 Position-Captured-On-Gated-Index Flag
(=0 on read of captured position register, =1 on trigger capture)
 - 11 Position-Captured Flag (on any trigger)
(=0 on read of captured position register, =1 on trigger capture)
 - 12 Handwheel 1 Channel A (HWAn) Input Value
 - 13 Handwheel 1 Channel B (HWB_n) Input Value
 - 14 Handwheel 1 Channel C (Index, HWC_n) Input Value (ungated)
 - 15 Amplifier Fault (FAUL_{Tn}) Input Value
 - 16 Home Flag (HMFL_n) Input Value
 - 17 Positive End Limit (PLIM_n) Input Value
 - 18 Negative End Limit (MLIM_n) Input Value
 - 19 User Flag (USER_n) Input Value
 - 20 FlagW_n Input Value
 - 21 FlagV_n Input Value
 - 22 FlagUn Input Value
 - 23 FlagT_n Input Value

Chan #	1*	2*
Hex	[\$C091]	[\$C099]

Y: \$C09x Channel n Encoder Time Since Last Encoder Count (SCLK cycles)

X: \$C09x Channel n Encoder Phase Position Capture Register (counts)

Chan #	1*	2*
Hex	[\$C092]	[\$C09A]

Y: \$C09x Channel n Output A Command Value

- Bits:
- 8-23: PWM Command Value
 - 6-23: Serial DAC Command Value
 - 0-5: Not Used

X: \$C09x Channel n Encoder Servo Position Capture Register

- Bits:
- 0: Direction of last count (0=up, 1=down)
 - 1-23: Position counter (units of counts)

Chan #	1*	2*
Hex	[\$C093]	[\$C09B]

Y: \$C09x Channel n Output B Command Value

- Bits: 8-23: PWM Command Value

6-23: Serial DAC Command Value

0-5: Not used

X:\$C09x Channel n Flag Position Capture Value; 24 bits, units of counts

Chan #	1*	2*
Hex	[\$C094]	[\$C09C]

Y:\$C09x Channel n Output C Command Value

Bits: 8-23: PWM Command Value

0-23: PFM Command Value

X:\$C094 Channel 9-10 ADC Strobe Word, 24 bits
(Shifted out MSB first one bit per DAC_CLK cycle, starting on rising edge of phase clock)

X:\$C09C Channel 9-10 PWM, PFM, MaxPhase Control Word

Bits: 0-7: PWM Dead Time (16*PWM CLK cycles)
also PFM pulse width (PFM CLK cycles)

8-23: PWM Max Count Value

PWM Frequency = 117.96MHz / [10(MaxCount+1)]

"MaxPhase" Frequency = 2*PWM* Frequency

Chan #	1*	2*
Hex	[\$C095]	[\$C09D]

Y:\$C09x Supplementary Channel n* ADC A Input Value

Bits: 6-23: Serial ADC Value

0-5: Not used

X:\$C09x Channel n Control Word

Bits 0-1: Encoder Decode Control
00: Pulse and direction decode
01: x1 quadrature decode
10: x2 quadrature decode
11: x4 quadrature decode

2-3: Direction & Timer Control
00: Standard timer control, external signal source, no inversion
01: Standard timer control, external signal source, invert direction
10: Standard timer control, internal PFM source, no inversion
11: Alternate timer control, external signal source

4-5: Position Capture Control
00: Software capture (by setting bit 6)
01: Use encoder index alone
10: Use capture flag alone
11: Use encoder index and capture flag

6: Index Capture Invert Control (0=no inversion, 1=inversion)

7: Flag Capture Invert Control (0=no inversion, 1=inversion)

8-9: Capture Flag Select Control
00: Home Flag (HMFLn)
01: Positive Limit (PLIMn)
10: Negative Limit (MLIMn)

- 11: User Flag (USERn)
- 10: Encoder Counter Reset Control (1=reset)
- 11: Position Compare Initial State Write Enable
- 12: Position Compare Initial State Value
- 13: Position Compare Channel Select
(0= use this channel's encoder; 1=use first encoder on IC)
- 14: AENAn output value
- 15: Gated Index Select for Position Capture
(0=ungated index, 1=gated index)
- 16: Invert AB for Gated Index
(0: Gated Signal=A&B&C; 1: Gated Signal=A/&B/&C)
- 17: Index channel demultiplex control (0=no demux, 1=demux)
- 18: Reserved for future use (reports as 0)
- 19: Invert PFM Direction Control (0=no inversion, 1=invert)
- 20: Invert A & B Output Control (0=no inversion, 1=invert)
- 21: Invert C Output Control (0=no inversion, 1=invert)
- 22: Output A & B Mode Select (0=PWM, 1=DAC)
- 23: Output C Mode Select (0=PWM, 1=PFM)

Chan #	1*	2*
Hex	[\$C096]	[\$C09E]

Y: \$C09x Supplementary Channel n* ADC B Input Value (uses SEL3 in dedicated mode)

Bits: 6-23: Serial ADC Value
 0-5: Not used

X: \$C09x Channel n Encoder Compare Auto-increment value (24 bits, units of counts)

Chan #	1*	2*
Hex	[\$C097]	[\$C09F]

Y: \$C09x Channel n Encoder Compare A Value (24 bits, units of counts)

X: \$C09x Channel n Encoder Compare B Value (24 bits, units of counts)