HARDWARE REFERENCE MANUAL

Accessory 14E

OPTO 48-Bit Input/Output Board

3A□-603474-□U□□

February 2, 2022

Document # MN-000210

Copyright Information

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 your local Omron representative

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. Before powering, please ensure there is no visible damage to the product.

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. Our products should not be placed in locations that can accrue a lot of dust, salt, or conductive iron-like powder. 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. For your own safety, please keep the product's environmental conditions within the range outlined by the Environment Specifications section that can be located from the table of contents in this manual.

Trademarks

Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

EN Dispose in accordance with applicable regulations.

REVISION HISTORY				
REV.	**DESCRIPTION**	**DATE**	**CHG**	**APPVD**
1	Added CE declaration	06/07/06	CP	S. FIERRO
2	Added opto22 note & section; updated e5 settings; fixed latch & greycode control word settings; corrected inversion control example; added absolute power-on position section	10/23/07	CP	S. FIERRO
3	Added UL seal to manual cover updated agency approval/safety section	10/01/09	CP	S.FIERRO
4	Added DSUB connector pinout details	02/09/16	MA	S. MILICI
5	Added KC conformity	10/17/18	SM	RN
6	Added explanation of sampling at servo on power PMAC	3/11/20	AA	RN
7	Added mounting and installation section	12/09/20	SM	RN
A	Added UKCA Marking to front cover and added description in Agency of Approval section	09/01/21	AE	SM
B	Updated UKCA standard	02/02/22	AE	SF

Table of Contents

INTRODUCTION

The PMAC Accessory 14E is a general-purpose input/output board to the UMAC-Turbo or UMAC-MACRO systems. This accessory provides expanded and flexible digital I/O capabilities for the controller. It may be configured for a wide variety of different uses to serve many diverse applications. It is commonly used for discrete I/O and for parallel binary feedback or greyscale code (absolute encoders and laser inferometers). This accessory also allows easy connection to OPTO22 and Grayhill Module I/O cards. It provides UMAC with 48-bits of digital I/O which may be configured according to specific needs.

Other UMAC IO Accessories:

ACC-9E	48 optically isolated inputs
ACC-10E	48 optically isolated outputs, low power
ACC-11E	24 inputs and 24 outputs, low power, all optically isolated
ACC-12E	24 inputs and 24 outputs, high power, all optically isolated
ACC-14E	48-bits TTL level I/O
ACC-65E	Sourcing 24 inputs and 24 outputs, self protecting, all optically isolated, 250mA outputs
ACC-66E	48 optically isolated, self protecting, sourcing inputs
ACC-67E	48 optically isolated, self protecting, sourcing outputs, 250mA outputs
ACC-68E	Sinking 24 inputs and 24 outputs, self protecting, all optically isolated

SPECIFICATIONS

Environmental Specifications

Description	Specification	Notes
Operating Temperature	0°C to 45°C,	
Storage Temperature	-25°C to 70°C	
Humidity	10% to 95 % non-condensing	

Physical Specifications

Description	Specification	Notes
Dimensions	Length: 16.256 cm (6.4 in.) Height: 10 cm (3.94 in.) Width: 2.03 cm (0.8 in.)	
Weight	170 g	Front plate included
IDC Connectors	50-pin box header	Thermoplastic, UL-94V0
The width is the width of the front plate. The length and height are the dimensions of the PCB.		

Electrical Specifications

Description	Specification	Notes
ACC-14E Power Requirements	5V @ 0.15 A (±10%)	

Agency Approval and Safety

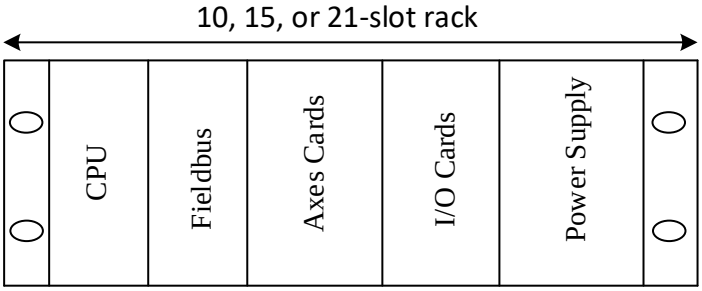
Item	Description
CE Mark	EN61326-1
EMC	EN55011 Class A Group 1 EN61000-4-2 EN61000-4-3 EN61000-4-4 EN61000-4-5 EN61000-4-6
UL	UL 61010-1 File E314517
cUL	CAN/CSA C22.2 No. 1010.1-92 File E314517
Flammability Class	UL 94V-0
KC	EMI: KN 11 EMS: KN 61000-6-2
UKCA	2016 No. 1091

사 용 자 안 내 문
이 기기는 업무용 환경에서 사용할 목적으로 적합성평가를 받은 기기로서 가정용 환경에서 사용하는 경우 전파간섭의 우려가 있습니다.

한국 EMC적용제품 준수사항
본 제품은 전파법(KC 규정)을 준수합니다. 제품을 사용하려면 다음 사항에 유의하십시오. 이 기기는 업무용 환경에서 사용할 목적으로 적합성평가를 받은 기기로서 가정용 환경에서 사용하는 경우 전파간섭의 우려가 있습니다.

MOUNTING AND INSTALLATION

To connect a UMAC accessory, simply slide the board into any open slot of the UMAC rack. Customarily, accessories are installed from left to right as follows:



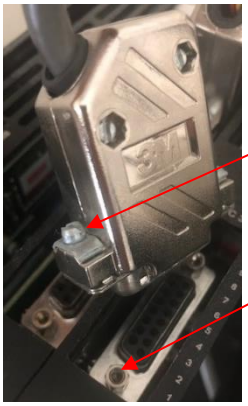
Prior to installation, make sure that you have set the jumpers and address settings to your desired requirements. Use the guide tracks that have been installed in the empty slots of your UMAC system when installing a board.

As you slide the board into the rack, use caution to ensure none of the components on the board make contact with the front plates of the boards on either side. Getting the front plate flush with the front of the rack and turning the front screws firmly will ensure a good connection with the backplane.

When removing a board from the system, the user must first pull out any wired connections from the top, bottom, and front panels then loosen the pem-nuts on the front of the rack. Next, the user can gently pull the board from the rack and use caution to ensure that none of the components on the board make contact with the boards on either side.

Caution

This product contains D-SUB style connectors. Do not overtighten any screws on mating connectors, and when possible, only tighten by hand. Overtightening, such as with manual or electric tools, may lead to the mating nut detaching when subsequently unscrewed, which may result in damage to the product, other electronics, and/or injury.



HARDWARE SETUP

The ACC-14E uses expansion port memory locations defined by the type of PMAC (3U Turbo or MACRO Station) it is communicating to directly. Typically, these memory locations are used with other 3U I/O accessories such as:

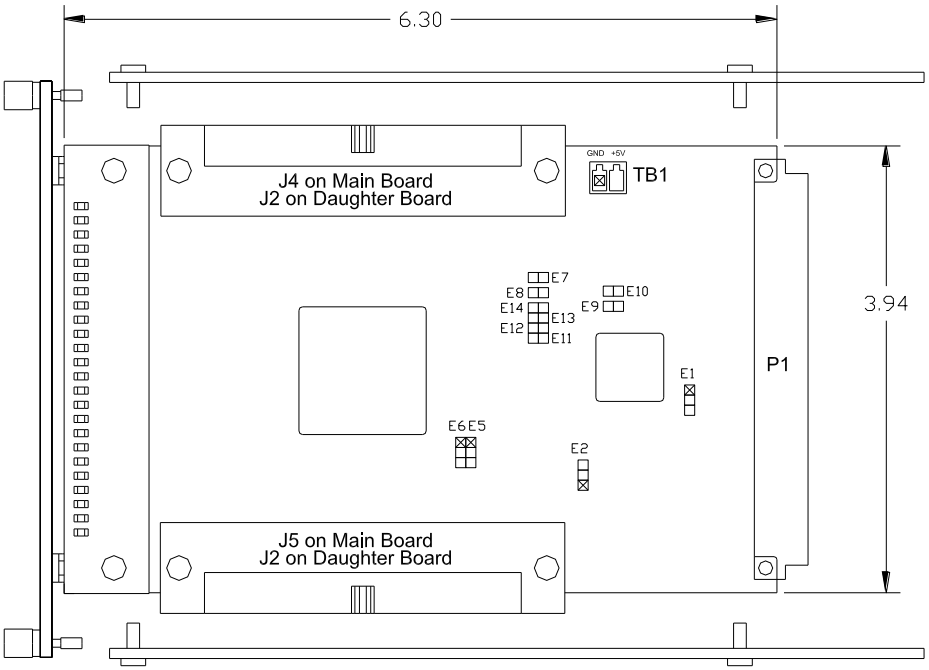
ACC-9E, ACC-10E, ACC-11E ACC-12E
ACC-65E, ACC66E, ACC-67E, ACC-68E
ACC-28E 16-bit A/D Converter Inputs (up to four per card)
ACC-53E SSI Encoder Inputs

All of these accessories have settings, which tell them where the information is to be processed at either the PMAC 3U Turbo or the MACRO Station.

3U Turbo PMAC Memory Locations	**MACRO Station Memory Locations**
$078C00, $079C00 $07AC00, $07BC00	$8800,$9800 $A800,$B800
$078D00, $079D00 $07AD00, $07BD00	$8840,$9840 $A840,$B840
$078E00, $079E00 $07AE00, $07EC00	$8880,$9880 $A880,$B880
$078F00, $079F00 $07AF00, $07BF00	$88C0,$98C0 $A8C0,$B8C0

The ACC-14E has a set of dipswitches telling it where to process its data. Once the information is at these locations, the binary word can be processed in the encoder conversion table to use for servo loop closure. Proper setting of the dipswitches ensures all of the U bus I/O boards used in the application do not interfere with each other.

Layout Diagram



Address Select DIP Switch S1

The switch one (S1) settings allows the selection of the starting address location for the first I/O gate on the ACC-14E. The following two tables show the dipswitch settings for both the Turbo PMAC 3U and the MACRO Station.

Turbo PMAC 3U Switch Settings

Chip Select	3U Turbo PMAC Address	Dipswitch SW1 Position					
		6	5	4	3	2	1
CS10	Y:$78C00-03	Close	Close	Close	Close	Close	Close
	Y:$79C00-03	Close	Close	Close	Open	Close	Close
	Y:$7AC00-03	Close	Close	Open	Close	Close	Close
	Y:$7BC00-03	Close	Close	Open	Open	Close	Close
CS12	Y:$78D00-03	Close	Close	Close	Close	Close	Open
	Y:$79D00-03	Close	Close	**Close**	Open	Close	Open
	Y:$7AD00-03	Close	Close	Open	Close	Close	Open
	Y:$7BD00-03	Close	Close	Open	Open	Close	Open
CS14	Y:$78E00-03	Close	Close	Close	Close	Open	Close
	Y:$79E00-03	Close	Close	Close	Open	Open	Close
	Y:$7AE00-03	Close	Close	Open	Close	Open	Close
	Y:$7BE00-03	Close	Close	Open	Open	Open	Close

Chip Select	3U Turbo PMAC Address	Dipswitch SW1 Position					
		6	5	4	3	2	1
CS16	Y:$78F00-03	Close	Close	Close	Close	Open	Open
	Y:$79F00-03	Close	Close	Close	Open	Open	Open
	Y:$7AF00-03	Close	Close	Open	Close	Open	Open
	Y:$7BF00-03	Close	Close	Open	Open	Open	Open

MACRO Station Switch Settings

Chip Select	3U Turbo PMAC Address	Dipswitch SW1 Position					
		6	5	4	3	2	1
CS10	Y:$8800	Close	Close	Close	Close	Close	Close
	Y:$9800	Close	Close	Close	Open	Close	Close
	Y:$A800	Close	Close	Open	Close	Close	Close
	Y:$B800 ($FFE0*)	Close	Close	Open	Open	Close	Close
CS12	Y:$8840	Close	Close	Close	Close	Close	Open
	Y:$9840	Close	Close	Close	Open	Close	Open
	Y:$A840	Close	Close	Open	Close	Close	Open
	Y:$B840 ($FFE8*)	Close	Close	Open	Open	Close	Open
CS14	Y:$8880	Close	Close	Close	Close	Open	Close
	Y:$9880	Close	Close	**Close**	Open	Open	Close
	Y:$A880	Close	Close	Open	Close	Open	Close
	Y:$B880 ($FFF0*)	Close	Close	Open	Open	Open	Close
CS16	Y:$88C0	Close	Close	Close	Close	Open	Open
	Y:$98C0	Close	Close	Close	Open	Open	Open
	Y:$A8C0	Close	Close	Open	Close	Open	Open
	Y:$B8C0	Close	Close	Open	Open	Open	Open
The default setting is all closed position.							

JUMPERS

Refer to the layout diagram of ACC-14E for the location of the jumpers on the board.

E-Point Jumpers

Jumper	Config	Description	Settings	Default
E1	1-2	Turbo-PMAC/MACRO select	Jump 1-2 for Turbo 3U CPU and MACRO CPU * Jump 2-3 for legacy MACRO CPU (before 6/00)	1-2
E2	1-2	Sample clock select	1-2 servo clock is sample clock in Turbo 2-3 phase clock is sample clock ** Remove to use servo clock as sample clock in Power	1-2
E5	1-2-3	Output clock polarity	1-2 OCLK is sample clock- 2-3 OCLK is sample clock+ Remove for OPTO22 use	1-2
E6	1-2-3	Latch clock polarity	1-2 ENA_CLK is sample clock+ 2-3 ENA_CLK is sample clock-	1-2
E7	1-2	Input latch signal for bits 0-23 (port A)	ON passes the ICLK1 5V input from J4 to latch port A I/O OFF passes the ICLK1 0V input from J4 to latch port A I/O	1-2
E8	1-2	Error latch signal for bits 0-23 (port A)	ON allows a low ERR1/ input to disable the port A Latch OFF allows a high ERR1/ input to disable the port A Latch	1-2
E9	1-2	Input latch signal for bits 24-47	ON passes the ICLK2 5Vinput from J5 to latch port B I/O OFF passes the ICLK2 0Vinput from to latch port B I/O	1-2
E10	1-2	Error latch signal for bits 24-47	ON allows a low ERR2/ input to disable the port B latch OFF allows a high ERR2/ input to disable the port B latch	1-2
E11-E14	1-2	Not used	No jumper	No Jumper

*For legacy MACRO Stations (part number 602804-100 thru 602804-104)
**In Power PMAC, users must remove this jumper and set Cid[j].Dir to 1 to use servo clock as sample clock
Phase+ and Phase- are designated on the ACC-14E schematic for OCLK and ENA_CLK even though they are selected by E2 to either be the servo clock or phase clock.
The E7, E8, E9, and E10 jumpers are pulled up to 5V if not using ICLK or ERR inputs from the encoder.

Using with OPTO22 Compatible Devices

The Acc-14E is designed to work directly with OPTO22/Grayhill IO module devices. The 50-pin J4 and J5 IDC header connectors are compatible with the 50-pin headers on the OPTO22 mounting racks.

However, the user must remove the E5 jumper to disable the OCLK output on pin 48. The OCLK is used for binary encoder synchronization and is not needed on OPTO22 racks.

Note

Failure to remove E5 when using OPTO22 racks could result in incorrect data reads.

Hardware Address Limitations

Some of the older UMAC I/O accessories might create a hardware address limitation relative to the newer series of UMAC high-speed I/O cards. The ACC-14E would be considered a newer high speed I/O card. The new I/O cards have four addresses per chip select (CS10, CS12, CS14, and CS16). This enables these cards to have up to 16 different addresses. The ACC-9E, ACC-10E, ACC-11E, and ACC-12E all have one address per chip select but also have the low-byte, middle-byte, and high-byte type of addressing scheme and allow for a maximum of twelve of these I/O cards.

UMAC Card Types

UMAC Card	Number of Addresses	Category	Maximum # of cards	Card Type
ACC-9E , ACC-10E ACC-11E, ACC-12E	4	General I/O	12	A
ACC-65E, ACC-66E ACC-67E, ACC-68E ACC-14E	16	General I/O	16	B
ACC-28E, ACC-36E ACC-59E	16	ADC and DAC	16	B
ACC-53E, ACC-57E ACC-58E	16	Feedback devices	16	B

Chip Select Addresses

Chip Select	UMAC Turbo Type A Card	MACRO Type A Card	UMAC Turbo Type B Card	MACRO Type B Card
10	$078C00	$FFE0 or $8800	$078C00, $079C00 $07AC00, $07BC00	$8800,$9800 $A800,$B800
12	$078D00	$FFE8 or $8840	$078D00, $079D00 $07AD00, $07BD00	$8840,$9840 $A840,$B840
14	$078E00	$FFF0 or $8880	$078E00, $079E00 $07AE00, $07EC00	$8880,$9880 $A880,$B880
16	$078F00	$88C0	$078F00, $079F00 $07AF00, $07BF00	$88C0,$98C0 $A8C0,$B8C0

Addressing Conflicts

When using only the type A UMAC cards or the type B UMAC cards in an application, do not worry about potential addressing conflicts other than making sure the individual cards are set to the addresses as specified in the manual.

If using both type A and type B UMAC cards in the rack, be aware of the possible addressing conflicts. If using the type A card on a particular chip select (CS10, CS12, CS14, or CS16) then do not use a type B card with the same chip select address unless the type B card is a general I/O type. If the type B card is a general I/O type, then the type B card will be the low-byte card at the chip select address and the type A card(s) will be setup at as the middle-byte and high-byte addresses.

Type A and Type B Example 1: ACC-11E and ACC-36E

If using an ACC-11E and ACC-36E both cards cannot use the same chip select because the data from both cards will be overwritten by the other card.

Make sure to not address both cards to the same chip select.

Type A and Type B Example 2: ACC-11E and ACC-14E

For this example, the two cards are allowed to share the same chip select because the ACC-14E is a general purpose I/O type B card. The only restriction in doing so is that the ACC-14E must be considered the low-byte addressed card and the ACC-11E must be jumpered to either the middle or high bytes (jumper E6A-E6H).

USING ACC-14E WITH UMAC TURBO

For the UMAC-Turbo, the ACC-14E can be used for either general purpose I/O or as latched inputs for servo loop position or velocity feedback. The registers used for general I/O use are 8-bit registers three 8-bit registers are defined for each 24-bit I/O port. To use the ACC-14E for closed loop servo data, set up various I-variables for the encoder conversion table and power-on position. The encoder conversion table is set up using variables I8000 through I8192. Each variable is an entry in the conversion table and its setup is described in the Turbo PMAC Software Reference Manual.

UMAC-Turbo Memory Mapping for ACC-14E

The I/O Gate used on the ACC-14E is an 8-bit processor and therefore the memory mapping to the I/O bits is processed as 8-bit words at the Turbo UMAC. Using this simple scheme, up to 768 (48×16) bits of data can be processed for general purpose I/O.

	CS10 **SW1-1 ON** **SW1-2 ON**	**CS12** **SW1-1 OFF** **SW1-2 ON**	**CS14** **SW1-1 ON** **SW1-2 OFF**	**CS16** **SW1-1 OFF** **SW1-2 OFF**	**Description**
SW1-3 ON SW1-4 ON	Y:$078C00,0,8	Y:$078D00,0,8	Y:$078E00,0,8	Y:$078F00,0,8	**I/O bits 0-7**
	Y:$078C01,0,8	Y:$078D01,0,8	Y:$078E01,0,8	Y:$078F01,0,8	I/O bits 8-15
	Y:$078C02,0,8	Y:$078D02,0,8	Y:$078E02,0,8	Y:$078F02,0,8	I/O bits 16-23
	Y:$078C03,0,8	Y:$078D03,0,8	Y:$078E03,0,8	Y:$078F03,0,8	I/O bits 24-31
	Y:$078C04,0,8	Y:$078D04,0,8	Y:$078E04,0,8	Y:$078F04,0,8	I/O bits 32-39
	Y:$078C05,0,8	Y:$078D05,0,8	Y:$078E05,0,8	Y:$078F05,0,8	I/O bits 40-47
	Y:$078C07,0,8	Y:$078D07,0,8	Y:$078E07,0,8	Y:$078F07,0,8	Control Word
SW1-3 OFF SW1-4 ON	Y:$079C00,0,8	Y:$079D00,0,8	Y:$079E00,0,8	Y:$079F00,0,8	I/O bits 0-7
	Y:$079C01,0,8	Y:$079D01,0,8	Y:$079E01,0,8	Y:$079F01,0,8	I/O bits 8-15
	Y:$079C02,0,8	Y:$079D02,0,8	Y:$079E02,0,8	Y:$079F02,0,8	I/O bits 16-23
	Y:$079C03,0,8	Y:$079D03,0,8	Y:$079E03,0,8	Y:$079F03,0,8	I/O bits 24-31
	Y:$079C04,0,8	Y:$079D04,0,8	Y:$079E04,0,8	Y:$079F04,0,8	I/O bits 32-39
	Y:$079C05,0,8	Y:$079D05,0,8	Y:$079E05,0,8	Y:$079F05,0,8	I/O bits 40-47
	Y:$079C07,0,8	Y:$079D07,0,8	Y:$079E07,0,8	Y:$079F07,0,8	Control Word
SW1-3 ON SW1-4 OFF	Y:$07AC00,0,8	Y:$07AD00,0,8	Y:$07AE00,0,8	Y:$07AF00,0,8	I/O bits 0-7
	Y:$07AC01,0,8	Y:$07AD01,0,8	Y:$07AE01,0,8	Y:$07AF01,0,8	I/O bits 8-15
	Y:$07AC02,0,8	Y:$07AD02,0,8	Y:$07AE02,0,8	Y:$07AF02,0,8	I/O bits 16-23
	Y:$07AC03,0,8	Y:$07AD03,0,8	Y:$07AE03,0,8	Y:$07AF03,0,8	I/O bits 24-31
	Y:$07AC04,0,8	Y:$07AD04,0,8	Y:$07AE04,0,8	Y:$07AF04,0,8	I/O bits 32-39
	Y:$07AC05,0,8	Y:$07AD05,0,8	Y:$07AE05,0,8	Y:$07AF05,0,8	I/O bits 40-47
	Y:$07AC07,0,8	Y:$07AD07,0,8	Y:$07AE07,0,8	Y:$07AF07,0,8	Control Word
SW1-3 OFF SW1-4 OFF	Y:$07BC00,0,8	Y:$07BD00,0,8	Y:$07BE00,0,8	Y:$07BF00,0,8	**I/O bits 0-7**
	Y:$07BC01,0,8	Y:$07BD01,0,8	Y:$07BE01,0,8	Y:$07BF01,0,8	I/O bits 8-15
	Y:$07BC02,0,8	Y:$07BD02,0,8	**Y:$07BE02,0,8**	Y:$07BF02,0,8	I/O bits 16-23
	Y:$07BC03,0,8	Y:$07BD03,0,8	**Y:$07BE03,0,8**	Y:$07BF03,0,8	I/O bits 24-31
	Y:$07BC04,0,8	Y:$07BD04,0,8	Y:$07BE04,0,8	Y:$07BF04,0,8	I/O bits 32-39

	CS10 SW1-1 ON SW1-2 ON	CS12 SW1-1 OFF SW1-2 ON	CS14 SW1-1 ON SW1-2 OFF	CS16 SW1-1 OFF SW1-2 OFF	Description
	Y:$07BC05,0,8	Y:$07BD05,0,8	Y:$07BE05,0,8	Y:$07BF05,0,8	I/O bits 40-47
	Y:$078C07,0,8	Y:$078D07,0,8	Y:$078E07,0,8	Y:$078F07,0,8	Control Word
Note: SW1-5 and SW1-6 must be set to ON.					

Control Register

The control register at address {Base + 7} permits the configuration of the IOGATE IC to a variety of applications. The control register consists of eight write/read-back bits – Bits 0 - 7. The control register consists of two sections: Direction Control and Register Select.

The direction control the setting of input bytes to be read only. One of the advantages of the IOGATE IC is the ability to define the bits as inputs or outputs. This control mechanism ensures that the inputs will be always read properly. The traditional I/O accessories always define the inputs and outputs by hardware.

The register select bits define the input or output bytes, inversion control, or the latching input features.

Direction Control Bits

Bits 0 to 5 of the control register simply control the direction of the I/O for the matching numbered data register. That is, Bit *n* controls the direction of the I/O at {Base + *n*}. A value of 0 in the control bit (the default) permits a write operation to the data register, enabling the output function for each line in the register. Enabling the output function does not prevent the use of any or all of the lines as inputs, as long as the outputs are off (non-conducting). A value of 1 in the control bit does not permit a write operation to the data register, disabling the output, reserving the register for inputs.

For example, a value of 1 in Bit 3 disables the write function into the data register at address {Base + 3}, ensuring that lines IO24 - IO31 can be used always as inputs.

Register Select Control Bits

The control register also allows access to each of the setup registers for each of the 8-bit IO words. Bits 6 and 7 of the control register together select which of four possible registers can be accessed at each of the addresses {Base + 0} through {Base + 5}. They also select which of two possible registers can be selected at {Base + 6}. Only access the setup registers when setting the 8-bit I/O words for:

- Individual inversion control for each I/O point (bit 7 = 0, bit 6 = 1)
- Individual control on inputs of latched or transparent read (bit 7 = 1, bit 6 = 0)
- Automatic conversion of latched Gray code inputs (bit 7 = 1, bit 6 = 1)
- 8 additional inputs for latch control, maskable interrupts, or general inputs

The following table explains how these bits select registers:

Bit 7	Bit 6	Combined Value	{Base + 0} to {Base + 5} Register Selected	{Base + 6} Register Selected
0	0	0	Data Register	Data Register
0	1	1	Setup Register 1	Setup Register
1	0	2	Setup Register 2	Not applicable
1	1	3	Setup Register 3	Not applicable

In a typical application, non-zero combined values of Bits 6 and 7 are used only for initial configuration of the IC. These values are used to access the setup registers at the other addresses. After the configuration is finished, zeros are written to both Bits 6 and 7, so the data registers at the other registers can be accessed.

Setup Registers

There are four registers accessible at each of the IC addresses {Base + 0} to {Base + 5}: three 8-bit setup registers and an 8-bit data register. The setup registers control how data is written to and read from the data registers.

Setup Register 1: Inversion Control

Setup Register 1 at each address {Base + 0} through {Base + 5}, which is selected by writing a 1 to Bit 6 of the Control Word at {Base + 7} and a 0 to Bit 7, which is the inversion control register for the Data Register at the same address. Each bit of Setup Register 1 controls the inversion of the matching bit of the Data Register at the same address.

A value of 0 in a bit of Setup Register 1 specifies an inverting I/O point for the matching bit of the Data Register at the same address. That is, for an output, a value of 0 produces a low (conducting) output, and a value of 1 produces a high (non-conducting) output. For an input, a line pulled low produces a 1 value, and a line pulled high or permitted to float high produces a 0 value.

A value of 1 in a bit of Setup Register 1 specifies a non-inverting I/O point for the matching bit of the Data Register at the same address. That is, for an output, a value of 0 produces a high (non-conducting) output, and a value of 1 produces a low (conducting) output. For an input, a line pulled low produces a 0 value, and a line pulled high or permitted to float high produces a 1 value.

Setup Register 2: Read Control

Setup Register 2 at each address {Base + 0} through {Base + 5}, which is selected by writing a 0 to Bit 6 of the Control Word at {Base + 7} and a 1 to Bit 7 (the read control register for the Data Register at the same address). Each bit of Setup Register 2 controls what data is read from the matching bit of the Data Register at the same address.

The action of a bit of Setup Register 2 is dependent on the setting of the matching bit of Setup Register 3 for the same address. If the matching bit of Setup Register 3 is 0 (selecting unlatched inputs) the bit of Setup Register 2 controls whether the pin value is read, or the value in the writeable register is read. A value of 0 in the bit of Setup Register 2 selects the pin value to be read from the matching bit of the Data Register at the same address; a value of 1 in the bit selects the writeable register value.

If the matching bit of Setup Register 3 is 1 (selecting latched inputs) the bit of Setup Register 2 controls whether the directly latched data is read, or the value that is the result of a Gray-code-to-binary conversion. A value of 0 in the bit of Setup Register 2 selects the directly latched value to be read from the matching bit of the Data Register at the same address; a value of 1 in the bit selects the value that is the result of a Gray-code-to-binary conversion.

Setup Register 3: Latch Control

Setup Register 3 at each address {Base + 0} through {Base + 5}, which is selected by writing a 1 to Bit 6 of the Control Word at {Base + 7} and a 1 to Bit 7 (the latch control register for the Data Register at the same address). Each bit of Setup Register 3 controls whether latched or unlatched data is read from the matching bit of the Data Register at the same address.

A value of 0 in the bit of Setup Register 3 selects unlatched data to be read from the matching bit of the Data Register at the same address; a value of 1 in the bit selects latched data to be read.

For both the latched and unlatched settings, the matching bit of Setup Register 2 controls exactly what type of data is read from the Data Register.

Data Registers

The Data Register at each address {Base + 0} through {Base + 5}, which is selected by writing a 0 to Bit 6 of the Control Register at {Base + 7} and a 0 to Bit 7, provides the working interface for the eight input/output lines matched to that address. The processor reads from or writes to the data register to access the input/output lines.

If there is a value of 1 in Bit *n* (*n* = 0 to 5) of the Control Word, a write operation to the Data Register at address {Base + *n*} has no effect on the I/O line, effectively disabling the output function for all eight lines associated with the register.

A read operation from a Data Register can access one of four types of data for each I/O line associated with the register (individually selectable), depending on how Setup Registers 2 and 3 at the same address have been configured.

The following table summarizes how the Setup Register bits control what data is read in the matching bit of the Data Register:

Setup Register 3 Bit Value	Setup Register 2 Bit Value	Data Type Read
0	0	Pin Value Read
0	1	Writeable Register Read
1	0	Converted Gray Code Read
1	1	Latched Input Read

Inversion Control Example:

```
M2007->Y:$078C07,0,8   ;control word for bits 0-47
M2000->Y:$078C00,0,8   ;I/O bits 0-7
M2001->Y:$078C01,0,8   ;I/O bits 8-15
M2002->Y:$078C02,0,8   ;I/O bits 16-23

M2007=$40              ;access inversion control (Setup Register 1)
M2000=$FF              ;non-inverting logic for bits 0-7
M2001=$FF              ;non-inverting logic for bits 8-15
M2002=$00              ;inverting logic for bits 16-23 (default)
```

The user will have to also write to the inputs before using them because their logic has been switched and the Gate will treat these as outputs until we force them to be inputs.

```
M2007=$00              ;Set to read/write to Data Port
M2000=$FF              ;Set for inputs in non-inverted mode
M2001=$FF              ;Set for inputs in non-inverted mode
M2002=$00              ;Set for inputs in normal inverted mode (default)

M2007=$07              ;Set bits 0-23 to be for inputs only
```

Latch-Gray and Latch-Binary Scale Example:

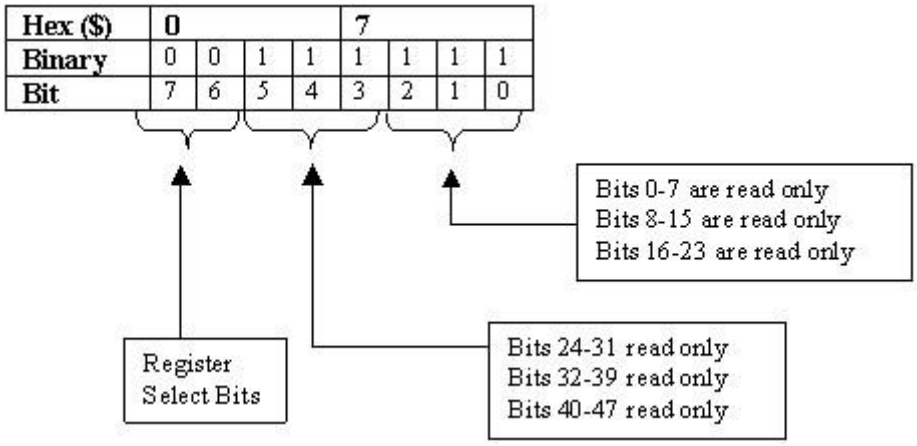
```
M2007->Y:$078C07,0,8  ;control word for bits 0-47
M2000->Y:$078C00,0,8  ;I/O bits 0-7
M2001->Y:$078C01,0,8  ;I/O bits 8-15
M2002->Y:$078C02,0,8  ;I/O bits 16-23
M2003->Y:$078C03,0,8  ;I/O bits 24-31
M2004->Y:$078C04,0,8  ;I/O bits 32-39
M2005->Y:$078C05,0,8  ;I/O bits 40-47

M2007=$C0             ;access Read control (Setup Register 3)
M2000=$FF             ;Latch bits 0-7
M2001=$FF             ;Latch 8-15
M2002=$FF             ;Latch 16-23
```

Control Word Setup Example

Setup the control words for the I/O card at power up. A simple PLC to setup the control word properly could accomplish this task. For this example, three ACC-14Es will be set up.

Control Word for ACC-14E**: (M2007->Y:$078C07,0,8)**




```
M2000->Y:$078C00,0,8          ;I/O bits 0-7 (ACC-14E #1 port A)
M2001->Y:$078C01,0,8          ;I/O bits 8-15 (ACC-14E #1 port A)
M2002->Y:$078C02,0,8          ;I/O bits 16-23 (ACC-14E #1 port A)
M2003->Y:$078C03,0,8          ;I/O bits 0-7 (ACC-14E #1 port B)
M2004->Y:$078C04,0,8          ;I/O bits 8-15 (ACC-14E #1 port B)
M2005->Y:$078C05,0,8          ;I/O bits 16-23 (ACC-14E #1 port B)
M2006->Y:$078C06,0,8          ;register selected
M2007->Y:$078C07,0,8          ;control register

M2008->Y:$079C00,0,8          ;I/O bits 0-7 (ACC-14E #2 port A)
M2009->Y:$079C01,0,8          ;I/O bits 8-15 (ACC-14E #2 port A)
M2010->Y:$079C02,0,8          ;I/O bits 16-23 (ACC-14E #2 port A)
M2011->Y:$079C03,0,8          ;I/O bits 0-7 (ACC-14E #2 port B)
M2012->Y:$079C04,0,8          ;I/O bits 8-15 (ACC-14E #2 port B)
M2013->Y:$079C05,0,8          ;I/O bits 16-23 (ACC-14E #2 port B)
M2014->Y:$079C06,0,8          ;register selected
M2015->Y:$079C07,0,8          ;control register

M2016->Y:$07AC00,0,8          ;I/O bits 0-7 (ACC-14E #3 port A)
M2017->Y:$07AC01,0,8          ;I/O bits 8-15 (ACC-14E #3 port A)
M2018->Y:$07AC02,0,8          ;I/O bits 16-23 (ACC-14E #3 port A)
M2019->Y:$07AC03,0,8          ;I/O bits 0-7 (ACC-14E #3 port B)
```

```
M2020->Y:$07AC04,0,8          ;I/O bits 8-15 (ACC-14E #3 port B)
M2021->Y:$07AC05,0,8          ;I/O bits 16-23 (ACC-14E #3 port B)
M2022->Y:$07AC06,0,8          ;register selected
M2023->Y:$07AC07,0,8          ;control register

M2007->Y:$078C07,0,8          ;control word for $78C00,0,8 - $78C05,0,8
M2015->Y:$079C07,0,8          ;control word for $78A00,0,8 - $79C05,0,8
M2023->Y:$07AC07,0,8          ;control word for $7AC00,0,8 - $7AC05,0,8
                        ;**** PLC to initialize read/write I/O bits ****

OPEN PLC 1 CLEAR
M2007=$3F               ;define bits 0-23 and 24-47 as inputs (ACC-14E)
M2015=$00               ;define bits 0-23 and 24-47 as outputs (ACC-14E)
M2023=$07               ;define bits 0-23 as inputs and bits 24-47 as outputs
                        ;(ACC-14E)
DIS PLC1
CLOSE
```

Accessory 14E I/O M-Variables for UMAC Turbo

The following is a list of suggested M-variables for the default jumper settings is provided. Assign any M-variable to these addresses. For this example, assume that it has 24 inputs and 24 outputs. These M-variable definitions may be used as general purpose I/O for PLCs or motion programs.

```
M7000->Y:$078C00,0,1   Input 0      M7024->Y:$078C03,0,1 Output 0
M7001->Y:$078C00,1,1   Input 1      M7025->Y:$078C03,1,1 Output 1
M7002->Y:$078C00,2,1   Input 2      M7026->Y:$078C03,2,1 Output 2
M7003->Y:$078C00,3,1   Input 3      M7027->Y:$078C03,3,1 Output 3
M7004->Y:$078C00,4,1   Input 4      M7028->Y:$078C03,4,1 Output 4
M7005->Y:$078C00,5,1   Input 5      M7029->Y:$078C03,5,1 Output 5
M7006->Y:$078C00,6,1   Input 6      M7030->Y:$078C03,6,1 Output 6
M7007->Y:$078C00,7,1   Input 7      M7031->Y:$078C03,7,1 Output 7
M7008->Y:$078C01,0,1   Input 8      M7032->Y:$078C04,0,1 Output 8
M7009->Y:$078C01,1,1   Input 9      M7033->Y:$078C04,1,1 Output 9
M7010->Y:$078C01,2,1   Input 10     M7034->Y:$078C04,2,1 Output 10
M7011->Y:$078C01,3,1   Input 11     M7035->Y:$078C04,3,1 Output 11
M7012->Y:$078C01,4,1   Input 12     M7036->Y:$078C04,4,1 Output 12
M7013->Y:$078C01,5,1   Input 13     M7037->Y:$078C04,5,1 Output 13
M7014->Y:$078C01,6,1   Input 14     M7038->Y:$078C04,6,1 Output 14
M7015->Y:$078C01,7,1   Input 15     M7039->Y:$078C04,7,1 Output 15
M7016->Y:$078C02,0,1   Input 16     M7040->Y:$078C05,0,1 Output 16
M7017->Y:$078C02,1,1   Input 17     M7041->Y:$078C05,1,1 Output 17
M7018->Y:$078C02,2,1   Input 18     M7042->Y:$078C05,2,1 Output 18
M7019->Y:$078C02,3,1   Input 19     M7043->Y:$078C05,3,1 Output 19
M7020->Y:$078C02,4,1   Input 20     M7044->Y:$078C05,4,1 Output 20
M7021->Y:$078C02,5,1   Input 21     M7045->Y:$078C05,5,1 Output 21
M7022->Y:$078C02,6,1   Input 22     M7046->Y:$078C05,6,1 Output 22
M7023->Y:$078C02,7,1   Input 23     M7047->Y:$078C05,7,1 Output 23
```

```
;****** Sample E-Stop PLC *****
; This PLC will abort all motion programs and kill the bus voltage to
; the motors when E-stop is depressed.  When E-Stop button in pulled
out
; the motors will servo to actual position (<ctrl> A command) after
; allowing 5 seconds for proper bus voltage.
```

```
;     P7000 used as a Latch variable
;     M7000 used Emergancy Stop Input
;     M7024 used as Main Contact for main AC for Bus Voltage
;     I5111 used as count down timer
OPEN PLC 5 CLEAR
IF (M7000=1 and P7000=0)    ;emergency stop condition
    CMD^A                   ;global motion program abort
    I5111=500*8388608/I10   ;500 msec delay for deceleration
    WHILE (I5111>0) ENDWHILE
    CMD^K                   ;kill all axes
    M7024=0                 ;turn off BUS voltage
    P7000=1                 ;latch input
Endif

IF (M7000=0 and P7000=1)
    M7024=1                 ;enable BUS volatge
    I5111=5000*8388608/I10  ;5000 msec delay for bus voltage
    WHILE (I5111>0) ENDWHILE
    CMD^A                   ;close loop for all servos
    P7000=0                 ;latch input
Endif

close
```

UMAC Turbo Closed Loop Control Using Acc-14E

If providing position information to UMAC Turbo as a parallel data word (as from an absolute encoder or processed from a laser interferometer), the encoder conversion table must be configured properly.

The encoder conversion table can be modified using either PMAC's Executive Program Encoder Conversion Table dialog box or the on-line commands in the Executive terminal mode. The following sections describe in detail PMAC's ACC-14E parallel feedback conversion process and actual setup.

Extended Entries ($F)

Encoder conversion table entries in which the first hex digit of the first line is $F are extended entries. In these entries, the actual method is dependent on the first digit of the second line. Extended entries are a minimum of two lines.

Byte-Wide Parallel Feedback Entries ($F/$2, F3)

An ECT entry in which the first hex digit of the first line is $F and the first hex digit of the second line is $2 or $3 processes the result of a parallel data feedback source whose data is in byte-wide pieces in consecutive Y-words. This is used to process feedback from 3U-format parallel-data I/O boards: the ACC-3E in stack form, and the ACC-14E in pack (UMAC) form.

Address Word

The first setup line (I-variable) of the entry contains $F in the first hex digit (bits 20-23). The bit-19 mode-switch bit in the first line controls whether the least significant bit (LSB) of the source register is placed in Bit 5 of the result register (normal shift), providing the standard five bits of (non-existent) fraction, or the LSB is placed in Bit 0 of the result register (unshifted), creating no fractional bits.

Normally, the Bit-19 mode switch is set to 0 to place the source LSB in Bit 5 of the result register. Bit 19 is set to 1 to place to source LSB in Bit 0 of the result register for one of three reasons:

- The data already comes with 5 bits of fraction, as from a Compact MACRO Station.

- The normal shift limits the maximum velocity too much ($V_{max}<2^{18}$ LSBs per servo cycle)
- The normal shift limits the position range too much (Range<$\pm 2^{47}$/Ix08/32 LSBs)

Unless this is done because the data already contains fractional information, the unshifted conversion will mean that the motor position loop will consider one LSB of the source to be 1/32 of a count, instead of one count.

Bits 0 to 18 of the first line contain the base address of the parallel data to be read. This is the address of the least significant byte in the parallel feedback word. The following table shows the possible entries when an ACC-14E I/O board is used:

The following table shows the possible entries when the ACC-14E UMAC I/O board is used:

DIP-Switch Setting	**SW1-1 ON (0) SW1-2 ON (0)**	**SW1-1 OFF (1) SW1-2 ON (0)**	**SW1-1 ON (0) SW1-2 OFF (1)**	**SW1-1 OFF (1) SW1-2 OFF (1)**
SW1-3 ON (0) SW1-4 ON (0)	$F78C0x	$F78D0x	$F78E0x	$F78F0x
SW1-3 OFF (1) SW1-4 ON (0)	$F79C0x	$F79D0x	$F79E0x	$F79F0x
SW1-3 ON (0) SW1-4 OFF (1)	$F7AC0x	$F7AD0x	$F7AE0x	$F7AF0x
SW1-3 OFF (1) SW1-4 OFF (1)	$F7BC0x	$F7BD0x	$F7BE0x	$F7BF0x
A switch that is ON is closed; a switch that is OFF is open.				

If bit 19 is set to 1, the second digit should be changed from a 7 to an F to disable the data shift.

The final digit, represented by an *x* in both of these tables, can take a value of 0 to 5, depending on which I/O point on the board is used for the LSB:

x=0:	I/O00-07	I/O48-55	I/O96-103
x=1:	I/O08-15	I/O56-63	I/O104-111
x=2:	I/O16-23	I/O64-71	I/O112-119
x=3:	I/O24-31	I/O72-79	I/O120-127
x=4:	I/O32-39	I/O80-87	I/O128-135
x=5:	I/O40-47	I/O88-95	I/O136-143

Width/Offset Word

The second setup line (I-variable) of this parallel read entry contains information about what data is to be read starting at the base address. This 24-bit value, usually represented as six hexadecimal digits, is split into four parts, as shown in the following table.

Hex Digit	1	2	3	4	5	6
Contents	2 or 3	Bit Width		Byte	LSB Location	

The first hex digit contains a 2 or a 3. If it has a 2, there is no filtering of the data, and the entry is a 2-line entry. If it has a 3, the input data is filtered to protect against noise or data corruption, and the entry is a 3-line entry, with the third line controlling the filtering.

The second and third digits represent the width of the parallel data in bits, and can range from $01 (one bit wide – not of much practical use) to $18 (24 bits wide). If the value of these digits is from $01 to $08, only the base address in the first line is used. If the value of these digits is from $09 to $10 (16), the base

address and the next higher-numbered address are used. If the value of these digits is from $11 to $18 (17 to 24), three addresses starting at the base address are used.

The fourth digit represents which byte of the source word is used. It has three valid values:

- 0: Low byte (bits 0 – 7)
- 1: Middle byte (bits 8 – 15)
- 2: High byte (bits 16 – 23)

The fifth and sixth digits contain the bit location of the LSB of the data in the source word at the base address, and can range from $00 (Bit 0 of the source address is the LSB), through $07 (Bit 7 of the source address is the LSB). To calculate this value, divide the number of the I/O point used for the LSB by 8 and use the remainder here. For example, if I/O19 is used for the LSB, the remainder of 19/16 is 3.

Maximum Change Word

If the method character for a parallel read is $3 or $7, specifying filtered parallel read, there is a third setup line (I-variable) for the entry. This third line contains the maximum change in the source data in a single cycle that will be reflected in the processed result, expressed in LSBs per servo cycle. The filtering that this creates provides an important protection against noise and misreading of data. This number is effectively a velocity value, and should be set slightly greater than the maximum true velocity ever expected.

Example UMAC Turbo Encoder Conversion Table Setup for ACC-14E

Two 18-bit encoders are used in an application with 10000 cts/in and a maximum velocity of 20 in/sec is specified. Accessory 14E port A will be used for the fist encoder and port B will be used for the second encoder. The servo update rate is set at the factory default of 2258 Hz. For this example, setup an encoder with filtering, without filtering, and with 24-bit resolution.

First, calculate the maximum velocity per servo cycle:

$$V_{\max} = \frac{10000\,cts}{in} \times \frac{20in}{\sec} \times \frac{\sec}{2258\,cyc} = \frac{88.57\,cts}{cyc}$$

ECT Setup for 18-bit Encoder without Filtering

```
I8000=$F78C00    ;extended feedback entry from $078C00 ($3501)
I8001=$212000    ;18-bit ($12) from $078C00 base 0          ($3502)
I8002=$F78C03    ;extended feedback entry from $078C03 ($3503)
I8003=$212000    ;18-bit ($12) from $078C03 base 0          ($3504)

I103=$3502       ;position feedback for motor 1 from $3502
I104=$3502       ;velocity feedback for motor 1 from $3502
I203=$3504       ;position feedback for motor 2 from $3504
I204=$3504       ;velocity feedback for motor 2 from $3504
```

ECT Setup for 18-bit Encoder with Filtering

```
I8000=$F78C00    ;extended feedback entry from $078C00 ($3501)
I8001=$312000    ;18-bit ($12) from $078C00 base 0          ($3502)
I8002=$00006E    ;filter 110cts/cyc  (88.57*1.25)           ($3503)
I8003=$F78C03    ;extended feedback entry from $078C03 ($3504)
I8004=$312000    ;18-bit ($12) from $078C03 base 0          ($3505)
I8005=$00006E    ;filter 110cts/cyc  (88.57*1.25)           ($3506)

I103=$3503       ;position feedback for motor 1 from $3502
I104=$3503       ;velocity feedback for motor 1 from $3502
I203=$3506       ;position feedback for motor 2 from $3504
```

```
I204=$3506        ;velocity feedback for motor 2 from $3504
```

Absolute Power-On Position for UMAC

The Turbo PMAC allows the user to obtain absolute position at power up or upon request (#n$*). To read this absolute position the Turbo PMAC needs both Ixx10 and Ixx95 setup properly to enable this power on position function.

Ixx10 tells the PMAC what address to read the position from and Ixx95 tells the PMAC how to process the data in Ixx10.

The data in Ixx95 should be read as a Y-word, Byte-wide format, starting at the low byte (bits 0-7). This means bit 22 is always zero and the last digit of Ixx95 is always 4. The amount of bits to be read and whether or not it is a signed read is determined by the hardware.

Example: Encoder1 is an 18-bit encoder from Y:$78C00 and Encoder2 is an 18-bit encoder from Y:$78C003.

```
I110=$78C00       ;Mtr1 Abs Position from $78C00
I210=$78C03       ;Mtr2 Abs Position from $78C03
I195=$012004      ;Y-word 18-bit Byte-wide from bits 0-7
I295=$012004      ;Y-word 18-bit Byte-wide from bits 0-7
```

ABSOLUTE ENCODER LATCHING AND HANDSHAKING

When using a parallel-word absolute encoder, it is important to properly latch the encoder data to prevent PMAC from reading the encoder data during an encoder transition. ACC-14E allows several latching and handshaking methods to fit most types of latching schemes.

> ***Note:***
>
> It is equally important to set up the Encoder Conversion Table Filter Word as a software protection against bad encoder data.

PMAC reads the encoder data when it processes the Encoder Conversion Tables. This happens shortly (approximately 2 μsec) after the falling edge of the servo clock (the phase calculations are performed first). Therefore, most of the following latching methods will be synchronized to the falling edge of the servo clock.

The error signal inputs (ERR1/ and ERR2/) allow the feedback device to send a signal to the latch circuit to interrupt the latch. Jumpers E8 and E10 are used to set the polarity for the error inputs signals. If the feedback device does not have an error signal output, then set the jumpers E8 and E10 from 1-2 to allow the latch circuit to work properly. See the Jumper Descriptions section of this manual for details on E8 and E10.

UMAC Turbo Latching Setup

For these examples, two standard 24-bit absolute encoders will be read from port A and port B of the first ACC-14E at base address Y:$078C00 for the Turbo 3U PMAC .

The 24-bits of data will be addressed to:

```
M2000->Y:$078C00,0,8  ;Bits 0-7 for absolute encoder port A
M2001->Y:$078C01,0,8  ;Bits 8-15 for absolute encoder port A
M2002->Y:$078C02,0,8  ;Bits 16-23 for absolute encoder port A
M2003->Y:$078C03,0,8  ;Bits 24-31 for absolute encoder port B
M2004->Y:$078C04,0,8  ;Bits 32-39 for absolute encoder port B
M2005->Y:$078C05,0,8  ;Bits 40-47 for absolute encoder port B

M2007->Y:$078C07,0,8  ;Control word to setup Latch
```

Enabling ACC-14E Latching for UMAC Turbo

To enable the ACC-14E latching mechanism, set up each bit of the input word to be latched. This is accomplished by writing to the Setup Variable 2 and Setup Variable 3 when accessed by Bits 6 and 7 of the control word. For MACRO, MI152 and MI153 will setup the control word for latching.

Setup Variable 2 (Control Word Bit 6=0 and Bit 7=1)

```
M2007=$80    ;Address Setup Variable 2 for Binary or Greycode
M2000=$00    ;Binary Data Read ($FF for Greycode) for bits 0-7 port A
M2001=$00    ;Binary Data Read ($FF for Greycode) for bits 8-15 port A
M2002=$00    ;Binary Data Read ($FF for Greycode) for bits 16-23 port A
M2003=$00    ;Binary Data Read ($FF for Greycode) for bits 24-31 port B
M2004=$00    ;Binary Data Read ($FF for Greycode) for bits 32-39 port B
M2005=$00    ;Binary Data Read ($FF for Greycode) for bits 40-47 port B
```

Setup Variable 3 (Control Word Bit 6=1 and Bit 7=1)

```
M2007=$C0      ;Address Setup Variable 2 for Binary or Greycode
M2000=$FF      ;Latched Data Read ($00 for unlatched) for bits 0-7
               ;port A
M2001=$FF      ;Latched Data Read ($00 for unlatched) for bits 8-15
               ;port A
M2002=$FF      ;Latched Data Read ($00 for unlatched) for bits 16-23
               ;port A
M2003=$FF      ;Latched Data Read ($00 for unlatched) for bits 24-31
               ;port B
M2004=$FF      ;Latched Data Read ($00 for unlatched) for bits 32-39
               ;port B
M2005=$FF      ;Latched Data Read ($00 for unlatched) for bits 40-47
              port B
```

Once the method of latching has been defined, set up the control word to read the data. Also, set up the bits of the encoder inputs to be read only as shown below.

```
M2007=$3F      ;ensure bits 0-23 (portA) and 24-47 (port B) are
               ;read only
```

ACC-14E Latch Setup for MACRO

MI152 and MI153 permit the use of inputs latched by the phase clock on ACC-14E. This function is used to get reliable parallel-data feedback on the MACRO Station. It is useful mainly on ACC-3E stack boards and ACC-14E backplane boards.

MI52 and MI153 are 48-bit values represented by 12 hexadecimal digits. These digits have the following functions:

Digits	Function and Setting
1 & 2	3rd I/O ASIC Latch Control (Maps into high bytes; Option C on ACC-3E 144-I/O board) =$C0 for latched inputs =$00 for transparent inputs or ASIC not present
3 & 4	2nd I/O ASIC Latch Control (Maps into middle bytes; Option B on ACC-3E 144-I/O board) =$C0 for latched inputs =$00 for transparent inputs or ASIC not present
5 & 6	1st I/O ASIC Latch Control (Maps into low bytes; Option A on ACC-3E 144-I/O board or ACC14E) =$C0 for latched inputs =$00 for transparent inputs or ASIC not present
7	Number of bytes (1 to 6) on each ASIC (starting with lowest byte) to latch
8	(Reserved for future use; set to 0)
9 - 12	Base address of I/O Board =$FFC0 (ACC-3E board w/ E1 ON; ACC-4E board w/ E15 ON) =$FFC8 (ACC-3E board w/ E2 ON; ACC-4E board w/ E16 ON) =$FFD0 (ACC-3E board w/ E3 ON; ACC-4E board w/ E17 ON) =$FFD8 (ACC-3E board w/ E4 ON; ACC-4E board w/ E18 ON) =$FFE0 (ACC-14E) =$FFE8 (ACC-14E) =$FFF0 (ACC-14E) =$B8C0 (ACC-14E)

Example:

```
MS0,MI152=$0000C060FFE0     ; Latches inputs on ACC-14E for 6 bytes
                            ; (48-bits)
```

ACC-14E Latching Examples

Method 1

This method requires the encoder outputs to be latched on the falling edge of the servo clock and no latching to be done on ACC-14E. For latching the encoder outputs, the servo clock is accessed through ACC-14E OCLK1 (see J4 pinout) and/or OCLK2 (see J5 pinout). If the encoder requires a rising edge for its latch, then E5 should be jumpered 1 to 2 for OCLK1/OCLK2 respectively. If a falling edge is required, E5 should be jumpered 2 to 3.

Required Signal	E5	E6	E7/E9	Latch
Rising edge of OCLK	1 to 2	Don't care	Don't care	No
Falling edge of OCLK	2 to 3	Don't care	Don't care	No

The advantages and disadvantages of this method are as follows:

Advantage	Easy to configure and set up.
Disadvantage	Encoder's output latch must typically happen within 2 μsec.

Method 2

This method requires the encoder outputs to be latched on the rising edge of the servo clock and ACC-14E to latch (strobe) the encoder inputs on the falling edge of the servo clock. For latching the encoder outputs, the servo clock is accessed through ACC-14E OCLK1 (see J4 pinout) and/or OCLK2 (see J5 pinout). If the encoder requires a rising edge for its latch, then E5 should be jumpered 1 to 2 for OCLK1/OCLK2 respectively. If a falling edge is required, E5 should be jumpered 2 to 3. The control words for the data must be setup for latching to allow the ACC-14E latch the data.

Required Signal	E5	E6	E7/E9	Latch
Rising edge of OCLK	1 to 2	2 to 3	ON	Yes
Falling edge of OCLK	2 to 3	1 to 2	ON	Yes

The advantages and disadvantages of this method are as follows:

Advantage	Encoder latch-time is not very critical (have almost 1 servo cycle to latch).
Disadvantage	Almost a 1-servo cycle delay between encoder output latch and ACC-14E encoder read.

Method 3

This method requires a self-latching encoder that outputs a signal that indicates it is latched and an ACC-14E that latches (strobes) the encoder inputs on the falling edge of the ICLK only when the servo clock is low. The encoder-latched indicator is brought into ACC-14E via the ICLK1/2 inputs (see J4 and/or J5 pinout). If the encoder outputs a rising edge for its latch indicator, then E7/E9 should be jumpered, so that a rising ICLK latches the data when the servo is low. If a falling edge indicator is output, E7/E9 should not be jumpered, so that a falling ICLK latches the data when the servo is low. The control words for the data must be setup for latching to allow the ACC-14E to latch the data.

Required Signal	E5	E6	E7/E9	Latch
High ICLK means latched	Don't care	2 to 3	ON	Yes
Low ICLK means latched	Don't care	2 to 3	OFF	Yes

The advantages and disadvantages of this method are as follows:

Advantage	Can only read latched encoder data.
Disadvantage	Encoder latch is asynchronous to PMAC's servo cycle.

Method 4

This method is a combination of methods 1 and 3 above. It requires that the encoder outputs be latched on the falling edge of the servo clock and the encoder to signal that it is latched. Also, ACC-14E must latch (strobe) the encoder inputs on an edge of the ICLK only when the servo clock is low. For latching the encoder outputs, the servo clock is accessed through ACC-14E OCLK1 (see J4) and/or OCLK2 (see J5). If the encoder requires a rising edge for its latch, then E5 should be jumpered 1 to 2 for OCLK1/OCLK2 respectively. If a falling edge is required, E5 should be jumpered 2 to 3. The encoder-latched indicator is brought into ACC-14E via the ICLK1/2 inputs (see J4 & J5 pinouts). If the encoder outputs a rising edge for its latch indicator, then E7/E9 should be jumpered so that a rising ICLK latches the data when the servo is low. If a falling edge indicator is output, E7/E9 should not be jumpered so that a falling ICLK latches the data when the servo is low. The control words for the data must be setup for latching to allow the ACC-14E to latch the data.

Required Signal	E5	E6	E7/E9	Latch
High OCLK means latch High ICLK means latched	1 to 2	1 to 2	ON	Yes
Low OCLK means latch High ICLK means latched	2 to 3	1 to 2	ON	Yes
High OCLK means latch Low ICLK means latched	1 to 2	1 to 2	OFF	Yes
Low OCLK means latch Low ICLK means latched	2 to 3	1 to 2	OFF	Yes

The advantages and disadvantages of this method are as follows:

Advantage	Can only read latched encoder data. Have full handshaking between PMAC and encoder
Disadvantage	Typically, encoder's output latch must happen within 2 μsec. More complex wiring and timing.

Method 5

This method requires no latching on the encoder outputs and latching on the ACC-14E inputs at the falling edge of the servo clock. For the encoder, no signals are used so the state of the OCLK does not matter. E7/E9 must be jumpered and latching must be enabled from the control to allow ACC-14E to latch (strobe) its inputs with the falling edge of the servo clock. The control words for the data must be set up for latching to allow the ACC-14E to latch the data.

Required Signal	E5	E6	E7/E9	Latch
Latch at Falling Edge	Don't care	2 to 3	ON	Yes
Latch at Rising Edge	Don't care	1 to 2	ON	Yes

The advantages and disadvantages of this method are as follows:

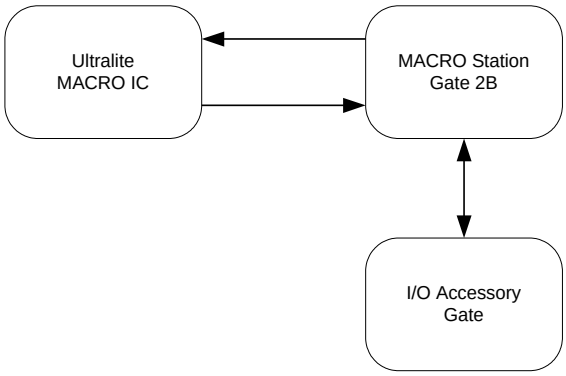
Advantage	Easy to configure and set up.
Disadvantage	The encoder data may be latched into ACC-14E at an encoder transition causing bad encoder data for that servo cycle. Must set up the Encoder Conversion Table Filter

MACRO-STATION I/O TRANSFER

A fundamental understanding of the MACRO Station I/O transfer is needed to set up the MACRO I/O family of accessories.

Typically, the MACRO station will have up 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). There are two types of I/O transfers allowed to send the information to the Ultralite from the MACRO-Station: 48-bit transfer and 24-bit transfer. The PMAC2 Ultralite and the MACRO-Station can transfer up to 72 bits per I/O node. For a multi Master system, 432 bits (6×72) of data may be transferred for each Master (Ultralite) in the ring. If only one Master is used in the ring, node 14 could be used for I/O transfer, which would give 504 bits (7×72) of I/O transfer data.

For all MACRO-Station I/O accessories, the information is transferred to or from the accessory I/O Gate to the MACRO-Station CPU Gate 2B. Information from the MACRO-Station Gate 2B is then read or written directly to the MACRO IC on the Ultralite. Once the information is at the Ultralite, it can be used in the application's motion programs or PLC programs.



Each I/O board has jumper and software settings to select the I/O transfer memory locations at both the I/O transfer Gate and the MACRO transfer addresses. These jumpers and software settings are discussed in this manual.

MACRO Station I/O Node Transfer Addresses

Node(s)	Node 24-bit: Transfer Addresses	Node 16-bit (upper 16 bits): Transfer Addresses
2	X:$C0A0	X:$C0A1, X:$C0A2, X:$C0A3
3	X:$C0A4	X:$C0A5, X:$C0A6, X:$C0A7
6	X:$C0A8	X:$C0A9, X:$C0AA, X:$C0AB
7	X:$C0AC	X:$C0AD, X:$C0AE, X:$C0AF
10	X:$C0B0	X:$C0B1, X:$C0B2, X:$C0B3
11	X:$C0B4	X:$C0B5, X:$C0B6, X:$C0B7

PMAC2 Ultralite I/O Node Addresses

Node	Node 24-bit: Transfer Addresses	Node 16-bit (upper 16 bits): Transfer Addresses
2	X:$C0A0	X:$C0A1, X:$C0A2, X:$C0A3
3	X:$C0A4	X:$C0A5, X:$C0A6, X:$C0A7
6	X:$C0A8	X:$C0A9, X:$C0AA, X:$C0AB
7	X:$C0B0	X:$C0B1, X:$C0B2, X:$C0B3
10	X:$C0B4	X:$C0B5, X:$C0B6, X:$C0B7
11	X:$C0B8	X:$C0B9, X:$C0BA, X:$C0BB

PMAC2 Turbo Ultralite I/O Node Addresses

MACRO IC Node	User Node	Node 24-bit: Transfer Addresses	Node 16-bit (upper 16 bits): Transfer Addresses
(IC0) 2	2	X:$078420	X:$078421, X:$078422, X:$078423
(IC0) 3	3	X:$078424	X:$078425, X:$078426, X:$078427
(IC0) 6	6	X:$078428	X:$078429, X:$07842A, X:$07842B
(IC0) 7	7	X:$07842C	X:$07842D, X:$07842E, X:$07842F
(IC0) 10	10	X:$078430	X:$078431, X:$078432, X:$078433
(IC0) 11	11	X:$078434	X:$078435, X:$078436, X:$078437
(IC1) 2	18	X:$079420	X:$079421, X:$079422, X:$079423
(IC1) 3	19	X:$079424	X:$079425, X:$079426, X:$079427
(IC1) 6	22	X:$079428	X:$079429, X:$07942A, X:$07942B
(IC1) 7	23	X:$07942C	X:$07942D, X:$07942E, X:$07942F
(IC1) 10	26	X:$079430	X:$079431, X:$079432, X:$079433
(IC1) 11	27	X:$079434	X:$079435, X:$079436, X:$079437
(IC2) 2	34	X:$078420	X:$07A421, X:$07A422, X:$07A423
(IC2) 3	35	X:$07A424	X:$07A425, X:$07A426, X:$07A427
(IC2) 6	38	X:$07A428	X:$07A429, X:$07A42A, X:$07A42B
(IC2) 7	39	X:$07A42C	X:$07A42D, X:$07A42E, X:$07A42F
(IC2) 10	42	X:$07A430	X:$07A431, X:$07A432, X:$07A433
(IC2) 11	43	X:$07A434	X:$07A435, X:$07A436, X:$07A437
(IC3) 2	50	X:$07B420	X:$07B421, X:$07B422, X:$07B423
(IC3) 3	51	X:$07B424	X:$07B425, X:$07B426, X:$07B427
(IC3) 6	54	X:$07B428	X:$07B429, X:$07B42A, X:$07B42B
(IC3) 7	55	X:$07B42C	X:$07B42D, X:$07B42E, X:$07B42F

MACRO IC Node	User Node	Node 24-bit: Transfer Addresses	Node 16-bit (upper 16 bits): Transfer Addresses
(IC3) 10	58	X:$07B430	X:$07B431, X:$07B432, X:$07B433
(IC3) 11	59	X:$07B434	X:$07B435, X:$07B436, X:$07B437

Example: To read the inputs from the MACRO Station of the first 24-bit I/O node address of node 2 (X:$C0A0), then point an M-variable to the Ultralite or Turbo Ultralite I/O node registers to monitor the inputs.

```
M980->X:$C0A0,0,24              ;Ultralite node2 address
M1980->X:$078420,0,24           ;Turbo Ultralite MACRO IC0 node 2 address
```

Then these M-variable definitions (M980 or M1980) could be used to monitor the inputs for either the Ultralite or Turbo Ultralite, respectively.

MACRO I/O SOFTWARE SETTINGS

The MACRO-Station I/O can be configured as either an input or an output. The hardware connected to the MACRO I/O boards determines whether the addresses are defined as inputs or outputs. Each I/O node has 72-bits of data to be transferred to the Ultralite automatically. As stated previously, there are three methods of transfer: 3×16-bit, 1×24-bit, or 72-bit transfer.

There are several variables at the MACRO-Station and PMAC2 Ultralite that enable the I/O data transfer. Once these variables are set to the appropriate values, then the data can be processed like a normal PMAC or PMAC2. The MI19, MI69, MI70, or MI71 variables are modified at the MACRO Station.

To read multiple extended address UMAC I/O cards efficiently, a new function was added to MACRO firmware 1.16 to read consecutive extended addresses. The previous method supported the low-middle-high byte addressing of the Type A I/O cards (see the Hardware Address Limitations section in this manual).

MI19 Variable

This variable controls the data transfer period on a Compact MACRO Station between the MACRO node interface registers and the I/O registers, as specified by station MI-variables MI20 through MI71. If MI19 is set to 0, this data transfer is disabled. If MI19 is greater than 0, its value sets the period in Phase clock cycles (the same as MACRO communications cycles) at which the transfer is done.

MI975 Variable

This variable permits the enabling of MACRO I/O nodes on the Compact MACRO Station. MI975 is a 16-bit value (bits 0 to 15) with bit *n* controlling the enabling of MACRO node *n*. If the bit is set to 0, the node is disabled; if the bit is set to 1, the node is enabled. The I/O nodes on the Compact MACRO Station are 2, 3, 6, 7, 10, and 11, which can be enabled by MI975 bits of these numbers. Only bits 2, 3, 6, 7, 10, and 11 of MI975 should ever be set to 1.

MI975 is used at the power-on/reset of the Compact MACRO Station in combination with rotary switch SW1 and MI976 to determine which MACRO nodes are to be enabled. The net result can be read in Station variable MI996. To get a value of MI975 to take effect, the value must be saved (**MSSAVE{node}**) and the Station reset (**MS$$${node}**).

Example:

Set MI975 to enable nodes 2 and 3.
MS0, I975 — Set number of MACRO I/O nodes to be enabled.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

∴MS0, I975=$000C

```
MS0,MI975=$4          ; Enable I/O Node 2 alone
MS0,MI975=$C          ; Enable I/O Nodes 2 & 3
MS0,MI975=$4C         ; Enable I/O Nodes 2, 3, & 6
MS0,MI975=$CC         ; Enable I/O Nodes 2, 3, 6, & 7
MS0,MI975=$4CC        ; Enable I/O Nodes 2, 3, 6, 7, & 10
MS0,MI975=$CCC        ; Enable I/O Nodes 2, 3, 6, 7, 10, & 11
MS4,MI975=$40         ; Enable I/O Node 6 alone
MS4,MI975=$C0         ; Enable I/O Nodes 6 & 7
MS8,MI975=$400        ; Enable I/O Node 10 alone
MS8,MI975=$C00        ; Enable I/O Nodes 10 & 11
```

MI69 and MI70 Variables

Specify the registers used in 16-bit I/O transfers between MACRO node interface registers and I/O registers on the MACRO Station I/O accessory board. They are used only if MI19 is greater than 0.

Note

The examples for the setup of MI69 and MI70 require MACRO firmware 1.16 and above.

MI69 and MI70 are 48-bit variables represented as 12 hexadecimal digits. The first six digits specify the number and address of 48-bit (3 x 16) real-time MACRO-node register sets to be used. The second six digits specify the number and address of 16-bit I/O sets on the MACRO Station I/O accessory board to be used. The individual digits are specified as follows:

Digit #	Possible Values	Description
1	0, 1, 2, 3	Number of MACRO I/O nodes to use (0 disables); this should also match the number of 48-bit I/O sets you intend to use (see Digit 7)
2	0	(Reserved for future use)
3-6	$C0A1 (Node 2), $C0A5 (Node 3), $C0A9 (Node 6), $C0AD (Node 7), $C0B1 (Node 10), $C0B5 (Node 11)	MACRO Station X Address of MACRO I/O node first of three 16-bit registers
7	0, 1, 2, 3	Number of 16-bit I/O sets to use (1x16, 2x16, 3x16; 0 disables)
8	1	Set to 1 for ACC-14E, ACC-65E, ACC-66E, ACC-67E consecutive address read (Base, +$1000, +$2000)
9-12	$8800, $8840 $8880, $88C0	MACRO Station Y Base Address of ACC-14E, ACC-65E, ACC-66E, ACC-67E,

When this function is active, the MACRO Station will copy values from the MACRO command (input) node registers to the I/O board addresses; it will copy values from the I/O board addresses to the MACRO feedback (output) node registers. Writing a 0 to a bit of the I/O board enables it as an input, letting the output pull high. Writing a 1 to a bit of the I/O board enables it as an output and pulls the output low.

Example:

1. 48 bit I/O transfer using node 2 with IO card base address of $8800

```
MS0,MI69=$10C0A1318800
```

2. 96 bit I/O transfer using nodes 2 and 3, with IO card base address of $8800 and $9800

```
MS0,MI69=$20C0A1318800
```

3. 288 bit I/O transfer using nodes 2, 3, 6, 7, 10, and 11 using 6 IO cards. Setup using 144-bit transfer with MI69 and 144-bit transfer with MI70. The first three IO cards are addressed at $8800, $9800, and $A800. The second three IO cards are addressed at $8840, $9840, and $A840.

```
MS0,MI69=$30C0A1318800
MS0,MI70=$30C0AD318840
```

MI71 Variable

This variable specifies the registers used in 24-bit I/O transfers between MACRO I/O node interface registers and I/O registers on the MACRO Station I/O accessory board. It is used only if MI19 is greater than 0.

Note

The examples for the setup of MI71 require MACRO firmware 1.16 and above.

MI71 is a 48-bit variable represented as 12 hexadecimal digits. The first six digits specify the number and address of 48-bit real-time MACRO-node register sets to be used. The second six digits specify the number and address of 48-bit I/O sets on the MACRO Station I/O accessory board to be used. The individual digits are specified as follows:

Digit #	Possible Values	Description
1	0, 1, 2, 3	Number of MACRO I/O nodes to use times 2 (0 disables); this should also match the number of 48-bit I/O sets you intend to use (see Digit 7)
2	0	(Reserved for future use)
3-6	$C0A0 (Node 2), $C0A4 (Node 3), $C0A8 (Node 6), $C0AC (Node 7), $C0B0 (Node 10), $C0B4 (Node 11)	MACRO Station X Address of MACRO I/O node first of three 16-bit registers
7	0, 1, 2	Number of 24-bit I/O sets to use (1x24, 2x24; 0 disables)
8	1	Set to 1 for ACC-14E, ACC-65E, ACC-66E, ACC-67E consecutive address read (Base, +$1000, +$2000)
9-12	$8800, $8840 $8880, $88C0	MACRO Station Y Base Address of ACC-14E, ACC-65E, ACC-66E, ACC-67E,

When this function is active, the MACRO Station will copy values from the MACRO command (input) node registers to the I/O board addresses. In addition, it will copy values from the I/O board addresses to the MACRO feedback (output) node registers. Writing a 0 to a bit of the I/O board enables it as an input, letting the output pull high. Writing a 1 to a bit of the I/O board enables it as an output and pulls the output low.

Example:

1. 48 bit I/O transfer using nodes 2 and 3 with I/O card base address of $8800

   ```
   MS0,MI71=$10C0A01218800
   ```

2. 96 bit I/O transfer using nodes 2, 3, 6, and 7 with I/O card base address of $8800 and $9800

   ```
   MS0,MI71=$20C0A0218800
   ```

3. (3) 144-bit I/O transfer using nodes 2, 3, 6, 7, 10, and 11 using three I/O cards at base address $8800, $9800 and $A800.

   ```
   MS0,MI71=$30C0A0218800
   ```

READING AND WRITING TO NODE ADDRESSES

Read and write to the node address as complete words. If the node address is 24-bits wide or 16-bits wide, read or write to the M-Variable assigned to that address:

Example:

Ultralite	TURBO Ultralite
M970->X:$C0A0,0,24	M970->X:$78420,0,24
M980->X:$C0A1,8,16	M980->X:$78421,8,16
M981->X:$C0A2,8,16	M981->X:$78422,8,16
M982->X:$C0A3,8,16	M982->X:$78423,8,16
M1000->X:$0770,0,24	M1000->X:$0010F0,0,24 ;image word
M1001->X:$0771,8,16	M1001->X:$0010F0,8,16 ;image word
For Outputs:	
M970=$F00011	;sets bits 0,4,20,21,22,& 23
M980=$8101	;sets bits 0,8,& 23
M970=M1000	;sets M970 equal to M1000
M980=M1001	;sets M980 equal to M1001
For Inputs:	
M1000=M970	;sets M1000 equal to M970
M1001=M980	;sets M1001 equal to M980

If using the 48-bit read/write method, use the inputs and outputs in multiples of 16. For example, 48 inputs, 32 inputs, 16 outputs, 16 inputs 32 outputs, or 48 outputs (see Example 2).

If the 16-bit word is to be split (eight in and eight out), then we would read the word at the beginning of the PLC and write the word at the end of the PLC. However, instead of writing the value of the inputs to the output word, write zeros to all input bits of this in/out word. This is because writing a value of 1 to a MACRO-I/O register makes that I/O bit an output only bit. The best method to ensure proper input reads is to write directly to the control word of the I/O gate array to set the input words as read only (see Setting Up Control Word for MACRO I/O section in this manual).

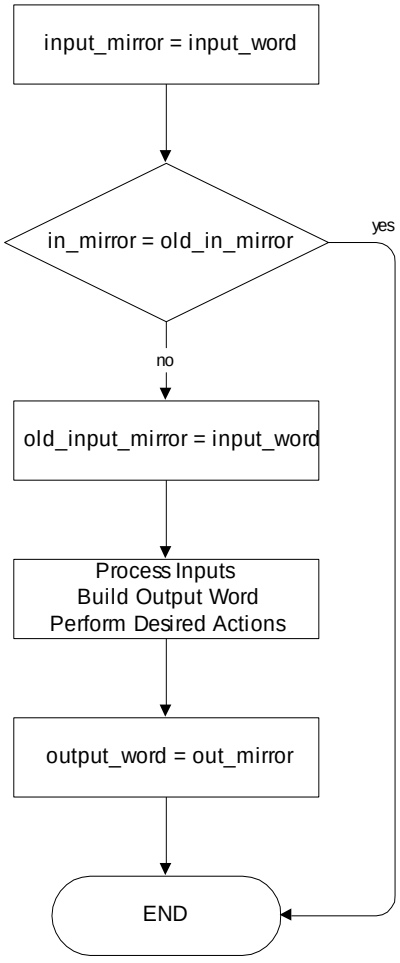
Example Setup

System Configuration:	8-axis PWM System w/ 96 bit I/0 (48 inputs & 48 outputs) ACC-14E
PMAC Ultralite Setup: **I996=$FB33F**	;activates nodes 1,2,3,4,5,8,9,12, and 13 at Ultralite
Turbo PMAC Ultralite Setup **I6841=$FB33F**	;activates nodes 1,2,3,4,5,8,9,12, and 13 at Turbo Ultralite
Macro Station Definitions:	
MS0,MI69=$20C0A131FFE0	;sets up macro to transfer data for ACC14E
MS0,MI975=$C	;enable node 2 and 3 for I/O
MS0,MI19=4	;sets interrupt period for data transfer
MSSAVE0	;save to macro station
MS$$$0	;reset macro station to enable

Active Nodes for Compact MACRO I/O Station

Option	Node(s)	Gate Addresses	Node Transfer Addresses
48-Bit	2	$8800	$C0A1,$C0A2,$C0A3
96-Bit	2,3	$8800 $9800	$C0A1,$C0A2,$C0A3 $C0A5,$C0A6,$C0A7
144-Bit	2,3,6	$8800 $9800 $A800	$C0A1,$C0A2,$C0A3 $C0A5,$C0A6,$C0A7 $C0A9,$C0AA,$C0AB

The data in this application will transfer 48-bits of data per node as specified by MI69. These memory locations could be utilized by pointing an M-variable to the node locations. In the PLC program, these M-variables would be considered the actual input words and actual output words or a combination of the two (eight inputs/eight outputs for 16-bit read/write). To efficiently read and write to these memory locations, Delta Tau suggests using image input words to read the actual input words and then write to the actual output word if the inputs have changed states. The following block diagram shows the typical logic for PMAC's inputs and outputs.



For this application, we are using six 16-bit data transfers and will use the following M-Variable definitions in our application.

PMAC2 Ultralite Example M-Variable Definitions

```
M980->X:$C0A1,8,16          ;IO word #1, 1st 16 bit word node2
M981->X:$C0A2,8,16          ;IO word #2, 2nd 16 bit word node 2
M982->X:$C0A3,8,16          ;IO word #3, 3rd 16 bit word node 2
M983->X:$C0A5,8,16          ;IO word #1, 1st 16 bit word node 3
M984->X:$C0A6,8,16          ;IO word #2, 2nd 16 bit word node 3
M985->X:$C0A7,8,16          ;IO word #3, 3rd 16 bit word node 3

M1000->X:$0770,8,16         ;Input mirror word #1
M1001->Y:$0770,8,16         ;Input mirror word #2
M1002->X:$0771,8,16         ;Input mirror word #3
M1003->Y:$0771,8,16         ;Output mirror word #1
M1004->X:$0772,8,16         ;Output mirror word #2
M1005->Y:$0772,8,16         ;Output mirror word #3
M1010->X:$0773,8,16         ;Old Image mirror word #1
M1011->Y:$0773,8,16         ;Old Image mirror word #2
M1012->X:$0774,8,16         ;Old Image mirror word #3
```

IO word #1	IO Word #2	IO Word #3
M800->X:$770,8	M816->Y:$770,8	M832->X:$771,8
M801->X:$770,9	M817->Y:$770,9	M833->X:$771,9
M802->X:$770,10	M818->Y:$770,10	M834->X:$771,10
M803->X:$770,11	M819->Y:$770,11	M835->X:$771,11
M804->X:$770,12	M820->Y:$770,12	M836->X:$771,12
M805->X:$770,13	M829->Y:$770,13	M837->X:$771,13
M806->X:$770,14	M822->Y:$770,14	M838->X:$771,14
M807->X:$770,15	M823->Y:$770,15	M839->X:$771,15
M808->X:$770,16	M824->Y:$770,16	M840->X:$771,16
M809->X:$770,17	M825->Y:$770,17	M841->X:$771,17
M810->X:$770,18	M826->Y:$770,18	M842->X:$771,18
M811->X:$770,19	M827->Y:$770,19	M843->X:$771,19
M812->X:$770,20	M828->Y:$770,20	M844->X:$771,20
M813->X:$770,21	M829->Y:$770,21	M845->X:$771,21
M814->X:$770,22	M830->Y:$770,22	M846->X:$771,22
M815->X:$770,23	M831->Y:$770,23	M847->X:$771,23

IO word #4	IO Word #5	IO Word #6
M900->Y:$771,8	M916->X:$772,8	M932->Y:$772,8
M901->Y:$771,9	M917->X:$772,9	M933->Y:$772,9
M902->Y:$771,10	M918->X:$772,10	M934->Y:$772,10
M903->Y:$771,11	M919->X:$772,11	M935->Y:$772,11
M904->Y:$771,12	M920->X:$772,12	M936->Y:$772,12
M905->Y:$771,13	M129->X:$772,13	M937->Y:$772,13
M906->Y:$771,14	M922->X:$772,14	M938->Y:$772,14
M907->Y:$771,15	M923->X:$772,15	M939->Y:$772,15
M908->Y:$771,16	M924->X:$772,16	M940->Y:$772,16
M909->Y:$771,17	M925->X:$772,17	M941->Y:$772,17
M910->Y:$771,18	M926->X:$772,18	M942->Y:$772,18
M911->Y:$771,19	M927->X:$772,19	M943->Y:$772,19
M912->Y:$771,20	M928->X:$772,20	M944->Y:$772,20
M913->Y:$771,21	M129->X:$772,21	M945->Y:$772,21
M914->Y:$771,22	M930->X:$772,22	M946->Y:$772,22
M915->Y:$771,23	M931->X:$772,23	M947->Y:$772,23

PMAC2 Turbo Ultralite Example M-Variable Definitions

```
M980->X:$78421,8,16          ;IO word #1, 1st 16 bit word node2
M981->X:$78422,8,16          ;IO word #2, 2nd 16 bit word node 2
M982->X:$78423,8,16          ;IO word #3, 3rd 16 bit word node 2
M983->X:$78425,8,16          ;IO word #1, 1st 16 bit word node 3
M984->X:$78426,8,16          ;IO word #2, 2nd 16 bit word node 3
M985->X:$78427,8,16          ;IO word #3, 3rd 16 bit word node 3
M1000->X:$0010F0,8,16        ;Input mirror word #1
M1001->Y:$0010F0,8,16        ;Input mirror word #2
M1002->X:$0010F1,8,16        ;Input mirror word #3
M1003->Y:$0010F1,8,16        ;Output mirror word #1
M1004->X:$0010F2,8,16        ;Output mirror word #2
M1005->Y:$0010F2,8,16        ;Output mirror word #3
M1010->X:$0010F3,8,16        ;Old Image mirror word #1
M1011->Y:$0010F3,8,16        ;Old Image mirror word #2
M1012->X:$0010F4,8,16        ;Old Image mirror word #3
```

IO word #1	IO Word #2	IO Word #3
M800->X:$0010F0,8	M816->Y:$0010F0,8	M832->X:$0010F1,8
M801->X:$0010F0,9	M817->Y:$0010F0,9	M833->X:$0010F1,9
M802->X:$0010F0,10	M818->Y:$0010F0,10	M834->X:$0010F1,10
M803->X:$0010F0,11	M819->Y:$0010F0,11	M835->X:$0010F1,11
M804->X:$0010F0,12	M820->Y:$0010F0,12	M836->X:$0010F1,12
M805->X:$0010F0,13	M829->Y:$0010F0,13	M837->X:$0010F1,13
M806->X:$0010F0,14	M822->Y:$0010F0,14	M838->X:$0010F1,14
M807->X:$0010F0,15	M823->Y:$0010F0,15	M839->X:$0010F1,15
M808->X:$0010F0,16	M824->Y:$0010F0,16	M840->X:$0010F1,16
M809->X:$0010F0,17	M825->Y:$0010F0,17	M841->X:$0010F1,17
M810->X:$0010F0,18	M826->Y:$0010F0,18	M842->X:$0010F1,18
M811->X:$0010F0,19	M827->Y:$0010F0,19	M843->X:$0010F1,19
M812->X:$0010F0,20	M828->Y:$0010F0,20	M844->X:$0010F1,20
M813->X:$0010F0,21	M829->Y:$0010F0,21	M845->X:$0010F1,21
M814->X:$0010F0,22	M830->Y:$0010F0,22	M846->X:$0010F1,22
M815->X:$0010F0,23	M831->Y:$0010F0,23	M847->X:$0010F1,23

IO word #4	IO Word #5	IO Word #6
M900->Y:$0010F1,8	M916->X:$0010F2,8	M932->Y:$0010F2,8
M901->Y:$0010F1,9	M917->X:$0010F2,9	M933->Y:$0010F2,9
M902->Y:$0010F1,10	M918->X:$0010F2,10	M934->Y:$0010F2,10
M903->Y:$0010F1,11	M919->X:$0010F2,11	M935->Y:$0010F2,11
M904->Y:$0010F1,12	M920->X:$0010F2,12	M936->Y:$0010F2,12
M905->Y:$0010F1,13	M129->X:$0010F2,13	M937->Y:$0010F2,13
M906->Y:$0010F1,14	M922->X:$0010F2,14	M938->Y:$0010F2,14
M907->Y:$0010F1,15	M923->X:$0010F2,15	M939->Y:$0010F2,15
M908->Y:$0010F1,16	M924->X:$0010F2,16	M940->Y:$0010F2,16
M909->Y:$0010F1,17	M925->X:$0010F2,17	M941->Y:$0010F2,17
M910->Y:$0010F1,18	M926->X:$0010F2,18	M942->Y:$0010F2,18
M911->Y:$0010F1,19	M927->X:$0010F2,19	M943->Y:$0010F2,19
M912->Y:$0010F1,20	M928->X:$0010F2,20	M944->Y:$0010F2,20
M913->Y:$0010F1,21	M129->X:$0010F2,21	M945->Y:$0010F2,21
M914->Y:$0010F1,22	M930->X:$0010F2,22	M946->Y:$0010F2,22
M915->Y:$0010F1,23	M931->X:$0010F2,23	M947->Y:$0010F2,23

Example 1: 48 Inputs 48 Outputs Using 1×24-Bit Transfers

For this example, the inputs and outputs are not sharing the same node transfer address ($C0A0,$C0A4,$C0A8, $C0AC). Each of the node transfer addresses can be defined as 24-bit addresses.

Ultralite (8 Axis)	Turbo Ultralite (8 Axis)	Description
I996=$0FB3FF	I6841=$0FB3FF	Enable nodes 0,1,2,3,4,5,6,7,8,9,12, & 13 at PMAC Ultralite
M970->X:$C0A0,0,24	M970->X:$78420,0,24	IO word #1, 24 bit word node2
M971->X:$C0A4,0,24	M971->X:$78424,0,24	IO word #2, 24 bit word node 3
M972->X:$C0A8,0,24	M972->X:$78428,0,24	IO word #3, 24 bit word node 6
M973->X:$C0AC,0,24	M973->X:$7842C,0,24	IO word #1, 24 bit word node 7
M1000->X:$0770,0,24	M1000->X:$0010F0,0,24	Input mirror word #1
M1001->Y:$0770,0,24	M1001->Y:$0010F0,0,24	Input mirror word #2
M1002->X:$0771,0,24	M1002->X:$0010F1,0,24	Output mirror word #1
M1003->Y:$0771,0,24	M1003->Y:$0010F1,0,24	Output mirror word #2
M1010->X:$0772,0,24	M1010->X:$0010F2,0,24	Old Input mirror word #2
M1011->Y:$0772,0,24	M1011->Y:$0010F2,0,24	Old Input mirror word #3

```
MS0,MI71=$20C0A0218800  ;sets up macro to transfer data for two

;ACC-14E's
MS0,MI975=$CC           ;enable node 2, 3, 6, and 7 for I/O at MACRO
                        ;Station
MS0,MI19=4              ;sets interrupt period for data transfer
MSSAVE0                 ;save to macro station
MS$$$0                  ;reset macro station to enable

OPEN PLC1 CLEAR

M1000=M970               new input mirror equal to actual input word
M1001=M971               new input mirror equal to actual input word

IF (M1000 != M1010)
OR (M1001 != M1011)     if inputs change, process outputs

M1010 = M1000           old input mirror equal to new input mirror
M1011 = M1001           old input mirror equal to new input mirror
      .
      .
      .
      .                 } Set outputs based on inputs or program logic
      .
      .
M973 = M1002            Output word equals Output Mirror Word
M974 = M1003            Output word equals Output Mirror Word

ENDIF
CLOSE
```

Example 2: 48 Inputs 48 Outputs Using 3×16-Bit Transfers

For this example, the inputs and outputs are not sharing the same node transfer address ($C0A1,$C0A2,$C0A3, $C0A5, $C0A6, and $C0A7). Each of the node transfer addresses can be defined as 16-bit addresses.

Ultralite (8 Axis)	Turbo Ultralite (8 Axis)	Description
I996=$0FB33F	I6841=$0FB33F	Enable nodes 0,1,2,3,4,5,8,9,12, & 13 at PMAC Ultralite
M980->X:$C0A1,8,16	M980->X:$78421,8,16	IO word #1, 1st 16 bit word node2
M981->X:$C0A2,8,16	M981->X:$78422,8,16	IO word #2, 2nd 16 bit word node 2
M982->X:$C0A3,8,16	M982->X:$78423,8,16	IO word #3, 3rd 16 bit word node 2
M983->X:$C0A5,8,16	M983->X:$78425,8,16	IO word #1, 1st 16 bit word node 3
M984->X:$C0A6,8,16	M984->X:$78426,8,16	IO word #2, 2nd 16 bit word node 3
M985->X:$C0A7,8,16	M985->X:$78427,8,16	IO word #3, 3rd 16 bit word node 3
M1000->X:$0770,8,16	M1000->X:$0010F0,8,16	Input mirror word #1
M1001->Y:$0770,8,16	M1001->Y:$0010F0,8,16	Input mirror word #2
M1002->X:$0771,8,16	M1002->X:$0010F1,8,16	Input mirror word #3
M1003->Y:$0771,8,16	M1003->Y:$0010F1,8,16	Output mirror word #1
M1004->X:$0772,8,16	M1004->X:$0010F2,8,16	Output mirror word #2
M1005->Y:$0772,8,16	M1005->Y:$0010F2,8,16	Output mirror word #3
M1010->X:$0773,8,16	M1010->X:$0010F3,8,16	Old Image mirror word #1
M1011->Y:$0773,8,16	M1011->Y:$0010F3,8,16	Old Image mirror word #2
M1012->X:$0774,8,16	M1012->X:$0010F4,8,16	Old Image mirror word #3

```
MS0,MI69=$20C0A1318800  ;sets up macro to transfer data two ACC-14E's
MS0,MI975=$C            ;enable node 2 and 3 for I/O
MS0,MI19=4              ;sets interrupt period for data transfer
MSSAVE0                 ;save to macro station
MS$$$0                  ;reset macro station to enable

OPEN PLC1 CLEAR
M1000=M980              new input mirror equal to actual input word
M1001=M981              new input mirror equal to actual input word
M1002=M982              new input mirror equal to actual input word

IF (M1000 != M1010)
OR (M1001 != M1011)     if inputs change, process outputs

M1010 = M1000           old input mirror equal to new input mirror
M1011 = M1001           old input mirror equal to new input mirror
      .
      .
      .
      .                 } Set outputs based on inputs or program logic
      .
      .
M983 = M1003            Output word equals Output Mirror Word
```

```
M984 = M1004            Output word equals Output Mirror Word
M985 = M1005            Output word equals Output Mirror Word
ENDIF
CLOSE
```

SETTING UP CONTROL WORD FOR MACRO I/O

The I/O gate array used on the UMAC I/O accessories has the ability to allow any of the 48-bits be used as an input (read) or an output (write). To protect the inputs to be read, define the individual bits as read only on a byte-by-byte basis. This is accomplished by writing to the control word of the I/O gate.

Each I/O gate has eight 8-bit words:
I/O word 0 - I/O bits 0-7
I/O word 1 - I/O bits 8-15
I/O word 2 - I/O bits 16-23
I/O word 3 - I/O bits 24-31
I/O word 4 - I/O bits 32-39
I/O word 5 - I/O bits 40-47
I/O word 6 - Data Word
I/O word 7 - Control Word

I/O words 0 through 5 contain the actual I/O data. I/O word 7 is the control word that turns any of the I/O words into read only bits. The lower six bits of the Control Word are used to tell the I/O gate whether the data in the six I/O word bytes are read only or read/write registers. For example, to make I/O word 0, I/O word 1, and I/O word 2 (bits 0-23) read only, set the IO control word equal to 7 (binary 000111).

As of MACRO firmware release 1.16, there are no MI-variables to support direct access to the I/O control words. An easy method can be used to write directly to the control word of the I/O gate using MI198 and MI199 (place the register to read or write to into MI198 and the read or write to that value using MI199). Usually, this will be done in a one time read PLC at power up.

Base Address from SW1 Setting	**Control Word Location**	**MI198 Setting**
Y:$8800	`Y:$8807,0,8`	`MI198=$408807`
Y:$9800	`Y:$9807,0,8`	`MI198=$409807`
Y:$A800	`Y:$A807,0,8`	`MI198=$40A807`
Y:$B800 ($FFE0*)	`Y:$B807,0,8`	`MI198=$40B807`
Y:$8840	`Y:$8847,0,8`	`MI198=$408847`
Y:$9840	`Y:$9847,0,8`	`MI198=$409847`
Y:$A840	`Y:$A847,0,8`	`MI198=$40A847`
Y:$B840 ($FFE8*)	`Y:$B847,0,8`	`MI198=$40B847`
Y:$8880	`Y:$8887,0,8`	`MI198=$408887`
Y:$9880	`Y:$9887,0,8`	`MI198=$409887`
Y:$A880	`Y:$A887,0,8`	`MI198=$40A887`
Y:$B880 ($FFF0*)	`Y:$B887,0,8`	`MI198=$40B887`
Y:$88C0	`Y:$88C7,0,8`	`MI198=$4088C7`
Y:$98C0	`Y:$98C7,0,8`	`MI198=$4098C7`
Y:$A8C0	`Y:$A8C7,0,8`	`MI198=$40A8C7`
Y:$B8C0	`Y:$B8C7,0,8`	`MI198=$40B8C7`
* for legacy systems		

Once the control word is defined to MI198, write to the individual bytes associated with the I/O gate and make them either read only or read/write (default).

Byte 0	Byte 1	Byte 2	Byte 3	Byte4	Byte 5
Y:$8800,0,8	Y:$8801,0,8	Y:$8802,0,8	Y:$8803,0,8	Y:$8804,0,8	Y:$8805,0,8
Y:$9800,0,8	Y:$9801,0,8	Y:$9802,0,8	Y:$9803,0,8	Y:$9804,0,8	Y:$9805,0,8

Example: MACRO Station has ACC-65E (24in/24out) and ACC-14E (48 in) set to base addresses $8800 and $9800 respectively.

```
#define Timer1   I5111       ;plc countdown timer for Turbo Ultralite
;#define Timer1  M70         ;plc countdown timer for Ultralite
;M70->X:$0700,0,24,s         ;countdown timer for non-turbo PMAC

Open PLC 10 Clear
Timer1=2000*8388608/I10           ;2 second delay to ensure MACRO
While (Timer1>0)Endwhile          ;Station is powered up properly

CMD"MS0,MI198=$408807"            ;set control word for ACC-65E
CMD"MS0,MI199=$07"                ;write $07 into Y:$8807 (control word)
Timer1=50*8388608/I10             ;50 msec delay
While (Timer1>0) Endwhile

CMD"MS0,MI198=$409807"            ;set control word for ACC-14E
CMD"MS0,MI199=$3F"                ;write $3F into Y:$9807 (control word)
Timer1=50*8388608/I10             ;50 msec delay
While (Timer1>0) Endwhile

Disable PLC10

Close
```

ACC-14E FEEDBACK SETUP FOR MACRO

If using the information from the ACC-14E converter for closed loop servo data, you could process the data at the encoder conversion table at the MACRO Station and have the information sent to the Ultralite automatically. The data from the ACC-14E is processed as a parallel word input at the MACRO Station and then transmitted back to the Ultralite using the traditional Servo Node. The encoder conversion table at the MACRO Station will have to be modified to process this data. From the Ultralite standpoint, nothing will need to be modified to read the position and velocity data.

The data being processed at the ACC-14E is processed at the base address + 0 for port A and base address + 3 for port B. Below are the possibilities for CS10.

Base Address based on SW1	Port A Feedback Address	Port B Feedback Address
$8800	$8800	$8803
$9800	$9800	$9803
$A800	$A800	$A803
$B800	$B800	$B803

At the MACRO Station, the encoder conversion table is located at memory locations $0010 through $002F:

X:$0010 - $002F Converter Encoder & Time Base
Y:$0010 - $002F Source and Format

MACRO Station Variables MI120 through MI151 are memory mapped to these locations for easy conversion table setup. The data is transferred to the Ultralite automatically via MACRO Station variable MI10x, where x stands for the encoder/feedback channel. The Ultralite will then read the information sent from the MACRO-Station as a parallel word (default at the Ultralite).

For example, if we had the following entry:

```
MS0,MI120=$248800   ($10 of ECT)
MS0,MI121=$FFFFFF   ($11 of ECT)
```

The output from the ECT for this encoder will reside in X:$11 and MI10x will be set to this value.

Since the ACC-14E data is typically absolute, the data can be sent at the Ultralite also as absolute data for correct position at power-up. This is accomplished with the proper setup of MSn,MI11x at the MACRO Station, and Ix10 at the Ultralite or Ix10 and Ix95 with the Turbo Ultralite. Regardless of the type of Ultralite, retrieving the power-on-position is the same. The information must be retrieved from MACRO Station variable MSn,MI920 for each node transfer as specified by Ix10 at the Ultralite. MSn,MI920 does not have to set up because the MACRO Station will place the power-on position the appropriate register at power-up.

Power-On Feedback Address for PMAC2 Ultralite

Both the Ultralite and the Turbo Ultralite obtains absolute position at power up or upon request (#n$*). The Ultralite must have Ix10 set up and the Turbo Ultralite needs both Ixx10 and Ixx95 set up to enable this power on position function. For power on position reads as specified in this document, MACRO firmware version 1.114 or newer is needed. The Turbo Ultralite firmware must be 1.936 or newer, and lastly the standard Ultralite must have firmware version 1.16H or newer.

Ix10 permits an automatic read of an absolute position sensor at power-on/reset. If Ix10 is set to 0, the power-on/reset position for the motor will be considered to be 0, regardless of the type of sensor used.

There are specific settings of PMAC's/PMAC2's Ix10 for each type of MACRO interface. The Compact MACRO Station has a corresponding variable I11x for each node that must be set.

Absolute Position for Ultralite

Compact MACRO Station Feedback Type (Firmware version 1.16H and above)	Ix10 (Unsigned)	Ix10 (Signed)
ACC-8D Opt 7 Resolver/Digital Converter	$73000n	$F3000n
ACC-8D Opt 9 Yaskawa Absolute Encoder Converter	$72000n	$F2000n
ACC-8D Opt 10 Sanyo Absolute Encoder Converter	$74000n	$F4000n
ACC-28B or ACC-28E Analog/Digital Converter	$74000n	$F4000n
MACRO Station Option 1C/ACC-6E A/D Converter	$74000n	$F4000n
MACRO Station Parallel Input	$74000n	$F4000n
MACRO Station MLDT Input	$74000n	$F4000n

n is the MACRO node number used for Motor x: 0, 1, 4, 5, 8, 9, C(12), or D(13).

(Ixx95=$720000 - $740000, $F20000 - $F40000)

Addresses are MACRO Node Numbers

MACRO Node Number	Ixx10 for MACRO IC 0	Ixx10 for MACRO IC 1	Ixx10 for MACRO IC 2	Ixx10 for MACRO IC 3
0	$000100	$000010	$000020	$000030
1	$000001	$000011	$000021	$000031
4	$000004	$000014	$000024	$000034
5	$000005	$000015	$000025	$000035
8	$000008	$000018	$000028	$000038
9	$000009	$000019	$000029	$000039
12	$00000C	$00001C	$00002C	$00003C
13	$00000D	$00001D	$00002D	$00003D

Compact MACRO Station Feedback Type	Ixx95 (Unsigned)	Ixx95 (Signed)
ACC-8D Opt 7 Resolver/Digital Converter	$730000	$F30000
ACC-8D Opt 9 Yaskawa Absolute Encoder Converter	$720000	$F20000
ACC-8D Opt 10 Sanyo Absolute Encoder Converter	$740000	$F40000
ACC-28B Analog/Digital Converter	$740000	$F40000
MACRO Station Option 1C/ACC-6E A/D Converter	$740000	$F40000
MACRO Station Parallel Input, MLDT, SSI	$740000	$F40000

When PMAC or PMAC2 has Ix10 set to get absolute position over MACRO, it executes a station auxiliary read command MS{node},I920 to request the absolute position from the Compact MACRO Station. The station then references its own I11x value to determine the type, format, and address of the data to be read.

MACRO Absolute Position Setup

MI111 through MI118 (MI11*x*) specify whether, where, and how absolute position is to be read on the Compact MACRO Station for a motor node (MI11*x* controls the *x*th motor node, which usually corresponds to Motor *x* on PMAC) and sent back to the Ultralite.

If MI11*x* is set to 0, no power-on reset absolute position value will be returned to PMAC. If MI11*x* is set to a value greater than 0, then when the PMAC requests the absolute position because its Ix10 and/or Ix81 values are set to obtain absolute position through MACRO (sending an auxiliary **`MS{node},MI920`** command), the Compact MACRO Station will use MI11*x* to determine how to read the absolute position, and report that position back to PMAC as an auxiliary response.

For the ACC-14E, the absolute data from the I/O Gate can be processed as a triple-Byte Parallel (24 bits) in low bytes of 24-bit words or as a double-Byte Parallel (16 bits) in low bytes of 16-bit words.

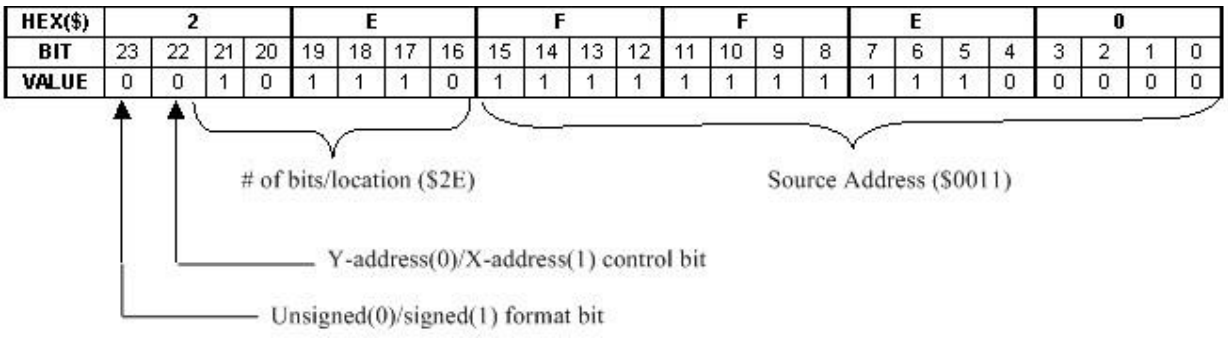
The following table shows the possible values for MI11*x*, organized by the first two digits:

MI11x Bits 16-23 for Unsigned (Signed)	**Type of Feedback**	**Notes**
$2B ($AB)	Double-Byte Parallel (16 bits) in low bytes of 24-bit words	Used for ACC-3E & ACC-14E parallel feedback; Most significant byte is at {address + 1}
$2E ($AE)	Triple-Byte Parallel (24 bits) in low bytes of 24-bit words	Used for ACC-3E & ACC-14E parallel feedback; Middle byte is at {address + 1}; Most significant byte is at {address + 2}

MI11*x* consists of two parts. The low 16 bits (last four hexadecimal digits) specify the address on the MACRO Station from which the absolute position information is read. The high eight bits (first two hexadecimal digits) tell the Compact MACRO Station how to interpret the data at that address (the method).

MACRO MI11x Parallel Word Example:

Signed 24-bit Absolute data from ACC-14E at $FFE0



X/Y Address Bit

If bit 22 of Ix10 is 0, the PMAC looks for the parallel sensor in its Y address space. The is the standard choice, since all I/O ports map into the Y address space. If this bit is 1, PMAC looks for the parallel sensor in its X address space.

If the most significant bit (MSB -- bit 23) of MI11x is 0, the value read from the absolute sensor is treated as an unsigned quantity. If the MSB is 1, which adds $80 to the high eight bits of MI11x, the value read from the sensor is treated as a signed, two's-complement quantity.

Summarizing the format of the variable through the example, which specifies a 24-bit parallel word at Y:$FFE0, treated as a signed quantity, write:

```
MS0,MI111=$2EFFE0      ;read signed 24-bit absolute power on position
                       ;from Y:$FFE0
```

ACC-14E Greyscale Decode with MACRO

As of MACRO firmware release 1.15, there are no MI-variables to support the greyscale read feature. To activate the greyscale read feature, write to the control word of the I/O gate using MI198 and MI199 (place the register to read or write to into MI198 and the read or write to that value using MI199). To access the greyscale read, write $80 to the control word. Usually this will be done in a one time read PLC at power up.

Base Address from SW1 Setting	Control Word Location	MI198 Setting
Y:$8800	Y:$8807,0,8	MI198=$408807
Y:$9800	Y:$980E,0,8	MI198=$40980E
Y:$A800	Y:$A807,0,8	MI198=$40A807
Y:$B800	Y:$B807,0,8	MI198=$40B807

Once the control word is defined to MI198, write to the individual bytes associated with the I/O gate. To enable greyscale reads, write $FF to each of the bytes associated with the IO gate.

Byte 0	Byte 1	Byte 2	Byte 3	Byte4	Byte 5
Y:$8800,0,8	Y:$8801,0,8	Y:$8802,0,8	Y:$8803,0,8	Y:$8804,0,8	Y:$8805,0,8
Y:$9800,0,8	Y:$9809,0,8	Y:$9802,0,8	Y:$980B,0,8	Y:$980C,0,8	Y:$980D,0,8
Y:$9800,0,8	Y:$A801,0,8	Y:$A802,0,8	Y:$A803,0,8	Y:$A804,0,8	Y:$A805,0,8
Y:$B800,0,8	Y:$B801,0,8	Y:$B802,0,8	Y:$B803,0,8	Y:$B804,0,8	Y:$B805,0,8

Example:

```
#define Timer1   I5111       ;plc countdown timer for Turbo Ultralite
;#define Timer1  M70         ;plc countdown timer for Ultralite
;M70->X:$0700,0,24,s         ;countdown timer for non-turbo PMAC

Open PLC 10 Clear
Timer1=2000*8388608/I10      ;2 second delay to ensure MACRO
While (Timer1>0)Endwhile     ;Station is powered up properly

CMD"MS0,MI198=$408807"
CMD"MS0,MI199=$80"           ;write $80 into Y:$8807 (control word)
Timer1=50*8388608/I10        ;50 msec delay
While (Timer1>0) Endwhile
```

```
CMD"MS0,MI198=$408800"
CMD"MS0,MI199=$FF"          ;write $FF into Y:$8800 (byte0)
Timer1=50*8388608/I10       ;50 msec delay
While (Timer1>0) Endwhile

CMD"MS0,MI198=$408801"
CMD"MS0,MI199=$FF"          ;write $FF into Y:$8801 (byte1)
Timer1=50*8388608/I10       ;50 msec delay
While (Timer1>0) Endwhile
CMD"MS0,MI198=$408802"
CMD"MS0,MI199=$FF"          ;write $FF into Y:$8802 (byte2)
Timer1=50*8388608/I10       ;50 msec delay
While (Timer1>0) Endwhile

CMD"MS0,MI198=$408803"
CMD"MS0,MI199=$FF"          ;write $FF into Y:$8803 (byte3)
Timer1=50*8388608/I10       ;50 msec delay
While (Timer1>0) Endwhile

CMD"MS0,MI198=$408804"
CMD"MS0,MI199=$FF"          ;write $FF into Y:$8804 (byte4)
Timer1=50*8388608/I10       ;50 msec delay
While (Timer1>0) Endwhile

CMD"MS0,MI198=$408805"
CMD"MS0,MI199=$FF"          ;write $FF into Y:$8805 (byte5)
Timer1=50*8388608/I10       ;50 msec delay
While (Timer1>0) Endwhile

CMD"MS0,MI198=$408807"
CMD"MS0,MI199=$3F"          ;write $3F into Y:$8807 (control word)
Timer1=50*8388608/I10       ;50 msec delay
While (Timer1>0) Endwhile

Disable PLC10

Close
```

The last command for this PLC wrote $3F into the control word. This action forces the inputs to be read only.

ACC-14E SETUP FOR CLOSED LOOP CONTROL WITH MACRO

The following is an example of the MACRO Station MACRO setup:

The S1 settings for this example are for the CS10 ($8800) selection.

1. Set jumper E1 to 1-2, Extended address mode
2. Set Jumper E2 to 2-3 for phase clock
3. Set Encoder Conversion Table at the MACRO Station (MI120-MI151)

```
MS0,MI120=$248800   ;process as parallel y word  ($10 station
                    ;address) for ACC-14E #1 port A
MS0,MI121=$FFFFFF   ;process all 24 bits ($11 station address) for
                    ENC #1
MS0,MI122=$248803   ;process as parallel y word  ($12 station
                    address) for ACC-14E #1 port B
MS0,MI123=$FFFFFF   ;process all 24 bits ($13 station address) for
                    ENC #2
```

4. Set Node Transfer Variables (MI101-MI108)

```
MS0,MI101=$11       ;processed for ECT for ENC #1
MS0,MI102=$13       ;processed for ECT for ENC #2
```

5. Set Absolute Power on Read (MI111-MI118)

```
MS0,M111=$2E8800    ; power on read from ECT #1
MS0,M112=$2E8803    ; power on read from ECT #2
```

6. Set Ix10 as specified by the appropriate Ultralite:

Axis	Ultralite Signed	Ultralite Unsigned	Turbo Ultralite Node address
1	$F30000	$730000	$000100
2	$F30001	$730001	$000001

7. If Turbo PMAC Ultralite, Set Ix95 for power on address and type

Axis	Turbo Ultralite Signed	Turbo Ultralite Unsigned
1	$F40000	$740000
2	$F40000	$740000

LEGACY MACRO SYSTEMS

The legacy systems are defined as MACRO CPU with the following part numbers:

- 602804-100
- 602804-101
- 602804-102
- 602804-103
- 602804-104

These systems do not have the extended addressing of the newer model MACRO CPUs (602804-105 through 602804-109).

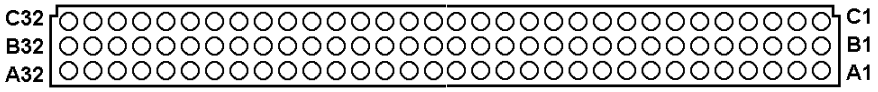
To use the older hardware address settings the table below specifies the settings required.

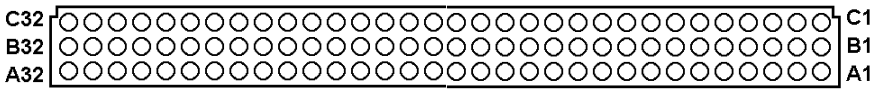
Chip Select	MACRO Address	DIP Switch SW1 Position					
		6	5	4	3	2	1
CS 10	$FFE0	Open	Open	Open	Open	Close	Close
CS 12	$FFE8	Open	Open	Open	Open	Close	Open
CS 14	$FFF0	Open	Open	Open	Open	Open	Close
CS 16	Cannot Use	Open	Open	Open	Open	Open	Open

To use the new I/O cards with the older firmware systems, the user can use each of the I/O transfer variables (MI69, MI70, MI71) to transfer 48-bits each. The main problem is that the older systems did not have the new extended addressing and only three I/O cards per MACRO station can be used.

- For systems with only one I/O card, do not change anything
- If any of these new I/O cards are used with the ACC-9E, ACC-10E, ACC-11E, or ACC-12E, then address the new I/O card as the first card (LOW byte) in addressing scheme.

ACC-14E PINOUTS

P1 UBUS (96-Pin Header)	 Front View		
Pin #	**Row A**	**Row B**	**Row C**
1	+5Vdc	+5Vdc	+5Vdc
2	GND	GND	GND
3	BD01	DAT0	BD00
4	BD03	SEL0	BD02
5	BD05	DAT1	BD04
6	BD07	SEL1	BD06
7	BD09	DAT2	BD08
8	BD11	SEL2	BD10
9	BD13	DAT3	BD12
10	BD15	SEL3	BD14
11	BD17	DAT4	BD16
12	BD19	SEL4	BD18
13	BD21	DAT5	BD20
14	BD23	SEL5	BD22
15	BS1	DAT6	BS0
16	BA01	SEL6	BA00
17	BA03	DAT7	BA02
18	BX/Y	SEL7	BA04
19	CS3-	BA06	CS2-
20	BA05	BA07	CS4-
21	CS12-	BA08	CS10-
22	CS16-	BA09	CS14-
23	BA13	BA10	BA12
24	BRD-	BA11	BWR-
25	BS3	MEMCS0-	BS2
26	WAIT-	MEMCS1-	RESET
27	PHASE+	IREQ1-	SERVO+
28	PHASE-	IREQ2-	SERVO-
29	ANALOG	GND IREQ3-	ANALOG GND
30	-15Vdc	PWRGND	+15Vdc

P1 UBUS (96-Pin Header)	 Front View		
31	GND	GND	GND
32	+5Vdc	+5Vdc	+5Vdc
For more details about the JEXP, see the UBUS Specification Document.			

I/O Terminal - J4 (50 Pin Header)

Pin	Symbol	Function	Description
1	MI/O23	In/Out	I/O at Base Address Bit 23
2	GND	Common	PMAC Common
3	MI/O22	In/Out	I/O at Base Address Bit 22
4	GND	Common	PMAC Common
5	MI/O21	In/Out	I/O at Base Address Bit 21
6	GND	Common	PMAC Common
7	MI/O20	In/Out	I/O at Base Address Bit 20
8	GND	Common	PMAC Common
9	MI/O19	In/Out	I/O at Base Address Bit 19
10	GND	Common	PMAC Common
11	MI/O18	In/Out	I/O at Base Address Bit 18
12	GND	Common	PMAC Common
13	MI/O17	In/Out	I/O at Base Address Bit 17
14	GND	Common	PMAC Common
15	MI/O16	In/Out	I/O at Base Address Bit 16
16	GND	Common	PMAC Common
17	MI/O15	In/Out	I/O at Base Address Bit 15
18	GND	Common	PMAC Common
19	MI/O14	In/Out	I/O at Base Address Bit 14
20	GND	Common	PMAC Common
21	MI/O13	In/Out	I/O at Base Address Bit 13
22	GND	Common	PMAC Common
23	MI/O12	In/Out	I/O at Base Address Bit 12
24	GND	Common	PMAC Common
25	MI/O11	In/Out	I/O at Base Address Bit 11
26	GND	Common	PMAC Common
27	MI/O10	In/Out	I/O at Base Address Bit 10
28	GND	Common	PMAC Common
29	MI/O9	In/Out	I/O at Base Address Bit 9
30	GND	Common	PMAC Common
31	MI/O8	In/Out	I/O at Base Address Bit 8
32	GND	Common	PMAC Common
33	MI/O7	In/Out	I/O at Base Address Bit 7
34	GND	Common	PMAC Common

Pin	Symbol	Function	Description
35	MI/O6	In/Out	I/O at Base Address Bit 6
36	GND	Common	PMAC Common
37	MI/O5	In/Out	I/O at Base Address Bit 5
38	GND	Common	PMAC Common
39	MI/O4	In/Out	I/O at Base Address Bit 4
40	GND	Common	PMAC Common
41	MI/O3	In/Out	I/O at Base Address Bit 3
42	GND	Common	PMAC Common
43	MI/O2	In/Out	I/O at Base Address Bit 2
44	ERR1	Input	Error Signal
45	MI/O1	In/Out	I/O at Base Address Bit 1
46	ICLK1	Input	
47	MI/O0	In/Out	I/O at Base Address Bit 0
48	OCLK1	Output	
49	+V	Output	5V Power
50	GND	Common	

I/O Terminal - J5 (50 Pin Header)

Pin	Symbol	Function	Description
1	MI/O47	In/Out	I/O at Base Address Bit 47
2	GND	Common	PMAC Common
3	MI/O46	In/Out	I/O at Base Address Bit 46
4	GND	Common	PMAC Common
5	MI/O45	In/Out	I/O at Base Address Bit 45
6	GND	Common	PMAC Common
7	MI/O44	In/Out	I/O at Base Address Bit 44
8	GND	Common	PMAC Common
9	MI/O43	In/Out	I/O at Base Address Bit 43
10	GND	Common	PMAC Common
11	MI/O42	In/Out	I/O at Base Address Bit 42
12	GND	Common	PMAC Common
13	MI/O41	In/Out	I/O at Base Address Bit 41
14	GND	Common	PMAC Common
15	MI/O40	In/Out	I/O at Base Address Bit 40
16	GND	Common	PMAC Common
17	MI/O39	In/Out	I/O at Base Address Bit 39
18	GND	Common	PMAC Common
19	MI/O38	In/Out	I/O at Base Address Bit 38
20	GND	Common	PMAC Common
21	MI/O37	In/Out	I/O at Base Address Bit 37
22	GND	Common	PMAC Common
23	MI/O36	In/Out	I/O at Base Address Bit 36
24	GND	Common	PMAC Common
25	MI/O35	In/Out	I/O at Base Address Bit 35
26	GND	Common	PMAC Common
27	MI/O34	In/Out	I/O at Base Address Bit 34
28	GND	Common	PMAC Common
29	MI/O33	In/Out	I/O at Base Address Bit 33
30	GND	Common	PMAC Common
31	MI/O32	In/Out	I/O at Base Address Bit 32
32	GND	Common	PMAC Common
33	MI/O31	In/Out	I/O at Base Address Bit 31

Pin	Symbol	Function	Description
34	GND	Common	PMAC Common
35	MI/O30	In/Out	I/O at Base Address Bit 30
36	GND	Common	PMAC Common
37	MI/O29	In/Out	I/O at Base Address Bit 29
38	GND	Common	PMAC Common
39	MI/O28	In/Out	I/O at Base Address Bit 28
40	GND	Common	PMAC Common
41	MI/O27	In/Out	I/O at Base Address Bit 27
42	GND	Common	PMAC Common
43	MI/O26	In/Out	I/O at Base Address Bit 26
44	ERR2	Input	Error Signal
45	MI/O25	In/Out	I/O at Base Address Bit 25
46	ICLK2	Input	
47	MI/O24	In/Out	I/O at Base Address Bit 24
48	OCLK2	Output	
49	+V	Output	5V
50	GND	Common	

I/O Terminal - J4 (50 Pin DSUB)

Pin	Symbol	Function	Description
1	MI/O23	In/Out	I/O at Base Address Bit 23
2	MI/O22	In/Out	I/O at Base Address Bit 22
3	MI/O21	In/Out	I/O at Base Address Bit 21
4	MI/O20	In/Out	I/O at Base Address Bit 20
5	MI/O19	In/Out	I/O at Base Address Bit 19
6	MI/O18	In/Out	I/O at Base Address Bit 18
7	MI/O17	In/Out	I/O at Base Address Bit 17
8	MI/O16	In/Out	I/O at Base Address Bit 16
9	MI/O15	In/Out	I/O at Base Address Bit 15
10	MI/O14	In/Out	I/O at Base Address Bit 14
11	MI/O13	In/Out	I/O at Base Address Bit 13
12	MI/O12	In/Out	I/O at Base Address Bit 12
13	MI/O11	In/Out	I/O at Base Address Bit 11
14	MI/O10	In/Out	I/O at Base Address Bit 10
15	MI/O9	In/Out	I/O at Base Address Bit 9
16	MI/O8	In/Out	I/O at Base Address Bit 8
17	MI/O7	In/Out	I/O at Base Address Bit 7
18	MI/O6	In/Out	I/O at Base Address Bit 6
19	MI/O5	In/Out	I/O at Base Address Bit 5
20	MI/O4	In/Out	I/O at Base Address Bit 4
21	MI/O3	In/Out	I/O at Base Address Bit 3
22	MI/O2	In/Out	I/O at Base Address Bit 2
23	MI/O1	In/Out	I/O at Base Address Bit 1
24	MI/O0	In/Out	I/O at Base Address Bit 0
25	+V	Output	5V Power
26	GND	Common	PMAC Common
27	GND	Common	PMAC Common
28	GND	Common	PMAC Common
29	GND	Common	PMAC Common
30	GND	Common	PMAC Common
31	GND	Common	PMAC Common
32	GND	Common	PMAC Common
33	GND	Common	PMAC Common
34	GND	Common	PMAC Common

Pin	Symbol	Function	Description
35	GND	Common	PMAC Common
36	GND	Common	PMAC Common
37	GND	Common	PMAC Common
38	GND	Common	PMAC Common
39	GND	Common	PMAC Common
40	GND	Common	PMAC Common
41	GND	Common	PMAC Common
42	GND	Common	PMAC Common
43	GND	Common	PMAC Common
44	GND	Common	PMAC Common
45	GND	Common	PMAC Common
46	GND	Common	PMAC Common
47	ERR1	Input	Error Signal
48	ICLK1	Input	
49	OCLK1	Output	
50	GND	Common	

I/O Terminal - J5 (50 Pin DSUB)

Pin	Symbol	Function	Description
1	MI/O47	In/Out	I/O at Base Address Bit 47
2	MI/O46	In/Out	I/O at Base Address Bit 46
3	MI/O45	In/Out	I/O at Base Address Bit 45
4	MI/O44	In/Out	I/O at Base Address Bit 44
5	MI/O43	In/Out	I/O at Base Address Bit 43
6	MI/O42	In/Out	I/O at Base Address Bit 42
7	MI/O41	In/Out	I/O at Base Address Bit 41
8	MI/O40	In/Out	I/O at Base Address Bit 40
9	MI/O39	In/Out	I/O at Base Address Bit 39
10	MI/O38	In/Out	I/O at Base Address Bit 38
11	MI/O37	In/Out	I/O at Base Address Bit 37
12	MI/O36	In/Out	I/O at Base Address Bit 36
13	MI/O35	In/Out	I/O at Base Address Bit 35
14	MI/O34	In/Out	I/O at Base Address Bit 34
15	MI/O33	In/Out	I/O at Base Address Bit 33
16	MI/O32	In/Out	I/O at Base Address Bit 32
17	MI/O31	In/Out	I/O at Base Address Bit 31
18	MI/O30	In/Out	I/O at Base Address Bit 30
19	MI/O29	In/Out	I/O at Base Address Bit 29
20	MI/O28	In/Out	I/O at Base Address Bit 28
21	MI/O27	In/Out	I/O at Base Address Bit 27
22	MI/O26	In/Out	I/O at Base Address Bit 26
23	MI/O25	In/Out	I/O at Base Address Bit 25
24	MI/O24	In/Out	I/O at Base Address Bit 24
25	+V	Output	5V
26	GND	Common	PMAC Common
27	GND	Common	PMAC Common
28	GND	Common	PMAC Common
29	GND	Common	PMAC Common
30	GND	Common	PMAC Common
31	GND	Common	PMAC Common
32	GND	Common	PMAC Common
33	GND	Common	PMAC Common
34	GND	Common	PMAC Common

Pin	Symbol	Function	Description
35	GND	Common	PMAC Common
36	GND	Common	PMAC Common
37	GND	Common	PMAC Common
38	GND	Common	PMAC Common
39	GND	Common	PMAC Common
40	GND	Common	PMAC Common
41	GND	Common	PMAC Common
42	GND	Common	PMAC Common
43	GND	Common	PMAC Common
44	GND	Common	PMAC Common
45	GND	Common	PMAC Common
46	GND	Common	PMAC Common
47	ERR2	Input	Error Signal
48	ICLK2	Input	
49	OCLK2	Output	
50	GND	Common	