

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

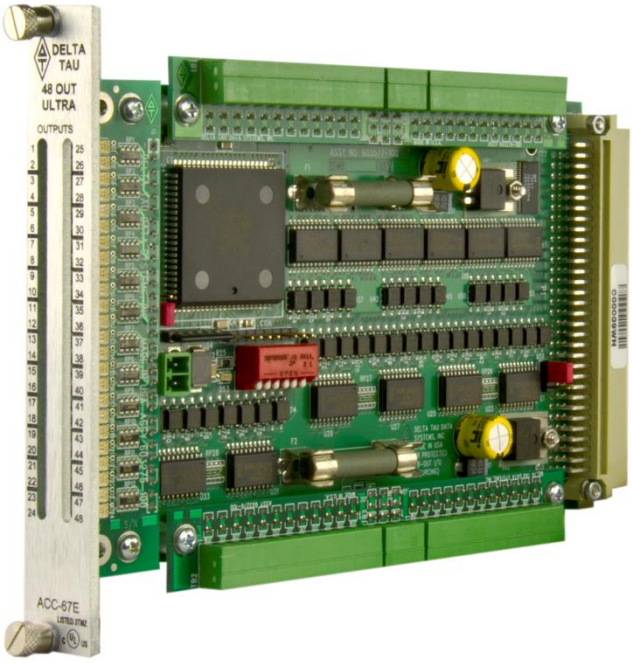
Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *OMRON / Delta Tau ACC-67E* at our website: ***Click HERE***

USER MANUAL

Accessory 67E



Digital I/O – 48 Sourcing Outputs

3Ax-603577-xUxx

February 14, 2015

DELTA TAU
Data Systems, Inc.

NEW IDEAS IN MOTION …

21314 Lassen St. Chatsworth, CA 91311 // Tel. (818) 998-2095 Fax. (818) 998-7807 // www.deltatau.com

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

Copyright Information

© 2015 Delta Tau Data Systems, Inc. All rights reserved.

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

Delta Tau Data Systems, Inc. Technical Support
Phone: (818) 717-5656
Fax: (818) 998-7807
Email: support@deltatau.com
Website: http://www.deltatau.com

Operating Conditions

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

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



WARNING

A Warning identifies hazards that could result in personal injury or death. It precedes the discussion of interest.



Caution

A Caution identifies hazards that could result in equipment damage. It precedes the discussion of interest.



Note

A Note identifies information critical to the understanding or use of the equipment. It follows the discussion of interest.

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

REVISION HISTORY				
REV.	**DESCRIPTION**	**DATE**	**CHG**	**APPVD**
1	Updated Max Power Output	01/29/08	CP	SM
2	Added UL Seal to Cover Page	10/01/09	CP	SF
3	Added Power PMAC/MACRO Reformatted manual	02/24/14	GS	RN
4	Corrected DIP Switch Settings Table	02/03/15	RN	RN

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

This page left blank intentionally

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

Table of Contents

INTRODUCTION **7**

SPECIFICATIONS **8**
Environmental Specifications 8
Electrical Specifications 8
Power Requirements *8*
Fuses *8*
Output Drivers 8
Physical Specifications 9
Terminal Block Layout *9*
D-Sub Layout *9*
Agency Approval and Safety 10

ADDRESSING THE ACC-67E **11**
Address Select DIP Switch SW1 11
Legacy MACRO Dip Switch Settings 12
Hardware Address Limitations 12

USING THE ACC-67E WITH POWER UMAC **14**
Declaring Pointers to I/O Structure Elements 14

USING THE ACC-67E WITH TURBO UMAC **15**
Assigning M-Variables to I/O Memory Locations 15

USING THE ACC-67E IN A MACRO STATION **16**
MACRO16 I/O Node Addressing 17
Ring Controller I/O Node Addressing 19
Turbo PMAC2 I/O Node Addressing *19*
Power PMAC2 I/O Node Addressing *20*
Power PMAC3 I/O Node Addressing *22*
Configuring MACRO I/O Transfers 24
MS{anynode},MI71 ..... *25*
MS{anynode},MI69/MI70 ..... *29*
Accessing the Transferred Data 32
Outputs Mirror Image Concept *32*
Turbo PMAC2 MI71 Mapping Example *33*
Turbo PMAC2 MI69/70 Mapping Example *35*
Power PMAC2 MI71 Mapping Example *37*
Power PMAC2 MI69/70 Mapping Example *39*
Power PMAC3 MI71 Mapping Example *41*
Power PMAC3 MI69/70 Mapping Example *42*

CONNECTOR PINOUTS AND WIRING **43**
Terminal Block Connectors 43
Top: Outputs *43*

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

Bottom: Outputs *44*
TB Wiring Diagram *45*
D-Sub Connectors 46
Top: Outputs *46*
Bottom: Outputs *46*
D-Sub Wiring diagram *47*
P1: UMAC Bus (UBUS) Connector 48

APPENDIX A: E-POINT JUMPERS 49

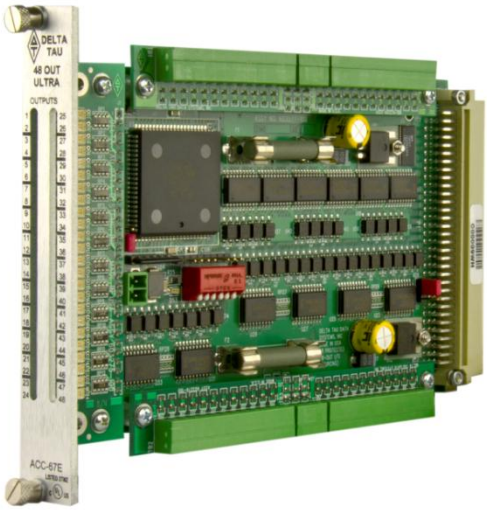
APPENDIX B: SCHEMATICS 50

APPENDIX C: USING ACC-67E IN THE C ENVIRONMENT 51
ACC-67E C Library *51*
Locations for the C Files *52*
Function Descriptions *53*
Example *54*

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

INTRODUCTION

The accessory 67E (ACC-67E) is a general purpose digital output card which provides 48 self-protected, optically isolated, sourcing outputs.



The ACC-67E is a 3U Euro style card intended to plug into the UMAC BUS backplane. It can be used with the:

- Turbo UMAC CPU
- Power UMAC CPU
- MACRO UMAC Station
  (MACRO16 CPU, PMAC2 or PMAC3 Style)

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

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

Electrical Specifications

Power Requirements

Whether providing the ACC-67E with power from the 3U backplane bus or externally (standalone mode) through TB1, the power requirements (± 10%) are:

+5 V @ 0.38 A (Board Logic Input)
+24 V @ 0.60 A (Per Output Pin)
+24 V @ 8.00 A (Maximum combined current for all top outputs (1-24)
+24 V @ 8.00 A (Maximum combined current for all bottom outputs (25-48)



Note

TB1 is a 2-pin 5V connector on the base board (used for troubleshooting in standalone operation only), not to be confused with the 12-pin I/O TB1 connectors.

Fuses

Manufacturer	Specification	Delta Tau Part Number	
Bussmann	125 V @ 20 A	MDA 20A	

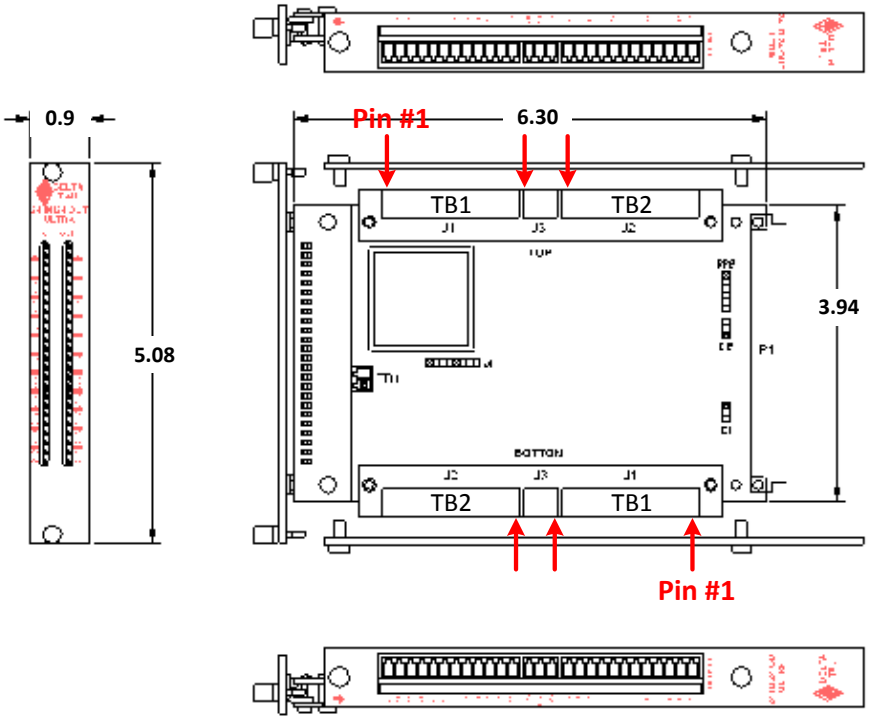
Output Drivers

The output drivers use the Diodes Inc. Zetex ZXMS6006DG chip. The current drawn from each output line should be limited to 600 mA at voltage levels between 12 VDC and 24 VDC. The total combined current draw for all outputs on the TB1/TB2 **top** terminals should be limited to 8 amps. Likewise, the total combined current draw for all outputs on the TB1/TB2 **bottom** terminals should be limited to 8 amps.

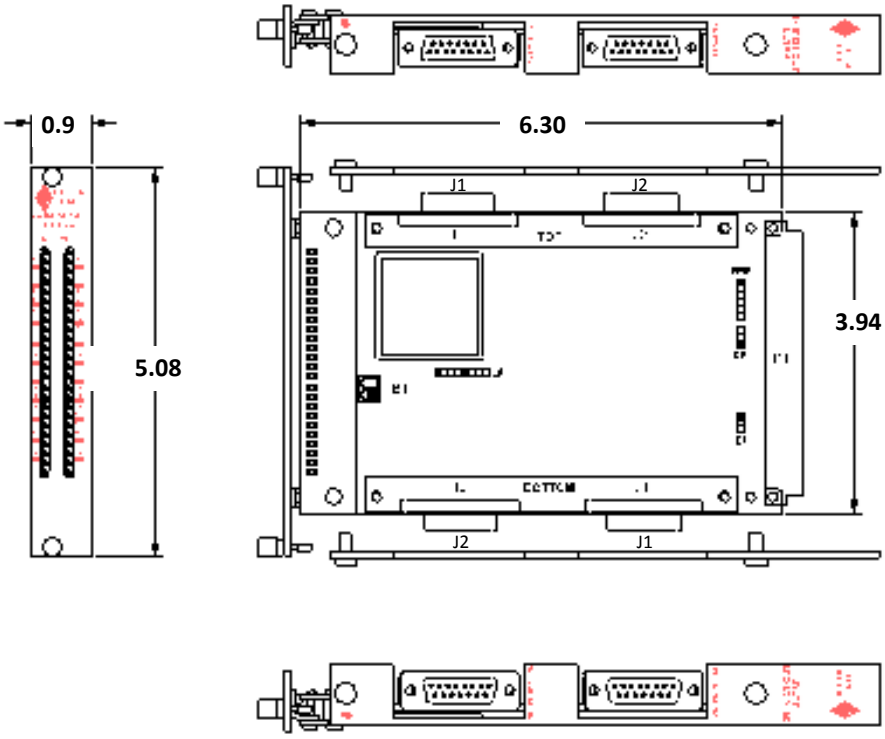
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Physical Specifications

Terminal Block Layout



D-Sub Layout



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

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 w/o Option 1A	180 g	Front Plate included
Terminal Block Connectors	FRONT-MC1,5/12-ST3,81 FRONT-MC1,5/5-ST3,81 FRONT-MC1,5/3-ST3,81	Terminal Blocks from Phoenix Contact. UL 94V-0
DB Option Connectors	DB15 Female	UL 94V-0



Note

The width is the width of the front plate. The length and height are the dimensions of the PCB. See Layout section for physical dimensions.

Agency Approval and Safety

Item	Description
UL	UL 61010-1 File E314517
cUL	CAN/CSA C22.2 No. 1010.1-92 File E314517

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

ADDRESSING THE ACC-67E

Address Select DIP Switch SW1

The switch SW1 selects the starting address location for the first I/O gate on the ACC-67E. The following table shows the dip switch settings for the Turbo, Power, and MACRO Station settings:

Chip Select	Base Address				SW1 Positions					
	TURBO	MACRO	POWER		6	5	4	3	2	1
			I/O Base Offset	Index ACC-67E[n]						
CS10	Y:$78C00	Y:$8800	$A00000	0	ON	ON	ON	ON	ON	ON
	Y:$79C00	Y:$9800	$A08000	4	ON	ON	ON	**OFF**	ON	ON
	Y:$7AC00	Y:$A800	$A10000	8	ON	ON	**OFF**	ON	ON	ON
	Y:$7BC00	Y:$B800	$A18000	12	ON	ON	**OFF**	**OFF**	ON	ON
CS12	Y:$78D00	Y:$8840	$B00000	1	ON	ON	ON	ON	ON	**OFF**
	Y:$79D00	Y:$9840	$B08000	5	ON	ON	ON	**OFF**	ON	**OFF**
	Y:$7AD00	Y:$A840	$B10000	9	ON	ON	**OFF**	ON	ON	**OFF**
	Y:$7BD00	Y:$B840	$B18000	13	ON	ON	**OFF**	**OFF**	ON	**OFF**
CS14	Y:$78E00	Y:$8880	$C00000	2	ON	ON	ON	ON	**OFF**	ON
	Y:$79E00	Y:$9880	$C08000	6	ON	ON	ON	**OFF**	**OFF**	ON
	Y:$7AE00	Y:$A880	$C10000	10	ON	ON	**OFF**	ON	**OFF**	ON
	Y:$7BE00	Y:$B880	$C18000	14	ON	ON	**OFF**	**OFF**	**OFF**	ON
CS16	Y:$78F00	Y:$88C0	$D00000	3	ON	ON	ON	ON	**OFF**	**OFF**
	Y:$79F00	Y:$98C0	$D08000	7	ON	ON	ON	**OFF**	**OFF**	**OFF**
	Y:$7AF00	Y:$A8C0	$D10000	11	ON	ON	**OFF**	ON	**OFF**	**OFF**
	Y:$7BF00	Y:$B8C0	$D18000	15	ON	ON	**OFF**	**OFF**	**OFF**	**OFF**



Note

ON designates **Closed**. **OFF** designates **Open**. Factory default is all ON.

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

Legacy MACRO Dip Switch Settings

Chip Select	Base Address (Alternate)	SW1 Positions					
		6	5	4	3	2	1
10	Y:$B800 (Y:$FFE0)	ON (**OFF**)	ON (**OFF**)	**OFF**	**OFF**	ON	ON
12	Y:$B840 (Y:$FFE8)	ON (**OFF**)	ON (**OFF**)	**OFF**	**OFF**	ON	**OFF**
14	Y:$B880 (Y:$FFF0)	ON (**OFF**)	ON (**OFF**)	**OFF**	**OFF**	**OFF**	ON
16	Y:$B8C0 (Y:$FFF8)	ON (**OFF**)	ON (**OFF**)	ON	ON	**OFF**	**OFF**



Note

The Legacy Macro base addresses are double mapped. Set SW1 positions 5 & 6 to OFF if the alternate addressing is desired.

Hardware Address Limitations

Historically, two types of accessory cards have been designed for the UMAC 3U bus type rack: type A and type B cards. They can be sorted out as follows:

Name	Type	Category	Possible Number of Base Addresses	Maximum Number of cards in 1 rack
ACC-9E	A	General I/O	4	10
ACC-10E	A	General I/O	4	
ACC-11E	A	General I/O	4	
ACC-12E	A	General I/O	4	
ACC-53E	B	Feedback	12	12
ACC-57E	B	Feedback	12	
ACC-58E	B	Feedback	12	
ACC-59E	B	Analog I/O	12	
ACC-14E	B	General I/O	16	16
ACC-28E	B	Analog I/O	16	
ACC-36E	B	Analog I/O	16	
ACC-65E	B	General I/O	16	
ACC-66E	B	General I/O	16	
ACC-67E	B	General I/O	16	
ACC-68E	B	General I/O	16	
ACC-84E	B	Feedback	12	12

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

Addressing Type A and Type B accessory cards in a UMAC or MACRO station rack requires attention to the following set of rules:

Populating Rack with Type A Cards Only (no conflicts)

In this case, the card(s) can potentially use any available Address/Chip Select.



Note

The type A cards have only one base address per chip select (CS10, CS12, and CS14). Each card can be set up (jumper settings) to use the low, middle, or high byte of a specific base address. This makes it possible to populate a single rack with 9 (3 bases x 3 byte locations) Type A accessory cards. A fourth address is available at CS16 in which only the high byte can be used, thus making the maximum 10 Type A accessory cards.

Populating Rack with Type B Cards Only (no conflicts)

In this case, the card(s) can potentially use any available Address/Chip Select.

Populating Rack with Type A & Type B Cards (possible conflicts)

Typically, Type A and Type B cards should not share the same Chip Select. If this configuration is possible, then the following rules apply:

- **Type A and Type B Feedback Cards**
  Type A cards cannot share the same base address as Type B Feedback cards.

- **Type A and Type B General I/O Cards**
- Type A cards can share base addresses with Type B general I/O cards; however, in this case, Type B cards naturally use the lower byte (default), and Type A cards must be set to the middle/high byte of the selected base address.

- **Type A Cards and Type B Analog Cards**
- Type A cards can share base addresses with Type B analog I/O cards; however, in this case, Type B cards naturally use the middle/high bytes (default), so Type A cards should be set to the low byte of the selected base address.



Note

The above conflicts only occur with the first base address in each Chip Select of the Type B cards (i.e. $78C00, $78D00, $78E00, and $78F00). Conflicts can more simply be avoided by using only the second to fourth addresses of each Chip Select.

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

USING THE ACC-67E WITH POWER UMAC

Using the ACC-67E with Power UMAC requires:

- Knowing and configuring the index (address offset) of the card
- Declaring pointers, with user given names, to the appropriate I/O structure elements

Declaring Pointers to I/O Structure Elements

Mapping an ACC-67E at index 0, for example, with user configurable pointer names:

```
PTR Output1->ACC67E[0].DataReg[0].0.1;
PTR Output2->ACC67E[0].DataReg[0].1.1;
PTR Output3->ACC67E[0].DataReg[0].2.1;
PTR Output4->ACC67E[0].DataReg[0].3.1;
PTR Output5->ACC67E[0].DataReg[0].4.1;
PTR Output6->ACC67E[0].DataReg[0].5.1;
PTR Output7->ACC67E[0].DataReg[0].6.1;
PTR Output8->ACC67E[0].DataReg[0].7.1;

PTR Output9->ACC67E[0].DataReg[1].0.1;
PTR Output10->ACC67E[0].DataReg[1].1.1;
PTR Output11->ACC67E[0].DataReg[1].2.1;
PTR Output12->ACC67E[0].DataReg[1].3.1;
PTR Output13->ACC67E[0].DataReg[1].4.1;
PTR Output14->ACC67E[0].DataReg[1].5.1;
PTR Output15->ACC67E[0].DataReg[1].6.1;
PTR Output16->ACC67E[0].DataReg[1].7.1;

PTR Output17->ACC67E[0].DataReg[2].0.1;
PTR Output18->ACC67E[0].DataReg[2].1.1;
PTR Output19->ACC67E[0].DataReg[2].2.1;
PTR Output20->ACC67E[0].DataReg[2].3.1;
PTR Output21->ACC67E[0].DataReg[2].4.1;
PTR Output22->ACC67E[0].DataReg[2].5.1;
PTR Output23->ACC67E[0].DataReg[2].6.1;
PTR Output24->ACC67E[0].DataReg[2].7.1;
```

```
PTR Output25->ACC67E[0].DataReg[3].0.1;
PTR Output26->ACC67E[0].DataReg[3].1.1;
PTR Output27->ACC67E[0].DataReg[3].2.1;
PTR Output28->ACC67E[0].DataReg[3].3.1;
PTR Output29->ACC67E[0].DataReg[3].4.1;
PTR Output30->ACC67E[0].DataReg[3].5.1;
PTR Output31->ACC67E[0].DataReg[3].6.1;
PTR Output32->ACC67E[0].DataReg[3].7.1;

PTR Output33->ACC67E[0].DataReg[4].0.1;
PTR Output34->ACC67E[0].DataReg[4].1.1;
PTR Output35->ACC67E[0].DataReg[4].2.1;
PTR Output36->ACC67E[0].DataReg[4].3.1;
PTR Output37->ACC67E[0].DataReg[4].4.1;
PTR Output38->ACC67E[0].DataReg[4].5.1;
PTR Output39->ACC67E[0].DataReg[4].6.1;
PTR Output40->ACC67E[0].DataReg[4].7.1;

PTR Output41->ACC67E[0].DataReg[5].0.1;
PTR Output42->ACC67E[0].DataReg[5].1.1;
PTR Output43->ACC67E[0].DataReg[5].2.1;
PTR Output44->ACC67E[0].DataReg[5].3.1;
PTR Output45->ACC67E[0].DataReg[5].4.1;
PTR Output46->ACC67E[0].DataReg[5].5.1;
PTR Output47->ACC67E[0].DataReg[5].6.1;
PTR Output48->ACC67E[0].DataReg[5].7.1;
```

Typically, these pointers would be put in a “Global Includes” file.



Note

To switch/add definitions to a different card, change the ACC67E[0] to ACC67E[**n**], where **n** is the card index set by the dip switch settings.

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

USING THE ACC-67E WITH TURBO UMAC

Using the ACC-67E with Turbo UMAC requires:

- Knowing (configuring) the base address of the card
- Pointing M-Variables to the appropriate I/O memory locations

Assigning M-Variables to I/O Memory Locations

Mapping an ACC-67E at $78C00, for example, with user configurable M-Variable numbers and name substitutions:

```
#define Output1  M7001   Output1->Y:$078C00,0,1
#define Output2  M7002   Output2->Y:$078C00,1,1
#define Output3  M7003   Output3->Y:$078C00,2,1
#define Output4  M7004   Output4->Y:$078C00,3,1
#define Output5  M7005   Output5->Y:$078C00,4,1
#define Output6  M7006   Output6->Y:$078C00,5,1
#define Output7  M7007   Output7->Y:$078C00,6,1
#define Output8  M7008   Output8->Y:$078C00,7,1

#define Output9  M7009   Output9->Y:$078C01,0,1
#define Output10 M7010   Output10->Y:$078C01,1,1
#define Output11 M7011   Output11->Y:$078C01,2,1
#define Output12 M7012   Output12->Y:$078C01,3,1
#define Output13 M7013   Output13->Y:$078C01,4,1
#define Output14 M7014   Output14->Y:$078C01,5,1
#define Output15 M7015   Output15->Y:$078C01,6,1
#define Output16 M7016   Output16->Y:$078C01,7,1

#define Output17 M7017   Output17->Y:$078C02,0,1
#define Output18 M7018   Output18->Y:$078C02,1,1
#define Output19 M7019   Output19->Y:$078C02,2,1
#define Output20 M7020   Output20->Y:$078C02,3,1
#define Output21 M7021   Output21->Y:$078C02,4,1
#define Output22 M7022   Output22->Y:$078C02,5,1
#define Output23 M7023   Output23->Y:$078C02,6,1
#define Output24 M7024   Output24->Y:$078C02,7,1
```

```
#define Output25 M7025   Output25->Y:$078C03,0,1
#define Output26 M7026   Output26->Y:$078C03,1,1
#define Output27 M7027   Output27->Y:$078C03,2,1
#define Output28 M7028   Output28->Y:$078C03,3,1
#define Output29 M7029   Output29->Y:$078C03,4,1
#define Output30 M7030   Output30->Y:$078C03,5,1
#define Output31 M7031   Output31->Y:$078C03,6,1
#define Output32 M7032   Output32->Y:$078C03,7,1

#define Output33 M7033   Output33->Y:$078C04,0,1
#define Output34 M7034   Output34->Y:$078C04,1,1
#define Output35 M7035   Output35->Y:$078C04,2,1
#define Output36 M7036   Output36->Y:$078C04,3,1
#define Output37 M7037   Output37->Y:$078C04,4,1
#define Output38 M7038   Output38->Y:$078C04,5,1
#define Output39 M7039   Output39->Y:$078C04,6,1
#define Output40 M7040   Output40->Y:$078C04,7,1

#define Output41 M7041   Output41->Y:$078C05,0,1
#define Output42 M7042   Output42->Y:$078C05,1,1
#define Output43 M7043   Output43->Y:$078C05,2,1
#define Output44 M7044   Output44->Y:$078C05,3,1
#define Output45 M7045   Output45->Y:$078C05,4,1
#define Output46 M7046   Output46->Y:$078C05,5,1
#define Output47 M7047   Output47->Y:$078C05,6,1
#define Output48 M7048   Output48->Y:$078C05,7,1
```



Note

To address a different card, replace the **xx** digits in the address locations $07**xx**00 to $07**xx**05 to correspond to the base address configured by the dip switch settings.

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

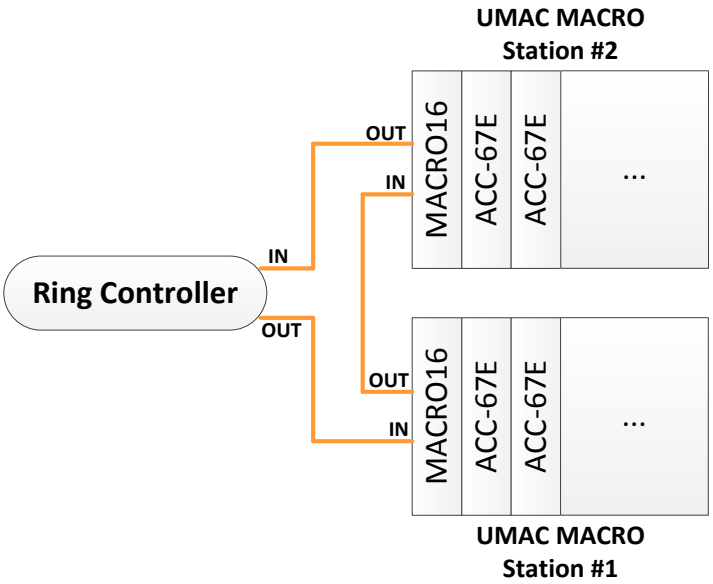
USING THE ACC-67E IN A MACRO STATION

In a MACRO configuration, the ACC-67E resides typically in a UMAC MACRO Station with a MACRO16 CPU, or MACRO8 (legacy).

The ring controller can be either a Turbo or a Power PMAC.

- Turbo PMAC ring controllers:
  Turbo Brick family
  Turbo UMAC with ACC-5E
  Turbo UltraLite

- Power PMAC ring controllers:
  Power Brick family
  Power UMAC with ACC-5E
  Power UMAC with ACC-5E3
  Power EtherLite



Generally speaking, the goal is to transfer ring controller data to the ACC-67E by writing to the 48-bits of outputs.

This data transfer is accomplished using I/O nodes; therefore, it is essential to know:

- Which I/O nodes are used
- Which MACRO station I/O node addresses correspond to which Ring Controller I/O node addresses

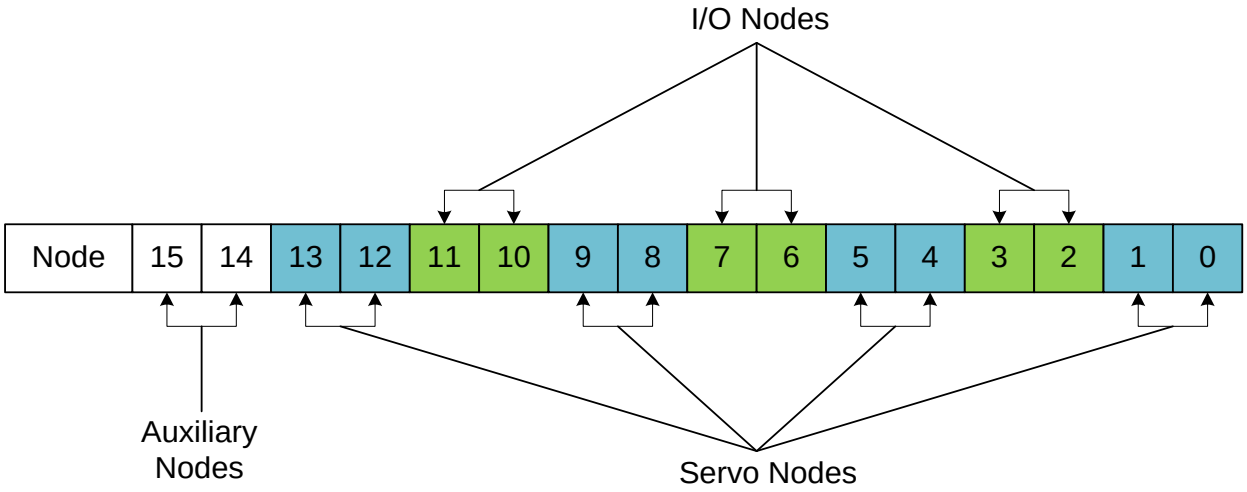
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

MACRO16 I/O Node Addressing

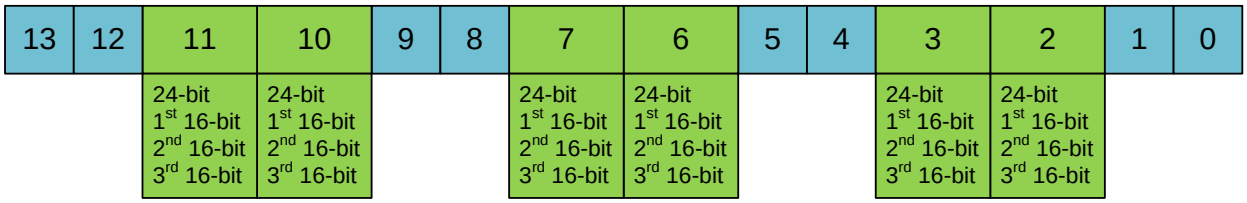
A MACRO IC consists of a number of auxiliary, servo, and I/O nodes:

- Auxiliary nodes are Master/Control registers and are for internal firmware use.
- Servo nodes carry information such as feedback, commands, and flags for motor control.
- I/O nodes are by default unoccupied and are configurable for transferring miscellaneous data.

The MACRO16 CPU is populated with two PMAC2 style MACRO ICs, each consisting of 16 nodes: 2 auxiliary, 8 servo, and 6 I/O nodes (the legacy MACRO8 has only one IC).



Each I/O node consists of one 24-bit and three 16-bit (upper) data registers:





Note

The MACRO8 CPU is populated with only 1 PMAC2 Style MACRO IC (IC# 0).

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

The I/O node data register addresses ($C0XX) for the two MACRO ICs on a station are:

	MACRO Station IC #0					
Node	**2**	**3**	**6**	**7**	**10**	**11**
24-bit	X:$C0A0	X:$C0A4	X:$C0A8	X:$C0AC	X:$C0B0	X:$C0B4
16-bit	X:$C0A1	X:$C0A5	X:$C0A9	X:$C0AD	X:$C0B1	X:$C0B5
16-bit	X:$C0A2	X:$C0A6	X:$C0AA	X:$C0AE	X:$C0B2	X:$C0B6
16-bit	X:$C0A3	X:$C0A7	X:$C0AB	X:$C0AF	X:$C0B3	X:$C0B7

	MACRO Station IC #1					
Node	**2**	**3**	**6**	**7**	**10**	**11**
24-bit	X:$C0E0	X:$C0E4	X:$C0E8	X:$C0EC	X:$C0F0	X:$C0F4
16-bit	X:$C0E1	X:$C0E5	X:$C0E9	X:$C0ED	X:$C0F1	X:$C0F5
16-bit	X:$C0E2	X:$C0E6	X:$C0EA	X:$C0EE	X:$C0F2	X:$C0F6
16-bit	X:$C0E3	X:$C0E7	X:$C0EB	X:$C0EF	X:$C0F3	X:$C0F7



Note

I/O nodes which will be chosen to transfer the ACC-67E(s) data are configurable and chosen by the user depending on availability and I/O node management in the MACRO station.



Note

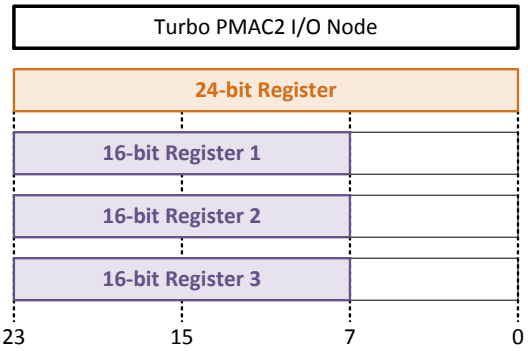
Non-Turbo PMAC2 Ultralite (legacy) I/O node addresses are the same as MACRO Station IC# 0 node registers. For instance, addresses on the ring controller start at $C0A0, instead of at $78420 as seen in the following section.

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

Ring Controller I/O Node Addressing

Turbo PMAC2 I/O Node Addressing

Turbo ring controllers interface with PMAC2 Style MACRO ICs. The data resides in bit fields as illustrated on the right:



Turbo ring controllers can be populated with up to 4 PMAC2 style MACRO ICs (reported by **I4902**). Below are the I/O node addresses ($7XXXX) for each of the PMAC2 Style MACRO ICs:

	Turbo Ring Controller MACRO IC #0 Node Registers					
Station I/O Node#	**2**	**3**	**6**	**7**	**10**	**11**
Ring Controller I/O Node#	**2**	**3**	**6**	**7**	**10**	**11**
24-bit	X:$78420	X:$78424	X:$78428	X:$7842C	X:$78430	X:$78434
16-bit	X:$78421	X:$78425	X:$78429	X:$7842D	X:$78431	X:$78435
16-bit	X:$78422	X:$78426	X:$7842A	X:$7842E	X:$78432	X:$78436
16-bit	X:$78423	X:$78427	X:$7842B	X:$7842F	X:$78433	X:$78437

	Turbo Ring Controller MACRO IC #1 Node Registers					
Station I/O Node#	**2**	**3**	**6**	**7**	**10**	**11**
Ring Controller I/O Node#	**18**	**19**	**22**	**23**	**26**	**27**
24-bit	X:$79420	X:$79424	X:$79428	X:$7942C	X:$79430	X:$79434
16-bit	X:$79421	X:$79425	X:$79429	X:$7942D	X:$79431	X:$79435
16-bit	X:$79422	X:$79426	X:$7942A	X:$7942E	X:$79432	X:$79436
16-bit	X:$79423	X:$79427	X:$7942B	X:$7942F	X:$79433	X:$79437

	Turbo Ring Controller MACRO IC #2 Node Registers					
Station I/O Node#	**2**	**3**	**6**	**7**	**10**	**11**
Ring Controller I/O Node#	**34**	**35**	**38**	**39**	**42**	**43**
24-bit	X:$7A420	X:$7A424	X:$7A428	X:$7A42C	X:$7A430	X:$7A434
16-bit	X:$7A421	X:$7A425	X:$7A429	X:$7A42D	X:$7A431	X:$7A435
16-bit	X:$7A422	X:$7A426	X:$7A42A	X:$7A42E	X:$7A432	X:$7A436
16-bit	X:$7A423	X:$7A427	X:$7A42B	X:$7A42F	X:$7A433	X:$7A437

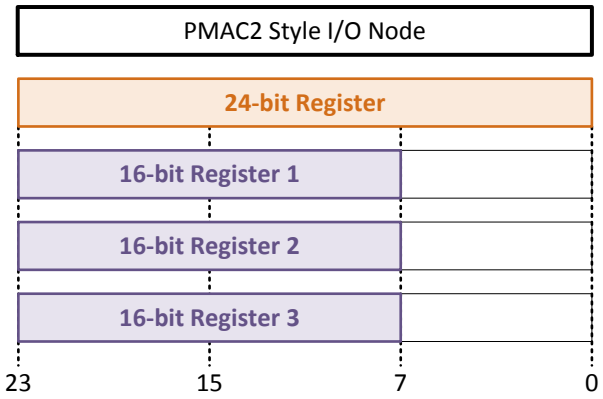
	Turbo Ring Controller MACRO IC #3 Node Registers					
Station I/O Node#	**2**	**3**	**6**	**7**	**10**	**11**
Ring Controller I/O Node#	**50**	**51**	**54**	**55**	**58**	**59**
24-bit	X:$7B420	X:$7B424	X:$7B428	X:$7B42C	X:$7B430	X:$7B434
16-bit	X:$7B421	X:$7B425	X:$7B429	X:$7B42D	X:$7B431	X:$7B435
16-bit	X:$7B422	X:$7B426	X:$7B42A	X:$7B42E	X:$7B432	X:$7B436
16-bit	X:$7B423	X:$7B427	X:$7B42B	X:$7B42F	X:$7B433	X:$7B437

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

Power PMAC2 I/O Node Addressing

The Power PMAC can interface with the ACC-5E which carries PMAC2 Style MACRO ICs.

The PMAC2 Style MACRO IC I/O node data registers reside in bit fields as illustrated on the right:



And the corresponding structure elements:

Structure Element	Data Register
Gate2[*i*].Macro[*j*][0]	24 bits
Gate2[*i*].Macro[*j*][1]	16 bits
Gate2[*i*].Macro[*j*][2]	16 bits
Gate2[*i*].Macro[*j*][3]	16 bits

Where:
- ***i*** is the PMAC2 Style MACRO IC index
- ***j*** is the I/O node number.



Note

Bitwise mapping into the PMAC2 Style MACRO structure elements requires Power PMAC firmware version **1.5.8.305** or newer.

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

The tables below show the I/O Node numbers of the 4 PMAC2 Style MACRO ICs:

	Gate2[**0**]					
Station I/O Node#	2	3	6	7	10	11
Ring Controller I/O Node [*j*]	**2**	**3**	**6**	**7**	**10**	**11**

	Gate2[**1**]					
Station I/O Node#	2	3	6	7	11	12
Ring Controller I/O Node [*j*]	**18**	**19**	**22**	**23**	**26**	**27**

	Gate2[**2**]					
Station I/O Node#	2	3	6	7	10	11
Ring Controller I/O Node [*j*]	**34**	**35**	**38**	**39**	**42**	**43**

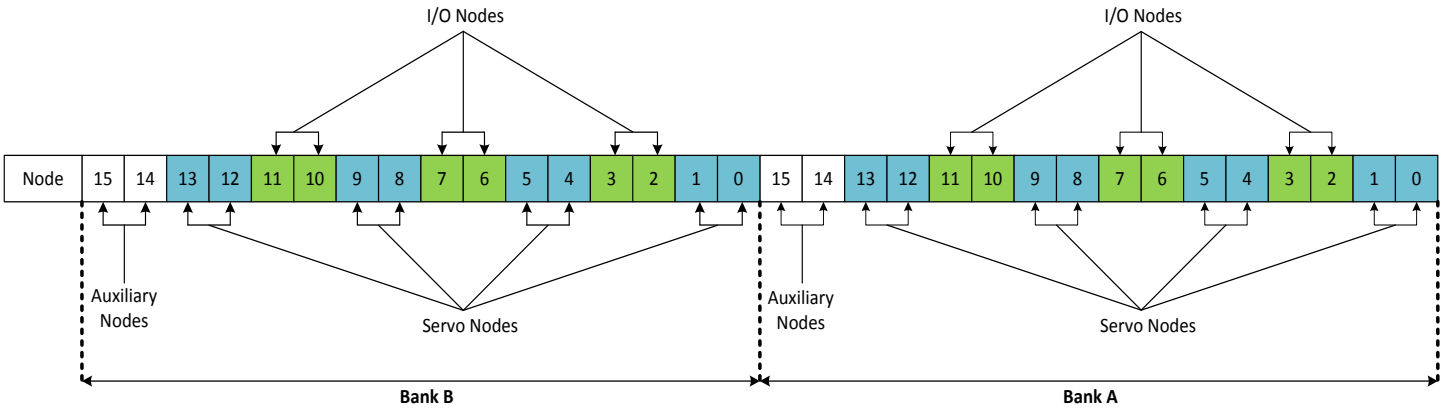
	Gate2[**3**]					
Station I/O Node#	2	3	6	7	11	12
Ring Controller I/O Node [*j*]	**50**	**51**	**54**	**55**	**58**	**59**

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

Power PMAC3 I/O Node Addressing

A PMAC3 style MACRO IC consists of 32 nodes: 4 auxiliary, 16 servo, and 12 I/O nodes. One or more of these ICs can be found in the following hardware:

- Power Brick Family
- Power UMAC with ACC-5E3
- Power EtherLite

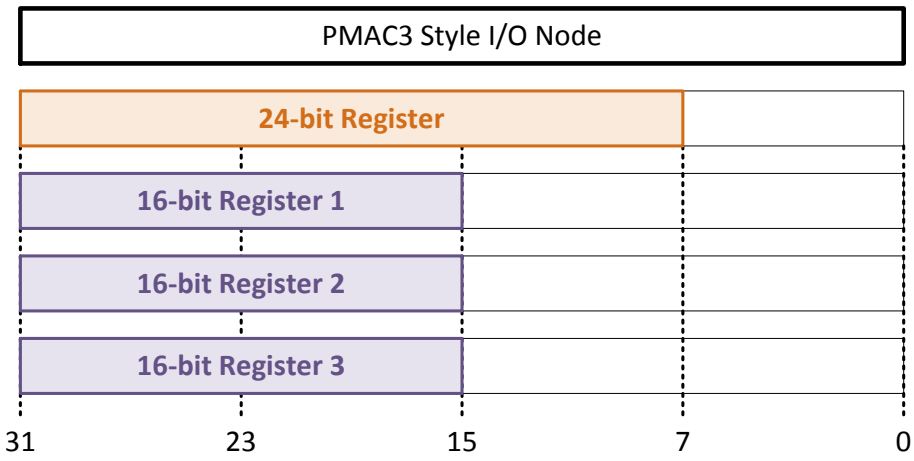




Note

The Power PMAC can interface with up to 16 PMAC3 Style MACRO ICs. ICs present are reported by the variable **Macro.IC3s.**

A PMAC3 Style MACRO IC I/O node consists of 4 data registers: 1 x 24-bit and 3 x 16-bit residing in the following bit fields:



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

With the PMAC3 Style MACRO IC, I/O node data registers are typically accessed using structure elements:

Bank B		Bank A		Data Register
Inputs	**Outputs**	**Inputs**	**Outputs**	
Gate3[*i*].MacroInB[*j*][0]	Gate3[*i*].MacroOutB[*j*][0]	Gate3[*i*].MacroInA[*j*][0]	Gate3[*i*].MacroOutA[*j*][0]	24-bit
Gate3[*i*].MacroInB[*j*][1]	Gate3[*i*].MacroOutB[*j*][1]	Gate3[*i*].MacroInA[*j*][1]	Gate3[*i*].MacroOutA[*j*][1]	1st 16-bit
Gate3[*i*].MacroInB[*j*][2]	Gate3[*i*].MacroOutB[*j*][2]	Gate3[*i*].MacroInA[*j*][2]	Gate3[*i*].MacroOutA[*j*][2]	2nd 16-bit
Gate3[*i*].MacroInB[*j*][3]	Gate3[*i*].MacroOutB[*j*][3]	Gate3[*i*].MacroInA[*j*][3]	Gate3[*i*].MacroOutA[*j*][3]	3rd 16-bit

Where:
- ***i*** is the PMAC3 Style MACRO IC index
- ***j*** is the I/O node number.



Note

Bitwise mapping into the PMAC3 Style MACRO structure elements requires Power PMAC firmware version **1.5.8.305** or newer.

Below are example tables showing I/O Node numbers of the first 4 PMAC3 Style MACRO ICs:

	Gate3[0]											
	Bank A						Bank B					
Station Node#	2	3	6	7	10	11	2	3	6	7	11	12
Ring Controller I/O Node **[*j*]**	**2**	**3**	**6**	**7**	**10**	**11**	**18**	**19**	**22**	**23**	**26**	**27**

	Gate3[1]											
	Bank A						Bank B					
Station Node#	2	3	6	7	10	11	2	3	6	7	11	12
Ring Controller I/O Node **[*j*]**	**34**	**35**	**38**	**39**	**42**	**43**	**50**	**51**	**54**	**55**	**58**	**59**

	Gate3[2]											
	Bank A						Bank B					
Station Node#	2	3	6	7	10	11	2	3	6	7	11	12
Ring Controller I/O Node **[*j*]**	**66**	**67**	**70**	**71**	**74**	**75**	**82**	**83**	**86**	**87**	**90**	**91**

	Gate3[3]											
	Bank A						Bank B					
Station Node#	2	3	6	7	10	11	2	3	6	7	11	12
Ring Controller I/O Node **[*j*]**	**98**	**99**	**102**	**103**	**106**	**107**	**114**	**115**	**118**	**119**	**122**	**123**

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

Configuring MACRO I/O Transfers

Having set up the following:

- Ring Controller MACRO communication for ASCII and MS commands
  I6840 with Turbo PMAC2
  Gate2[i].MacroMode with Power PMAC2
  Gate3[i].MacroModeA and **Gate3[i].MacroModeB** with Power PMAC3

- MACRO Station MACRO communication for ASCII and MS commands
  MI996

- Enabled the chosen I/O nodes on the ring controller
  I6841 with Turbo PMAC2
  Gate2[i].MacroEnable with Power PMAC2
  Gate3[i].MacroEnableA and **Gate3[i].MacroEnableB** with Power PMAC3

- Enabled the corresponding I/O nodes on the MACRO station IC
  MS{anynode},MI975
  Typically, masked with the enabled I/O node(s) E.g. MS2,MI975=$4 enables transfers using I/O node number 2.

- Set up the I/O Data Transfer Period
  MS{anynode},MI19
  Typically = 4

The ACC-67E(s) data should then be ready to transfer to the MACRO Station.

Each ACC-67E possesses 48 bits of transfer data. Depending on the number of cards in the MACRO station and I/O nodes available, one or more of the following methods can be used:

- **MS{anynode},MI71**
  48-bit transfer into 2 x 24-bit data registers (uses 2 x I/O nodes)

- **MS{anynode},MI69/MI70**
  48-bit transfer into 3 x 16-bit data registers (uses 1 x I/O nodes)



Note

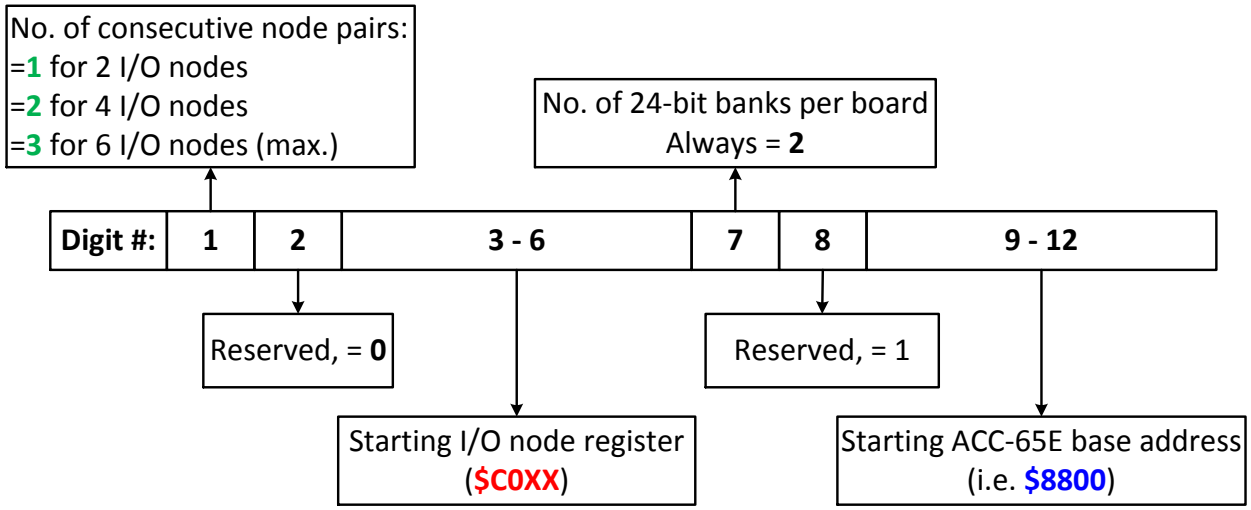
{anynode} refers to any activated node on a particular MACRO IC.

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

MS{anynode},MI71

MI71 transfers all 48 bits of an ACC-67E into 2 x consecutive I/O node 24-bit data registers. It handles up to 3 x ACC-67Es at consecutive base addresses (e.g. Y:$8800, Y:$9800, Y:$A800), placing the data in 6 x consecutive 24-bit I/O node data registers.

MI71 is a 48-bit variable represented as 12 hexadecimal digits which are set up as follows (digit #1 is leftmost when constructing the word):





Note

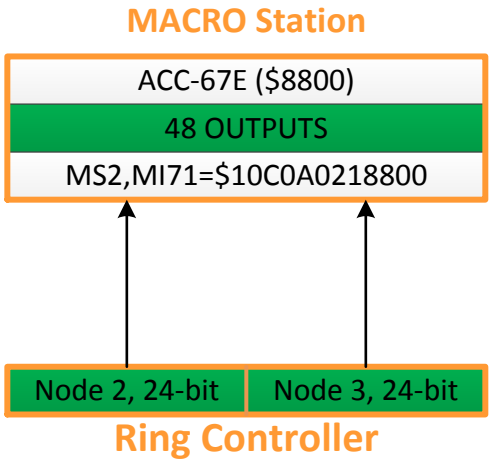
For multiple ACC-67E transfers with MI71, consecutive cards must be under the same chip select.

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

MI71 Example 1

Transferring data to one ACC-67E at address $8800 over nodes 2 and 3 of MACRO IC 0:

```
MS2,MI19=4
MS2,MI71=$10C0A0218800
MS2,MI975=$C
```



Having downloaded the above settings into the MACRO16 station, the outputs can then be transferred from the ring controller side (I/O nodes 2 and 3) in the following bit fields:

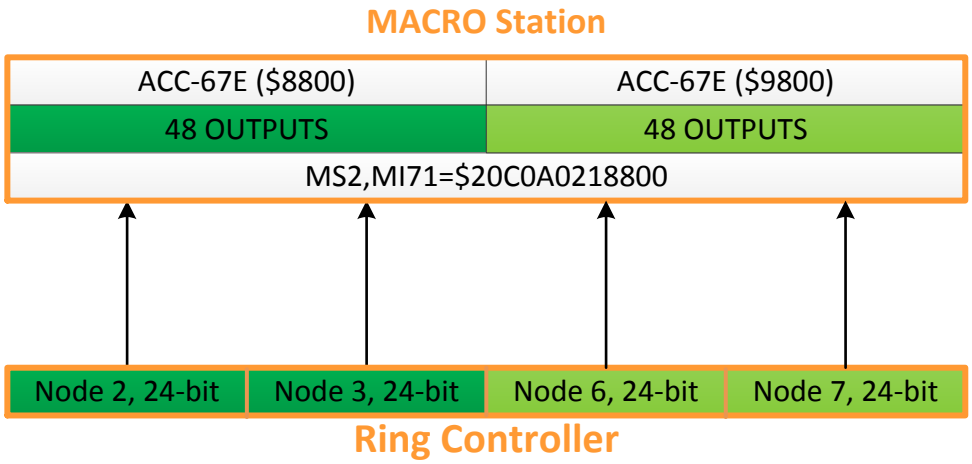
	Turbo		**Power**			
	I/O Node Address	Data Bits	Structure Element (PMAC2)	Data Bits	Structure Element (PMAC3)	Data Bits
Outputs 1-24	X:$78420	23:00	Gate2[0].Macro[2][0]	23:00	Gate3[0].MacroOutA[2][0]	31:08
Outputs 25-48	X:$78424	23:00	Gate2[0].Macro[3][0]	23:00	Gate3[0].MacroOutA[3][0]	31:08

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

MI71 Example 2

Transferring data to two ACC-67Es at consecutive base addresses $8800 and $9800 over nodes 2, 3, 6, and 7 of MACRO IC 0:

```
MS2,MI19=4
MS2,MI71=$20C0A0218800
MS2,MI975=$CC
```



Having downloaded the above settings into the MACRO16 station, the outputs can then be transferred from the ring controller side (I/O nodes 2, 3, 6, and 7) in the following bit fields:

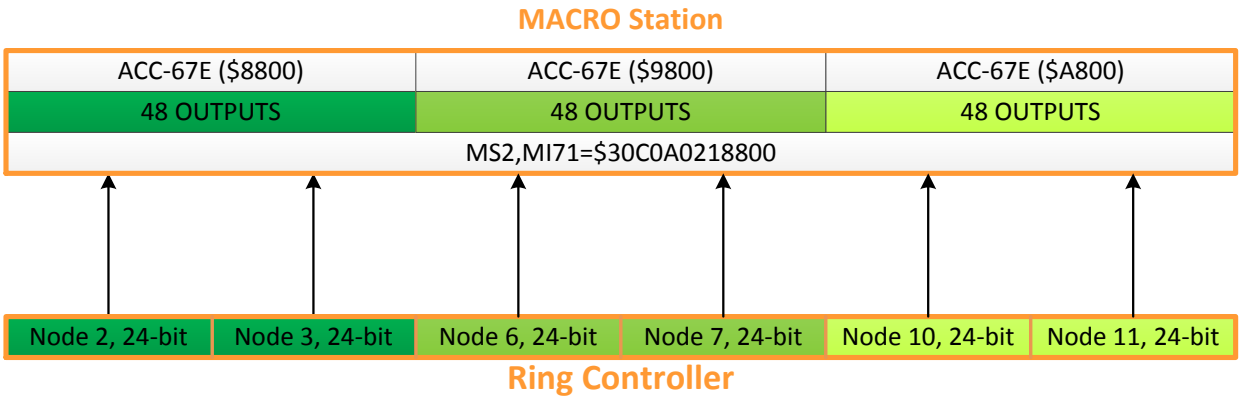
		Turbo		**Power**			
		I/O Node Address	Data Bits	Structure Element (PMAC2)	Data Bits	Structure Element (PMAC3)	Data Bits
1st ACC-67E	Outputs 1-24	X:$78420	23:00	Gate2[0].Macro[2][0]	23:00	Gate3[0].MacroOutA[2][0]	31:08
	Outputs 25-48	X:$78424	23:00	Gate2[0].Macro[3][0]	23:00	Gate3[0].MacroOutA[3][0]	31:08
2nd ACC-67E	Outputs 49-72	X:$78428	23:00	Gate2[0].Macro[6][0]	23:00	Gate3[0].MacroOutA[6][0]	31:08
	Outputs 73-96	X:$7842C	23:00	Gate2[0].Macro[7][0]	23:00	Gate3[0].MacroOutA[7][0]	31:08

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

MI71 Example 3

Transferring data to three ACC-67Es at consecutive base addresses $8800, $9800, and $A800 over nodes 2, 3, 6, 7, 10, and 11:

```
MS2,MI19=4
MS2,MI71=$30C0A0218800
MS2,MI975=$CCC
```



Having downloaded the above settings into the MACRO16 station, the outputs can then be transferred from the ring controller side (I/O nodes 2, 3, 6, 7, 10, and 11) in the following bit fields:

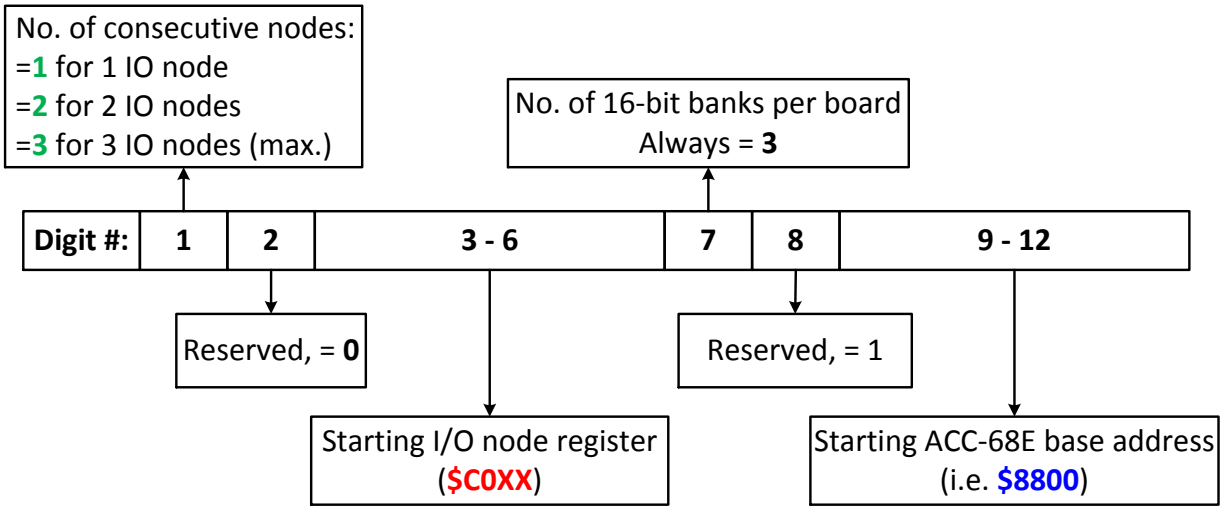
		Turbo		Power			
		I/O Node Address	Data Bits	Structure Element (PMAC2)	Data Bits	Structure Element (PMAC3)	Data Bits
1st ACC-67E	Outputs 1-24	X:$78420	23:00	Gate2[0].Macro[2][0]	23:00	Gate3[0].MacroOutA[2][0]	31:08
	Outputs 25-48	X:$78424	23:00	Gate2[0].Macro[3][0]	23:00	Gate3[0].MacroOutA[3][0]	31:08
2nd ACC-67E	Outputs 49-72	X:$78428	23:00	Gate2[0].Macro[6][0]	23:00	Gate3[0].MacroOutA[6][0]	31:08
	Outputs 73-96	X:$7842C	23:00	Gate2[0].Macro[7][0]	23:00	Gate3[0].MacroOutA[7][0]	31:08
3rd ACC-67E	Outputs 97-120	X:$78430	23:00	Gate2[0].Macro[10][0]	23:00	Gate3[0].MacroOutA[10][0]	31:08
	Outputs 121-144	X:$78434	23:00	Gate2[0].Macro[11][0]	23:00	Gate3[0].MacroOutA[11][0]	31:08

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

MS{anynode},MI69/MI70

MI69/MI70 transfer all 48 bits of an ACC-67E into 3 x I/O node 16-bit data registers. It handles up to 3 x ACC-67Es at consecutive base addresses (e.g. Y:$8800, Y:$9800, Y:$A800), and places the data in 9 x consecutive 16-bit I/O node data registers.

MI69/70 are 48-bit variables represented as 12 hexadecimal digits which are set up as follows (digit #1 is leftmost when constructing the word):





Note

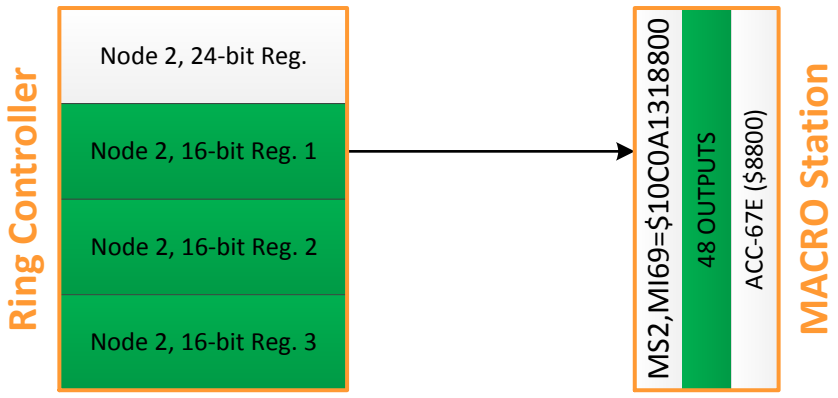
For multiple ACC-67E transfers with MI69/MI70, consecutive cards must be under the same chip select.

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

MI69/70 Example 1

Transferring data to one ACC-67E at address $8800 over node 2 of MACRO IC 0:

```
MS2,MI19=4
MS2,MI69=$10C0A1318800
MS2,MI975=$4
```



Having downloaded the above settings into the MACRO16 station, the outputs can then be transferred from the ring controller side (I/O node 2) in the following bit fields:

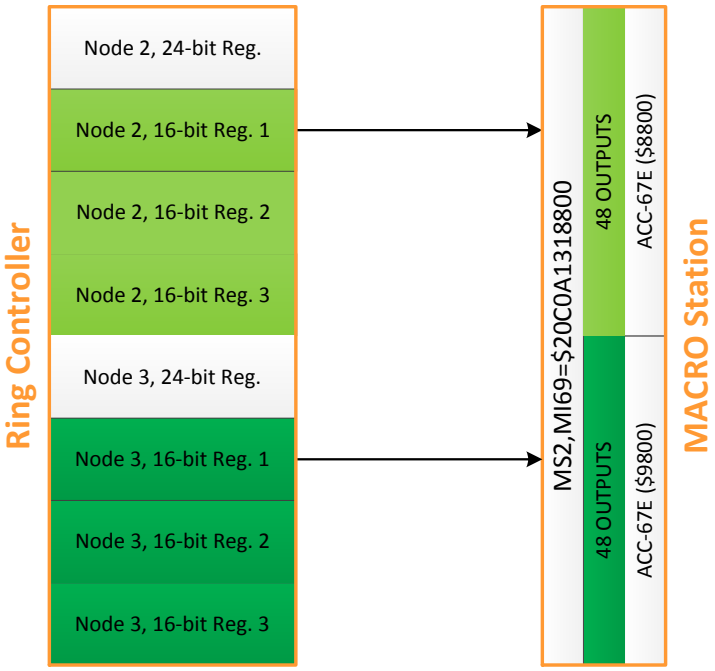
	Turbo		**Power**			
	I/O Node Address	Data Bits	Structure Element (PMAC2)	Data Bits	Structure Element (PMAC3)	Data Bits
Outputs 1-16	X:$78421	23:08	Gate2[0].Macro[2][1]	23:08	Gate3[0].MacroOutA[2][1]	31:16
Outputs 17-32	X:$78422	23:08	Gate2[0].Macro[2][2]	23:08	Gate3[0].MacroOutA[2][2]	31:16
Outputs 33-48	X:$78423	23:08	Gate2[0].Macro[2][3]	23:08	Gate3[0].MacroOutA[2][3]	31:16

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

MI69/70 Example 2

Transferring data to two ACC-67Es at consecutive base addresses $8800 and $9800 over nodes 2 and 3 of MACRO IC 0:

```
MS2,MI19=4
MS2,MI69=$20C0A1318800
MS2,MI975=$C
```



Having downloaded the above settings into the MACRO16 station, the outputs can then be transferred from the ring controller side (I/O nodes 2 and 3) in the following bit fields:

		Turbo		**Power**			
		I/O Node Address	Data Bits	Structure Element (PMAC2)	Data Bits	Structure Element (PMAC3)	Data Bits
1st ACC-67E	Outputs 1-16	X:$78421	23:08	Gate2[0].Macro[2][1]	23:08	Gate3[0].MacroOutA[2][1]	31:16
	Outputs 17-32	X:$78422	23:08	Gate2[0].Macro[2][2]	23:08	Gate3[0].MacroOutA[2][2]	31:16
	Outputs 33-48	X:$78423	23:08	Gate2[0].Macro[2][3]	23:08	Gate3[0].MacroOutA[2][3]	31:16
2nd ACC-67E	Outputs 49-64	X:$78425	23:08	Gate2[0].Macro[3][1]	23:08	Gate3[0].MacroOutA[3][1]	31:16
	Outputs 65-80	X:$78426	23:08	Gate2[0].Macro[3][2]	23:08	Gate3[0].MacroOutA[3][2]	31:16
	Outputs 81-96	X:$78427	23:08	Gate2[0].Macro[3][3]	23:08	Gate3[0].MacroOutA[3][3]	31:16

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

Accessing the Transferred Data

Having transferred the data from the ring controller's registers (outlined above) into the ACC-67E(s), bitwise mapping is the final step.

- With the PMAC2 style MACRO IC:
  The output state is not reported, and writing to two separate bits simultaneously (without using the full word) is not possible, thus an image word is required.

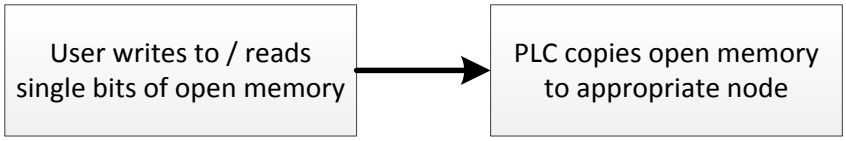
- With the PMAC3 style MACRO IC:
  The outputs can be directly bitwise mapped, written to, and reported.

Outputs Mirror Image Concept

A mirror image is a memory location on the ring controller, typically open or scratch memory that mimics the state of the ACC-67E outputs (residing in the MACRO station). This can be done in a PLC program. The following are recommended memory registers to use:

- Turbo PMAC Open Memory Registers (24 bits) **$10F0-$10FF** (X, and Y)

- Power PMAC Open Memory Registers (32 bits), unsigned **Sys.Udata[*i*]**
  Use Sys.Udata[1], Sys.Udata[2], Sys.Udata[3]…



Bitwise mapping to user memory structure elements in Power PMAC is not allowed. Explicit syntax with offset address must be used. The offset address is found by multiplying the element structure index by 4. To the right is an example of the first 4 unsigned open memory structure elements and their corresponding address offsets:

Structure Element	Address Offset
Sys.Udata[1]	U.USER:4
Sys.Udata[2]	U.USER:8
Sys.Udata[3]	U.USER:12
Sys.Udata[4]	U.USER:16



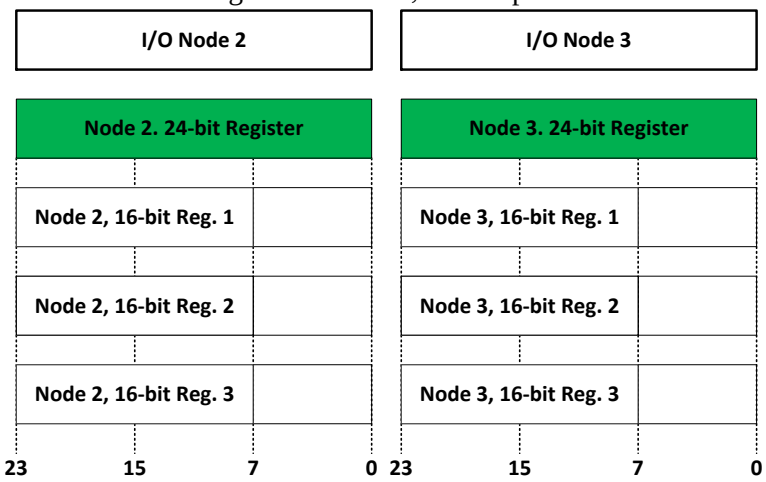
Note

A large number of self-addressed (default **Sys.pushm**) pointers in Power PMAC use **Sys.Udata[0]**; therefore, it is highly advised NOT to use it as a general purpose user memory.

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

Turbo PMAC2 MI71 Mapping Example

MI71 is used to transfer ACC-67E data to MACRO IC 0 nodes 2 and 3 of the ring controller, which places outputs 1-24 in the 24-bit data register of node 2, and outputs 25-48 in the 24-bit register of node 3:



We will use **Y:$10FF** (24 bits) open memory register (to copy the 24 bits of node 2) and **X:$10FF** (24 bits) open memory register (to copy the 24 bits of node 3); therefore, the bitwise mapping of the outputs should point to the open memory.

```
#define Output1  M7001  Output1->Y:$10FF,0,1
#define Output2  M7002  Output2->Y:$10FF,1,1
#define Output3  M7003  Output3->Y:$10FF,2,1
#define Output4  M7004  Output4->Y:$10FF,3,1
#define Output5  M7005  Output5->Y:$10FF,4,1
#define Output6  M7006  Output6->Y:$10FF,5,1
#define Output7  M7007  Output7->Y:$10FF,6,1
#define Output8  M7008  Output8->Y:$10FF,7,1
#define Output9  M7009  Output9->Y:$10FF,8,1
#define Output10 M7010  Output10->Y:$10FF,9,1
#define Output11 M7011  Output11->Y:$10FF,10,1
#define Output12 M7012  Output12->Y:$10FF,11,1
#define Output13 M7013  Output13->Y:$10FF,12,1
#define Output14 M7014  Output14->Y:$10FF,13,1
#define Output15 M7015  Output15->Y:$10FF,14,1
#define Output16 M7016  Output16->Y:$10FF,15,1
#define Output17 M7017  Output17->Y:$10FF,16,1
#define Output18 M7018  Output18->Y:$10FF,17,1
#define Output19 M7019  Output19->Y:$10FF,18,1
#define Output20 M7020  Output20->Y:$10FF,19,1
#define Output21 M7021  Output21->Y:$10FF,20,1
#define Output22 M7022  Output22->Y:$10FF,21,1
#define Output23 M7023  Output23->Y:$10FF,22,1
#define Output24 M7024  Output24->Y:$10FF,23,1
```

```
#define Output25 M7025  Output25->X:$10FF,0,1
#define Output26 M7026  Output26->X:$10FF,1,1
#define Output27 M7027  Output27->X:$10FF,2,1
#define Output28 M7028  Output28->X:$10FF,3,1
#define Output29 M7029  Output29->X:$10FF,4,1
#define Output30 M7030  Output30->X:$10FF,5,1
#define Output31 M7031  Output31->X:$10FF,6,1
#define Output32 M7032  Output32->X:$10FF,7,1
#define Output33 M7033  Output33->X:$10FF,8,1
#define Output34 M7034  Output34->X:$10FF,9,1
#define Output35 M7035  Output35->X:$10FF,10,1
#define Output36 M7036  Output36->X:$10FF,11,1
#define Output37 M7037  Output37->X:$10FF,12,1
#define Output38 M7038  Output38->X:$10FF,13,1
#define Output39 M7039  Output39->X:$10FF,14,1
#define Output40 M7040  Output40->X:$10FF,15,1
#define Output41 M7041  Output41->X:$10FF,16,1
#define Output42 M7042  Output42->X:$10FF,17,1
#define Output43 M7043  Output43->X:$10FF,18,1
#define Output44 M7044  Output44->X:$10FF,19,1
#define Output45 M7045  Output45->X:$10FF,20,1
#define Output46 M7046  Output46->X:$10FF,21,1
#define Output47 M7047  Output47->X:$10FF,22,1
#define Output48 M7048  Output48->X:$10FF,23,1
```


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

The following mirror image PLC (which should be executing constantly) will copy the outputs into nodes 2 and 3:

```
#define N2Twenty4      M7049   ; Node 2, 24-bit register
#define N3Twenty4      M7050   ; Node 3, 24-bit register
#define N2OutMirror    M7051   ; Node 2, Mirror word, open memory
#define N3OutMirror    M7052   ; Node 3, Mirror word, open memory

N2Twenty4->X:$78420,0,24       ; 24-bit register, node 2
N3Twenty4->X:$78424,0,24       ; 24-bit register, node 3
N2OutMirror->Y:$10FF,0,24      ; Open memmory register, node 2
N3OutMirror->X:$10FF,0,24      ; Open memmory register, node 3
N2OutMirror = 0                ; Save/Initialize to zero or desired state
N3OutMirror = 0                ; Save/Initialize to zero or desired state

Open PLC 1 Clear
N2Twenty4 = N2OutMirror        ; Update data register, node 2
N3Twenty4 = N3OutMirror        ; Update data register, node 3
Close
```



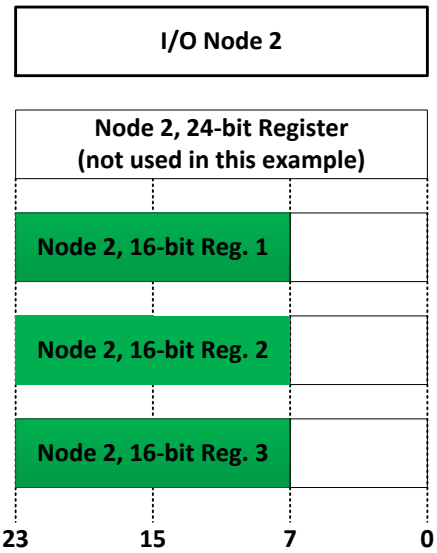
Note

The output states are now reported and can be toggled individually using the bitwise assignments to the open memory registers.

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

Turbo PMAC2 MI69/70 Mapping Example

MI69 / MI70 is used to transfer ACC-67E data through MACRO IC 0 node 2 of the ring controller, which places the data as illustrated on the right:



The 48 bits are divided into the three 16-bit registers of node 2.

We will use **Y:$10FF** (24 bits) open memory register (to copy the 24 bits of node 2) and **X:$10FF** (24 bits) open memory register (to copy the 24 bits of node 3); therefore, the bitwise mapping of the outputs should point to the open memory.

```
#define Output1  M7001   Output1->Y:$10FF,0,1
#define Output2  M7002   Output2->Y:$10FF,1,1
#define Output3  M7003   Output3->Y:$10FF,2,1
#define Output4  M7004   Output4->Y:$10FF,3,1
#define Output5  M7005   Output5->Y:$10FF,4,1
#define Output6  M7006   Output6->Y:$10FF,5,1
#define Output7  M7007   Output7->Y:$10FF,6,1
#define Output8  M7008   Output8->Y:$10FF,7,1
#define Output9  M7009   Output9->Y:$10FF,8,1
#define Output10 M7010   Output10->Y:$10FF,9,1
#define Output11 M7011   Output11->Y:$10FF,10,1
#define Output12 M7012   Output12->Y:$10FF,11,1
#define Output13 M7013   Output13->Y:$10FF,12,1
#define Output14 M7014   Output14->Y:$10FF,13,1
#define Output15 M7015   Output15->Y:$10FF,14,1
#define Output16 M7016   Output16->Y:$10FF,15,1
#define Output17 M7017   Output17->Y:$10FF,16,1
#define Output18 M7018   Output18->Y:$10FF,17,1
#define Output19 M7019   Output19->Y:$10FF,18,1
#define Output20 M7020   Output20->Y:$10FF,19,1
#define Output21 M7021   Output21->Y:$10FF,20,1
#define Output22 M7022   Output22->Y:$10FF,21,1
#define Output23 M7023   Output23->Y:$10FF,22,1
#define Output24 M7024   Output24->Y:$10FF,23,1
```

```
#define Output25 M7025   Output25->X:$10FF,0,1
#define Output26 M7026   Output26->X:$10FF,1,1
#define Output27 M7027   Output27->X:$10FF,2,1
#define Output28 M7028   Output28->X:$10FF,3,1
#define Output29 M7029   Output29->X:$10FF,4,1
#define Output30 M7030   Output30->X:$10FF,5,1
#define Output31 M7031   Output31->X:$10FF,6,1
#define Output32 M7032   Output32->X:$10FF,7,1
#define Output33 M7033   Output33->X:$10FF,8,1
#define Output34 M7034   Output34->X:$10FF,9,1
#define Output35 M7035   Output35->X:$10FF,10,1
#define Output36 M7036   Output36->X:$10FF,11,1
#define Output37 M7037   Output37->X:$10FF,12,1
#define Output38 M7038   Output38->X:$10FF,13,1
#define Output39 M7039   Output39->X:$10FF,14,1
#define Output40 M7040   Output40->X:$10FF,15,1
#define Output41 M7041   Output41->X:$10FF,16,1
#define Output42 M7042   Output42->X:$10FF,17,1
#define Output43 M7043   Output43->X:$10FF,18,1
#define Output44 M7044   Output44->X:$10FF,19,1
#define Output45 M7045   Output45->X:$10FF,20,1
#define Output46 M7046   Output46->X:$10FF,21,1
#define Output47 M7047   Output47->X:$10FF,22,1
#define Output48 M7048   Output48->X:$10FF,23,1
```


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

The following mirror image PLC (which should be executing constantly) will copy the outputs into nodes 2 and 3:

```
#define N2First16      M7049          ; Node 2, second 16-bit register
#define N2Second16     M7050          ; Node 2, second 16-bit register
#define N2Third16      M7051          ; Node 2, third 16-bit register
#define OutMirror      M7052          ; Mirror word, open memory

N2First16->X:$78421,8,16              ; First 16-bit register, node 2
N2Second16->X:$78422,8,16             ; Second 16-bit register, node 2
N2Third16->X:$78423,8,16              ; Third 16-bit register, node 2
OutMirror->D:$10FF                    ; Open memory, double word (X and Y registers)
OutMirror = 0                         ; Save/Initialize to zero or desired state

Open PLC 1 Clear
N2First16 = OutMirror & $FFFF                    ; Mask with lower 16 bits
N2Second16 = OutMirror & $FFFF0000 / $10000      ; Mask with middle 16 bits and shift to lower 16
N2Third16 = OutMirror & $FFFF00000000 / $100000000; Mask with upper 16 bits and shift to lower 16
Close
```



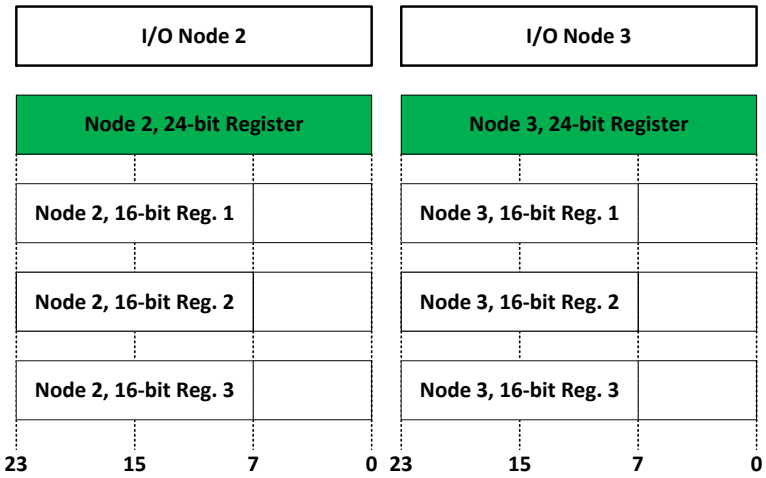
Note

The output states are now reported and can be toggled individually using the bitwise assignments to the open memory registers.

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

Power PMAC2 MI71 Mapping Example

MI71 is used to transfer ACC-67E data to MACRO IC 0 nodes 2 and 3 of the ring controller, which places 24 bits of outputs in the 24-bit register of node 2 and 24 bits of outputs in the 24-bit register of node 3:





Note

Bitwise mapping with PMAC2 style MACRO IC is supported starting with Power PMAC firmware version 1.5.8.305.

We will use **Sys.Udata[1]** open memory register (to copy the 24 bits of node 2) and **Sys.Udata[2]** open memory register (to copy the 24 bits of node 3) to create the image words. Therefore, the bitwise mapping of the outputs:

```
PTR Output1->U.USER:4.0.1;
PTR Output2->U.USER:4.1.1;
PTR Output3->U.USER:4.2.1;
PTR Output4->U.USER:4.3.1;
PTR Output5->U.USER:4.4.1;
PTR Output6->U.USER:4.5.1;
PTR Output7->U.USER:4.6.1;
PTR Output8->U.USER:4.7.1;
PTR Output9->U.USER:4.8.1;
PTR Output10->U.USER:4.9.1;
PTR Output11->U.USER:4.10.1;
PTR Output12->U.USER:4.11.1;
PTR Output13->U.USER:4.12.1;
PTR Output14->U.USER:4.13.1;
PTR Output15->U.USER:4.14.1;
PTR Output16->U.USER:4.15.1;
PTR Output17->U.USER:4.16.1;
PTR Output18->U.USER:4.17.1;
PTR Output19->U.USER:4.18.1;
PTR Output20->U.USER:4.19.1;
PTR Output21->U.USER:4.20.1;
PTR Output22->U.USER:4.21.1;
PTR Output23->U.USER:4.22.1;
PTR Output24->U.USER:4.23.1;
```

```
PTR Output25->U.USER:8.0.1;
PTR Output26->U.USER:8.1.1;
PTR Output27->U.USER:8.2.1;
PTR Output28->U.USER:8.3.1;
PTR Output29->U.USER:8.4.1;
PTR Output30->U.USER:8.5.1;
PTR Output31->U.USER:8.6.1;
PTR Output32->U.USER:8.7.1;
PTR Output33->U.USER:8.8.1;
PTR Output34->U.USER:8.9.1;
PTR Output35->U.USER:8.10.1;
PTR Output36->U.USER:8.11.1;
PTR Output37->U.USER:8.12.1;
PTR Output38->U.USER:8.13.1;
PTR Output39->U.USER:8.14.1;
PTR Output40->U.USER:8.15.1;
PTR Output41->U.USER:8.16.1;
PTR Output42->U.USER:8.17.1;
PTR Output43->U.USER:8.18.1;
PTR Output44->U.USER:8.19.1;
PTR Output45->U.USER:8.20.1;
PTR Output46->U.USER:8.21.1;
PTR Output47->U.USER:8.22.1;
PTR Output48->U.USER:8.23.1;
```


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

Bitwise mapping to user memory structure elements is not allowed. Explicit syntax with offset address must be used. The offset address is found by multiplying the element structure index by 4. To the right is an example of the first 4 unsigned open memory structure elements and their corresponding address offsets:

Structure Element	Address Offset
Sys.Udata[1]	U.USER:4
Sys.Udata[2]	U.USER:8
Sys.Udata[3]	U.USER:12
Sys.Udata[4]	U.USER:16



Note

A large number of self-addressed (default **Sys.pushm**) pointers in Power PMAC use **Sys.Udata[0]**; therefore, it is highly advised NOT to use it as general purpose user memory.

The following mirror image PLC (which should be executing constantly) will copy the outputs into nodes 2 and 3:

```
PTR N2Twenty4->Gate2[0].Macro;              // 24-bit register, node 2
PTR N3Twenty4->Gate2[0].Macro ;             // 24-bit register, node 3

PTR N2OutMirror->U.USER:4.0.24;             // Node 2 Mirror Word, shared user memory
PTR N3OutMirror->U.USER:8.0.24;             // Node 3 Mirror Word, shared user memory

N2OutMirror = 0                             // Save/initialize to zero or desired state
N3OutMirror = 0                             // Save/initialize to zero or desired state

Open PLC 1
N2Twenty4 = N2OutMirror                     // Update data register
N3Twenty4 = N3OutMirror                     // Update data register
Close
```



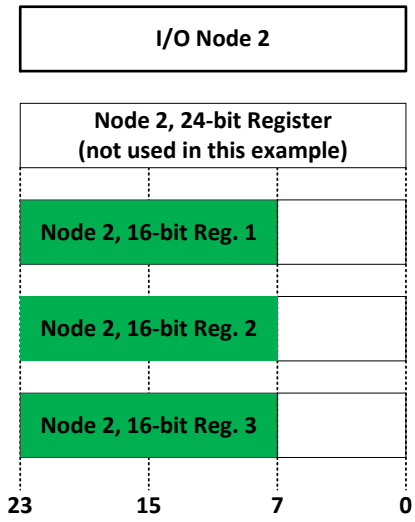
Note

The output states are now reported and can be toggled individually using the bitwise assignments to the open memory registers.

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

Power PMAC2 MI69/70 Mapping Example

MI69 / MI70 is used to transfer ACC-67E data through MACRO IC 0 node 2 of the ring controller, which places the data as illustrated on the right:



The 48 bits are divided into the three 16-bit registers of node 2.

We will use the 64-bit double-precision element **Sys.Ddata[1]** open memory register (to copy the 48 bits of node 2 registers 1,2, and 3) to create one image word. Therefore, the bitwise mapping of the outputs:

```
PTR Output1->D.USER:8.0.1;
PTR Output2->D.USER:8.1.1;
PTR Output3->D.USER:8.2.1;
PTR Output4->D.USER:8.3.1;
PTR Output5->D.USER:8.4.1;
PTR Output6->D.USER:8.5.1;
PTR Output7->D.USER:8.6.1;
PTR Output8->D.USER:8.7.1;
PTR Output9->D.USER:8.8.1;
PTR Output10->D.USER:8.9.1;
PTR Output11->D.USER:8.10.1;
PTR Output12->D.USER:8.11.1;
PTR Output13->D.USER:8.12.1;
PTR Output14->D.USER:8.13.1;
PTR Output15->D.USER:8.14.1;
PTR Output16->D.USER:8.15.1;
PTR Output17->D.USER:8.16.1;
PTR Output18->D.USER:8.17.1;
PTR Output19->D.USER:8.18.1;
PTR Output20->D.USER:8.19.1;
PTR Output21->D.USER:8.20.1;
PTR Output22->D.USER:8.21.1;
PTR Output23->D.USER:8.22.1;
PTR Output24->D.USER:8.23.1;
```

```
PTR Output25->D.USER:8.24.1;
PTR Output26->D.USER:8.25.1;
PTR Output27->D.USER:8.26.1;
PTR Output28->D.USER:8.27.1;
PTR Output29->D.USER:8.28.1;
PTR Output30->D.USER:8.29.1;
PTR Output31->D.USER:8.30.1;
PTR Output32->D.USER:8.31.1;
PTR Output33->D.USER:8.32.1;
PTR Output34->D.USER:8.33.1;
PTR Output35->D.USER:8.34.1;
PTR Output36->D.USER:8.35.1;
PTR Output37->D.USER:8.36.1;
PTR Output38->D.USER:8.37.1;
PTR Output39->D.USER:8.38.1;
PTR Output40->D.USER:8.39.1;
PTR Output41->D.USER:8.40.1;
PTR Output42->D.USER:8.41.1;
PTR Output43->D.USER:8.42.1;
PTR Output44->D.USER:8.43.1;
PTR Output45->D.USER:8.44.1;
PTR Output46->D.USER:8.45.1;
PTR Output47->D.USER:8.46.1;
PTR Output48->D.USER:8.47.1;
```

Bitwise mapping to user memory structure elements is not allowed. Explicit syntax with offset address must be used. A double register offset address is found by multiplying the element structure index by 8. To the right is an example of the first 4 unsigned open memory structure elements and their corresponding address offsets:

Structure Element	Address Offset
Sys.Ddata[1]	D.USER:8
Sys.Ddata[2]	D.USER:16
Sys.Ddata[3]	D.USER:24
Sys.Ddata[4]	D.USER:8

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



Note

Sys.Ddata[i] is a double-precision floating-point element. Sys.Ddata[1] occupies the same registers which Sys.Udata[2] and Sys.Udata[3] would occupy (64 bits). The user is responsible for preventing overlap issues.



Note

A large number of self-addressed (default **Sys.pushm**) pointers in Power PMAC use **Sys.Udata[0]**; therefore, it is highly advised NOT to use it as general purpose user memory.

The following mirror image PLC (which should be executing constantly) will copy the outputs into node 2:

```
PTR N2First16->Gate2[0].Macro[2][1]   ;       // Node 2, First 16-bit register
PTR N2Second16->Gate2[0].Macro[2][2];         // Node 2, Second 16-bit register
PTR N2Third16->Gate2[0].Macro[2][3]   ;       // Node 2, Third 16-bit register

PTR OutMirror->D.USER:8;                      // Mirror Word, shared user memory
OutMirror = 0                                 // Save/initialize to zero or desired state

Open PLC 1
N2First16 = OutMirror & FFFF <<8              // Update 1st 16-bit reg; shift to upper 16-bits
N2Second16 = OutMirror & $FFFF0000 >>8        // Update 2nd 16-bit reg; shift to upper 16-bits
N2Third16 = OutMirror & $FFFF00000000 >>24    // Update 3rdt 16-bit reg; shift to upper 16-bits
Close
```



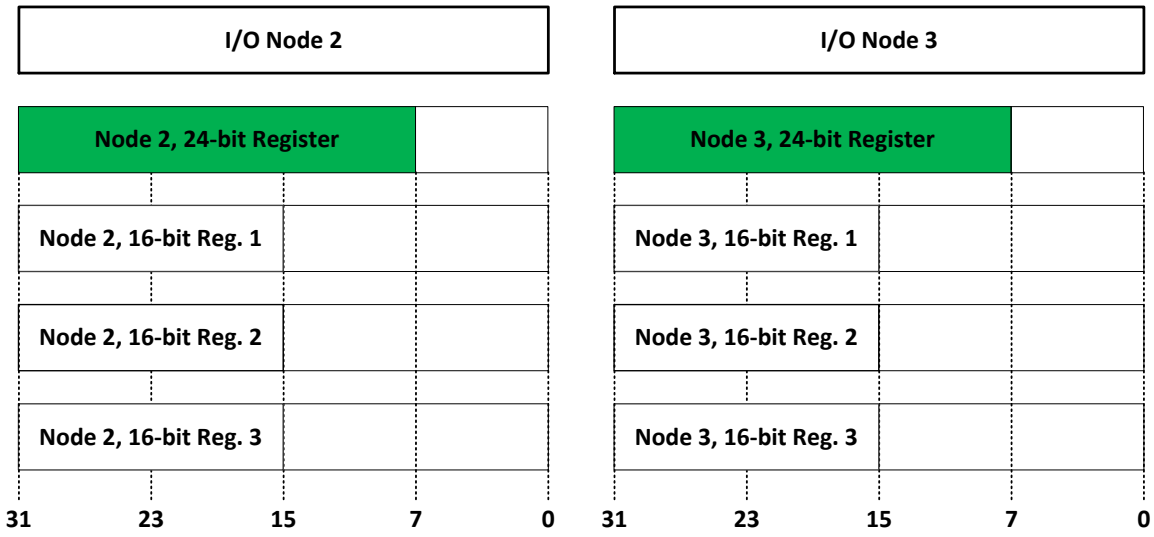
Note

The output states are now reported and can be toggled individually using the bitwise assignments to the open memory registers.

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

Power PMAC3 MI71 Mapping Example

MI71 is used to transfer ACC-67E data to MACRO IC 0 Bank A nodes 2 and 3 of the ring controller, which places outputs 1-24 in the **Out** 24-bit data register of node 2, and outputs 25-48 in the **Out** 24-bit register of node 3:



PMAC3 Style MACRO ICs allow bitwise mapping to user memory structure elements:

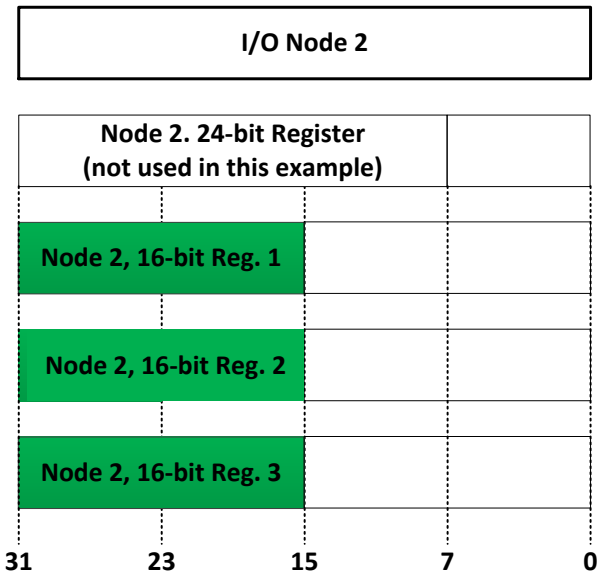
```
PTR Output1->Gate3[0].MacroOutA[2][0].8.1;
PTR Output2->Gate3[0].MacroOutA[2][0].9.1;
PTR Output3->Gate3[0].MacroOutA[2][0].10.1;
PTR Output4->Gate3[0].MacroOutA[2][0].11.1;
PTR Output5->Gate3[0].MacroOutA[2][0].12.1;
PTR Output6->Gate3[0].MacroOutA[2][0].13.1;
PTR Output7->Gate3[0].MacroOutA[2][0].14.1;
PTR Output8->Gate3[0].MacroOutA[2][0].15.1;
PTR Output9->Gate3[0].MacroOutA[2][0].16.1;
PTR Output10->Gate3[0].MacroOutA[2][0].17.1;
PTR Output11->Gate3[0].MacroOutA[2][0].18.1;
PTR Output12->Gate3[0].MacroOutA[2][0].19.1;
PTR Output13->Gate3[0].MacroOutA[2][0].20.1;
PTR Output14->Gate3[0].MacroOutA[2][0].21.1;
PTR Output15->Gate3[0].MacroOutA[2][0].22.1;
PTR Output16->Gate3[0].MacroOutA[2][0].23.1;
PTR Output17->Gate3[0].MacroOutA[2][0].24.1;
PTR Output18->Gate3[0].MacroOutA[2][0].25.1;
PTR Output19->Gate3[0].MacroOutA[2][0].26.1;
PTR Output20->Gate3[0].MacroOutA[2][0].27.1;
PTR Output21->Gate3[0].MacroOutA[2][0].28.1;
PTR Output22->Gate3[0].MacroOutA[2][0].29.1;
PTR Output23->Gate3[0].MacroOutA[2][0].30.1;
PTR Output24->Gate3[0].MacroOutA[2][0].31.1;
```

```
PTR Output25->Gate3[0].MacroOutA[3][0].8.1;
PTR Output26->Gate3[0].MacroOutA[3][0].9.1;
PTR Output27->Gate3[0].MacroOutA[3][0].10.1;
PTR Output28->Gate3[0].MacroOutA[3][0].11.1;
PTR Output29->Gate3[0].MacroOutA[3][0].12.1;
PTR Output30->Gate3[0].MacroOutA[3][0].13.1;
PTR Output31->Gate3[0].MacroOutA[3][0].14.1;
PTR Output32->Gate3[0].MacroOutA[3][0].15.1;
PTR Output33->Gate3[0].MacroOutA[3][0].16.1;
PTR Output34->Gate3[0].MacroOutA[3][0].17.1;
PTR Output35->Gate3[0].MacroOutA[3][0].18.1;
PTR Output36->Gate3[0].MacroOutA[3][0].19.1;
PTR Output37->Gate3[0].MacroOutA[3][0].20.1;
PTR Output38->Gate3[0].MacroOutA[3][0].21.1;
PTR Output39->Gate3[0].MacroOutA[3][0].22.1;
PTR Output40->Gate3[0].MacroOutA[3][0].23.1;
PTR Output42->Gate3[0].MacroOutA[3][0].24.1;
PTR Output42->Gate3[0].MacroOutA[3][0].25.1;
PTR Output43->Gate3[0].MacroOutA[3][0].26.1;
PTR Output44->Gate3[0].MacroOutA[3][0].27.1;
PTR Output45->Gate3[0].MacroOutA[3][0].28.1;
PTR Output46->Gate3[0].MacroOutA[3][0].29.1;
PTR Output47->Gate3[0].MacroOutA[3][0].30.1;
PTR Output48->Gate3[0].MacroOutA[3][0].31.1;
```


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

Power PMAC3 MI69/70 Mapping Example

MI69 / MI70 is used to transfer ACC-67E data through MACRO IC 0 Bank A node 2 of the ring controller, which places the data as illustrated on the right:



The 48 bits are divided into the three 16-bit registers of node 2, all in **Out** registers.

PMAC3 style MACRO ICs allow bitwise mapping to user memory structure elements:

```
PTR Output1->Gate3[0].MacroOutA[2][1].16.1;
PTR Output2->Gate3[0].MacroOutA[2][1].17.1;
PTR Output3->Gate3[0].MacroOutA[2][1].18.1;
PTR Output4->Gate3[0].MacroOutA[2][1].19.1;
PTR Output5->Gate3[0].MacroOutA[2][1].20.1;
PTR Output6->Gate3[0].MacroOutA[2][1].21.1;
PTR Output7->Gate3[0].MacroOutA[2][1].22.1;
PTR Output8->Gate3[0].MacroOutA[2][1].23.1;
PTR Output9->Gate3[0].MacroOutA[2][1].24.1;
PTR Output10->Gate3[0].MacroOutA[2][1].25.1;
PTR Output11->Gate3[0].MacroOutA[2][1].26.1;
PTR Output12->Gate3[0].MacroOutA[2][1].27.1;
PTR Output13->Gate3[0].MacroOutA[2][1].28.1;
PTR Output14->Gate3[0].MacroOutA[2][1].29.1;
PTR Output15->Gate3[0].MacroOutA[2][1].30.1;
PTR Output16->Gate3[0].MacroOutA[2][1].31.1;
PTR Output17->Gate3[0].MacroOutA[2][2].16.1;
PTR Output18->Gate3[0].MacroOutA[2][2].17.1;
PTR Output19->Gate3[0].MacroOutA[2][2].18.1;
PTR Output20->Gate3[0].MacroOutA[2][2].19.1;
PTR Output21->Gate3[0].MacroOutA[2][2].20.1;
PTR Output22->Gate3[0].MacroOutA[2][2].21.1;
PTR Output23->Gate3[0].MacroOutA[2][2].22.1;
PTR Output24->Gate3[0].MacroOutA[2][2].23.1;
```

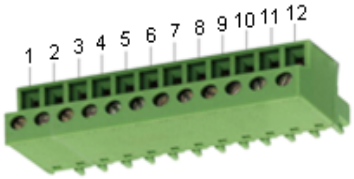
```
PTR Output25->Gate3[0].MacroOutA[2][2].24.1;
PTR Output26->Gate3[0].MacroOutA[2][2].25.1;
PTR Output27->Gate3[0].MacroOutA[2][2].26.1;
PTR Output28->Gate3[0].MacroOutA[2][2].27.1;
PTR Output29->Gate3[0].MacroOutA[2][2].28.1;
PTR Output30->Gate3[0].MacroOutA[2][2].29.1;
PTR Output31->Gate3[0].MacroOutA[2][2].30.1;
PTR Output32->Gate3[0].MacroOutA[2][2].31.1;
PTR Output33->Gate3[0].MacroOutA[2][3].16.1;
PTR Output34->Gate3[0].MacroOutA[2][3].17.1;
PTR Output35->Gate3[0].MacroOutA[2][3].18.1;
PTR Output36->Gate3[0].MacroOutA[2][3].19.1;
PTR Output37->Gate3[0].MacroOutA[2][3].20.1;
PTR Output38->Gate3[0].MacroOutA[2][3].21.1;
PTR Output39->Gate3[0].MacroOutA[2][3].22.1;
PTR Output40->Gate3[0].MacroOutA[2][3].23.1;
PTR Output41->Gate3[0].MacroOutA[2][3].24.1;
PTR Output42->Gate3[0].MacroOutA[2][3].25.1;
PTR Output43->Gate3[0].MacroOutA[2][3].26.1;
PTR Output44->Gate3[0].MacroOutA[2][3].27.1;
PTR Output45->Gate3[0].MacroOutA[2][3].28.1;
PTR Output46->Gate3[0].MacroOutA[2][3].29.1;
PTR Output47->Gate3[0].MacroOutA[2][3].30.1;
PTR Output48->Gate3[0].MacroOutA[2][3].31.1;
```


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

CONNECTOR PINOUTS AND WIRING

Terminal Block Connectors

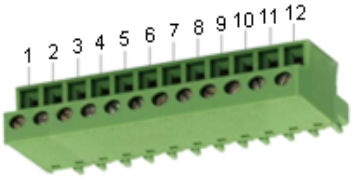
Top: Outputs

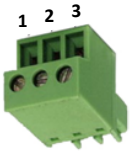
TB1 / TB2 Top (Outputs)		
Pin #	**Function**	**Description**
1	Output	Output #1 / 13
2	Output	Output #2 / 14
3	Output	Output #3 / 15
4	Output	Output #4 / 16
5	Output	Output #5 / 17
6	Output	Output #6 / 18
7	Output	Output #7 / 19
8	Output	Output #8 / 20
9	Output	Output #9 / 21
10	Output	Output #10 / 22
11	Output	Output #11 / 23
12	Output	Output #12 / 24

TB3 Top (Outputs Power)	
Pin #	**Function / Description**
1	+ 24V
2	+24V RETURN
3	+ 24V

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

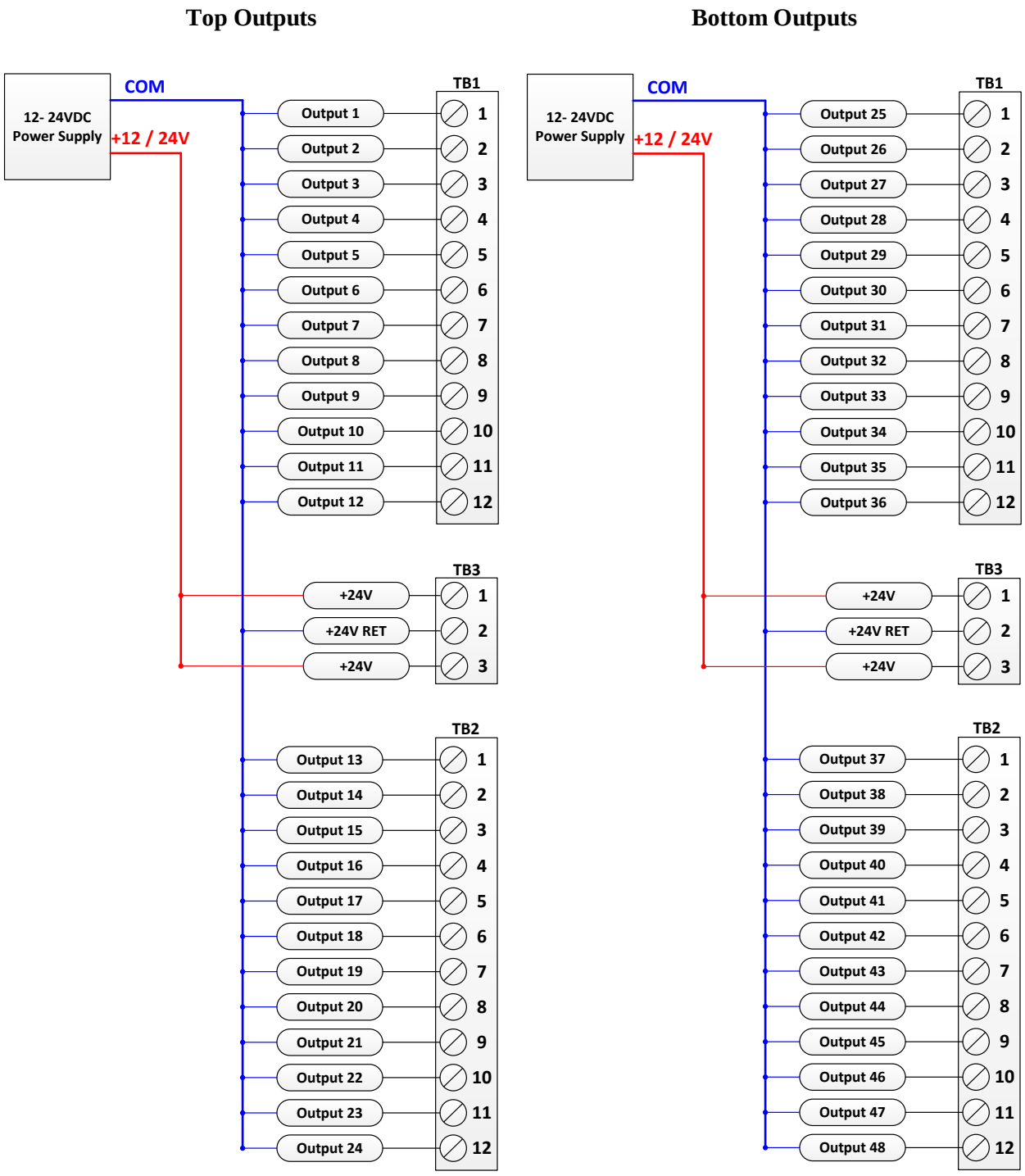
Bottom: Outputs

TB1 / TB2 Bottom (Outputs)		
Pin #	**Function**	**Description**
1	Output	Output #25 / 37
2	Output	Output #26 / 38
3	Output	Output #27 / 39
4	Output	Output #28 / 40
5	Output	Output #29 / 41
6	Output	Output #30 / 42
7	Output	Output #31 / 43
8	Output	Output #32 / 44
9	Output	Output #33 / 45
10	Output	Output #34 / 46
11	Output	Output #35 / 47
12	Output	Output #36 / 48

TB3 Bottom (Outputs Power)	
Pin #	**Function / Description**
1	+ 24V
2	+24V RETURN
3	+ 24V

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

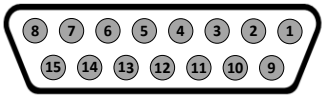
TB Wiring Diagram



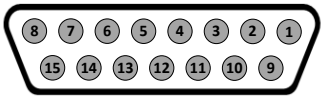
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D-Sub Connectors

Top: Outputs

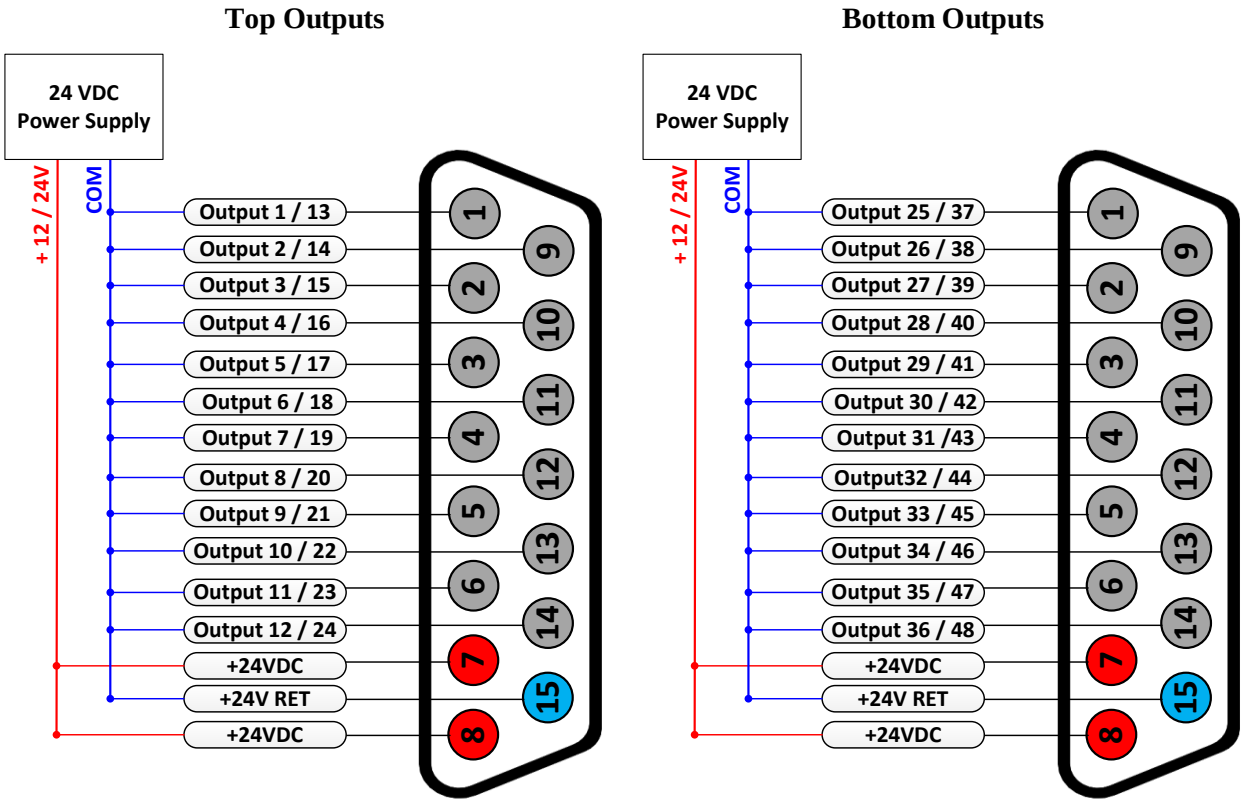
J1 / J2 Top: D-sub DA-15F Mating: D-sub DA-15M			
Pin #	**Function**	**J1**	**J2**
1	Output	Output #1	Output #13
2	Output	Output #3	Output #15
3	Output	Output #5	Output #17
4	Output	Output #7	Output #19
5	Output	Output #9	Output #21
6	Output	Output #11	Output #23
7	Power	+ 24V	
8	Power	+ 24V	
9	Output	Output #2	Output #14
10	Output	Output #4	Output #16
11	Output	Output #6	Output #18
12	Output	Output #8	Output #20
13	Output	Output #10	Output #22
14	Output	Output #12	Output #24
15	Return	+24V RETURN	

Bottom: Outputs

J1 / J2 Bottom: D-sub DA-15F Mating: D-sub DA-15M			
Pin #	**Function**	**J1**	**J2**
1	Output	Output #25	Output #37
2	Output	Output #27	Output #39
3	Output	Output #29	Output #41
4	Output	Output #31	Output #43
5	Output	Output #33	Output #45
6	Output	Output #35	Output #47
7	Power	+ 24V	
8	Power	+ 24V	
9	Output	Output #26	Output #38
10	Output	Output #28	Output #40
11	Output	Output #30	Output #42
12	Output	Output #32	Output #44
13	Output	Output #34	Output #46
14	Output	Output #36	Output #48
15	Return	+24V RETURN	

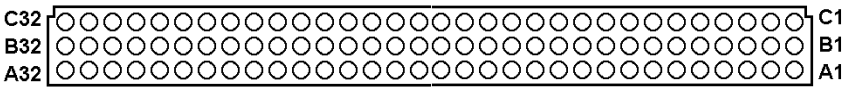
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

D-Sub Wiring diagram



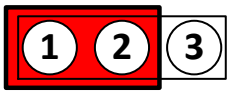
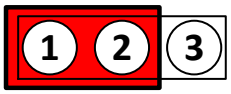
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

P1: UMAC Bus (UBUS) Connector

P1 UBUS (96-Pin Header)			
Pin #	**Row A**	**Row B**	**Row C**
1	+5 Vdc	+5 Vdc	+5 Vdc
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	-15 Vdc	PWRGND	+15 Vdc
31	GND	GND	GND
8	+5 Vdc	+5 Vdc	+5 Vdc

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

APPENDIX A: E-POINT JUMPERS

Jumper	Configuration	Default
E1: 	➢ 1 to 2 Turbo/Power/MACRO CPUs Revision 104 or newer ➢ 2 to 3 for Legacy MACRO CPUs Rev. 103 or older	Factory Set
E2: 	➢ 1 to 2 to sample at Servo Rate ➢ 2 to 3 to sample at Phase Rate	1 – 2

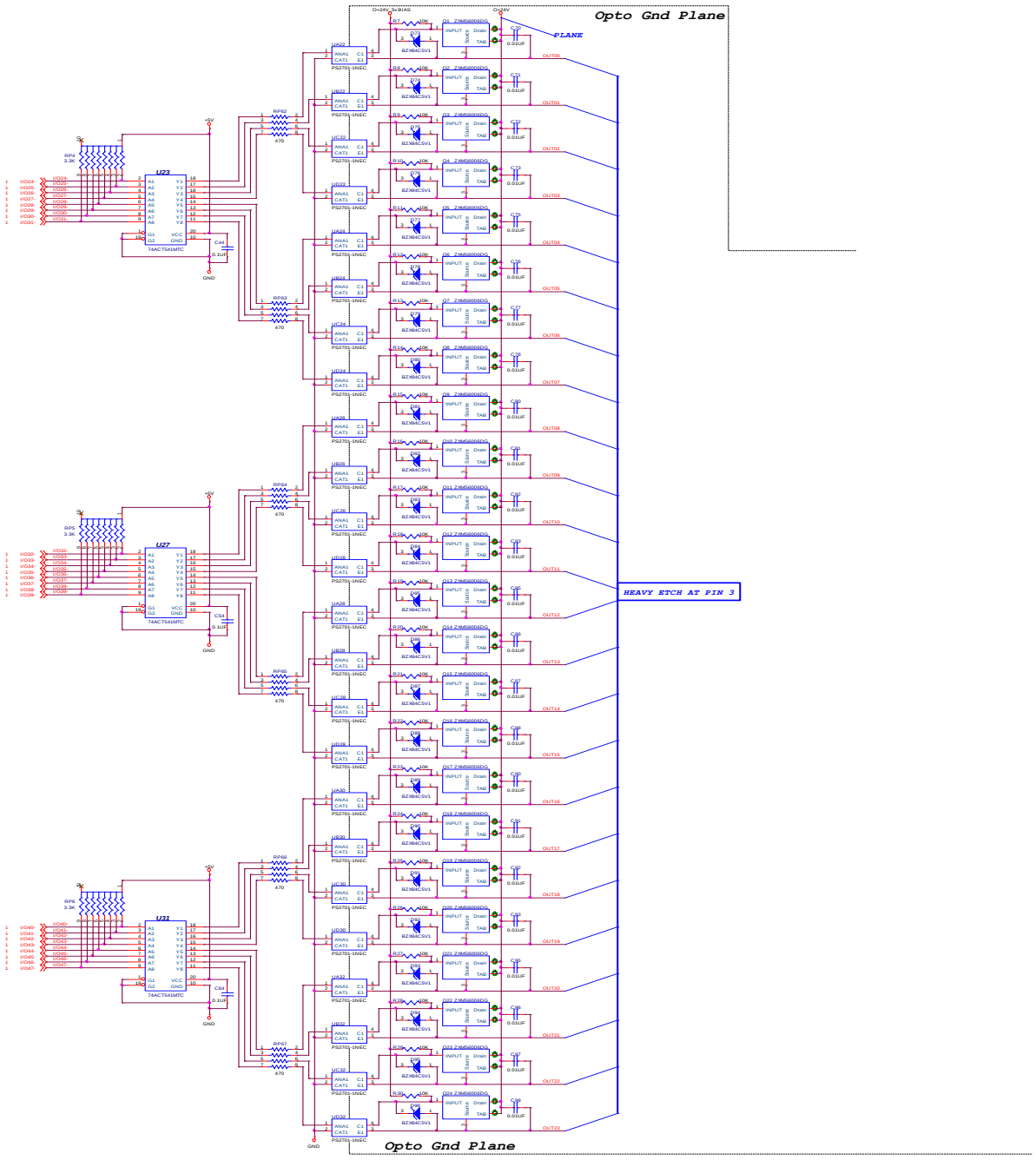


Note

Legacy MACRO CPU part numbers: 602804-100 through 602804-104.

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

APPENDIX B: SCHEMATICS



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

APPENDIX C: USING ACC-67E IN THE C ENVIRONMENT

This section shows how the ACC-67E can be programmed and used via the C programming language in Power PMAC.

ACC-67E C Library

Include the following header file (*acc67e.h*) into the Project:

```
#include <gplib.h>
#include <RtGpShm.h>   // Global Rt/Gp Shared memory pointers
//-------------------------------------------------------------
// The following is a projpp created file from the User defines
//-------------------------------------------------------------
#include "../../Include/pp_proj.h"

#define ON                    1       // Assumes that (logic high)=true
#define OFF                   0       // Assumes that (logic low)=false

unsigned int ACC67E_GetOutputState(unsigned int CardNumber, unsigned int OutputNumber);
void ACC67E_SetOutputState(unsigned int CardNumber, unsigned int OutputNumber, unsigned int
State);
```

Then put the following source code file (*acc67e.c*) into the same folder as the header above:

```
#include <gplib.h>
#include <RtGpShm.h>   // Global Rt/Gp Shared memory pointers
//-------------------------------------------------------------
// The following is a projpp created file from the User defines
//-------------------------------------------------------------
#include "../../Include/pp_proj.h"
#include "acc67e.h"

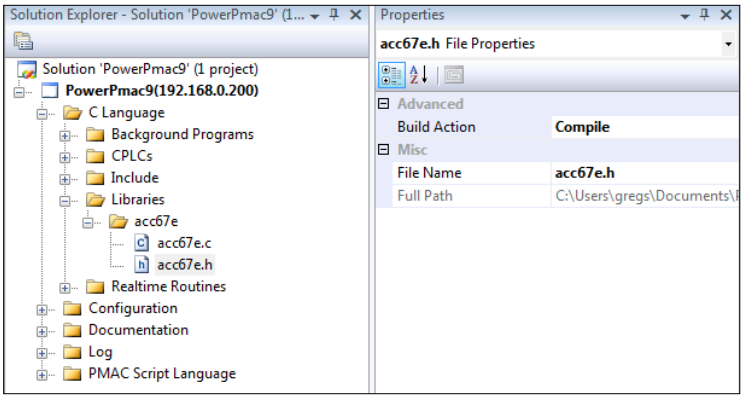
unsigned int ACC67E_GetOutputState(unsigned int CardNumber, unsigned int OutputNumber)
{
// CardNumber: Power PMAC I/O Card Index of this card (0-15)
// OutputNumber: Input Pin Number (1-48)
        volatile unsigned int *ioptr;
        OutputNumber--;
        ioptr = piom + pshm->OffsetCardIO[CardNumber]/4;
        ioptr += OutputNumber/8 + 3;
        // Shift all bits above the desired bit out
        // Then shift down to bit 0 and return
        return (unsigned int)((*ioptr << (31 - (OutputNumber%8 + 8))) >> 31);
void ACC67E_SetOutputState(unsigned int CardNumber, unsigned int OutputNumber, unsigned int
State)
{
// CardNumber: Power PMAC I/O Card Index of this card (0-15)
// OutputNumber: Input Pin Number (1-48)
// State: 1=ON, 0=OFF
        volatile unsigned int *ioptr;
        unsigned int HighBitInCorrectLocation;
        OutputNumber--;
        HighBitInCorrectLocation  = 1 << ((OutputNumber % 8) + 8);
        ioptr = piom + pshm->OffsetCardIO[CardNumber]/4;
        ioptr += OutputNumber/8 + 3;
        if(State == 1) // If the user wants the pin to be ON (high true)
        // Logical OR with the bit the user desires to activate
                *ioptr |= HighBitInCorrectLocation;
        else
        // Logical AND the register with a 0 in the desired location to bring the pin's state low
        // right shift to push out garbage in lowest 8 bits, then shift back up 8 bits to have
        // data in the proper location
                *ioptr &= (((~0)^HighBitInCorrectLocation) >> 8) << 8;
        return;
}
```


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

Locations for the C Files

Background C Programs

To use the above ACC-67E code in a Background C Program, make a new folder called "acc67e" in the C Language→Libraries folder under the Solution Explorer in the Power PMAC IDE. Then, put the code into this folder and make sure that the Build Action for acc67e.c under Properties is set to "Compile". The Solution Explorer should then look something like the screenshot to the right.

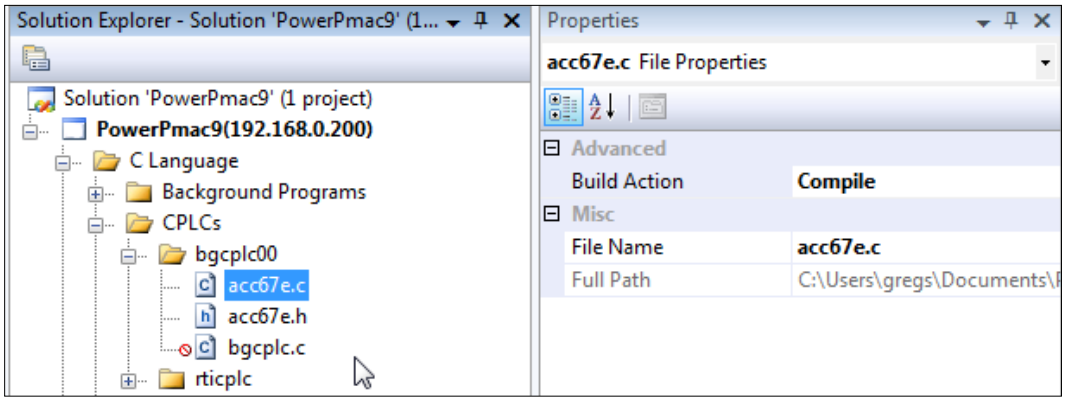


Then, in the Background Program, include the header file with the following preprocessor directive:

```
#include "../../Libraries/acc67e/acc67e.h"
```

Background CPLCs (BGCPLCs) and Real-Time CPLCs (RTICPLCs)

To use the above code in a Background CPLC (BGCPLC), put *acc67e.h* and *acc67e.c* into the same folder as the BGCPLC. For example, when using BGCPLC00, Solution Explorer might look like the following:

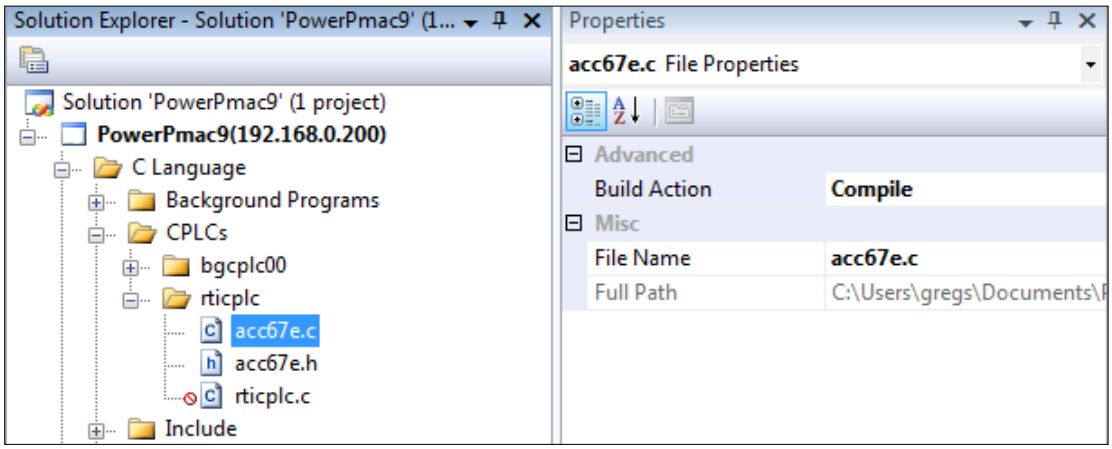


Then, include the header file with the following preprocessor directive:

```
#include "acc67e.h"
```


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

The same is true for RTICPLCs:



Function Descriptions

ACC65E_GetOutputState()

Description

This function returns the state of the output pin specified.

Input Arguments

CardNumber: The Power PMAC I/O Card Index of this card, given in the table in the Addressing ACC-65E section of this manual, corresponding to this card's I/O base address offset. Range: 0–15.

OutputNumber: The pin number of the output pin whose state the user desires to read. Outputs on ACC-65E are numbered 1–24, which is the range for this argument.

Return Value

Returns the state of the input pin:

0 = False/Low/Off
1 = True/High/On

ACC65E_SetOutputState()

Description

This function sets the state of the output pin specified.

Input Arguments

CardNumber: The Power PMAC I/O Card Index of this card, given in the table in the Addressing ACC-65E section of this manual, corresponding to this card's I/O base address offset. Range: 0–15.

OutputNumber: The pin number of the output pin whose state the user desires to set. Outputs on ACC-65E are numbered 1–24, which is the range for this argument.

Return Value

None; returns void.

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

Example

Setting Outputs in a BGCPLC

For a single ACC-67E at I/O base offset $B00000, this BGCPLC writes to outputs 1–48 using values stored in P200–P247, respectively. The states of output pins 1-48 are read and then stored in P300–P347, respectively, for sake of comparison or demonstration purposes.

```
#include <gplib.h>
#include <stdio.h>
#include <dlfcn.h>

#include "../../Include/pp_proj.h"
#include "acc67e.h"

#define ACC67E_CardNumber     1 // for base offset of $B00000

void user_plcc()
{
ArrayIndex]=(double)ACC67E_GetInputState(ACC67E_CardNumber,ChannelNumber);
               // Set the states of output pins 1-48 to the values stored in P200-P247
               ACC67E_SetOutputState(ACC67E_CardNumber,ChannelNumber,(unsigned int)pshm->P[200 +
ArrayIndex]);
               // Get the states of output pins 1-48 and store the states in P300-P347
               pshm->P[300 +
ArrayIndex]=(double)ACC67E_GetOutputState(ACC67E_CardNumber,ChannelNumber);
        }
        return;
}
```

To be able to run this at startup, make sure that the Run at Startup option is selected in the Properties menu of the folder containing this program. Make sure also that *acc67e.c* and *acc67e.h* have been placed into the "C language→Libraries→acc67e" folder .

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

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP