USER MANUAL

Accessory 59E

8-Channel 12-Bit ADC/DAC Board

3Ax-603494-xUxx

October 22, 2003

DELTA TAU
Data Systems, Inc.
NEW IDEAS IN MOTION ...

Single Source Machine Control *Power // Flexibility // Ease of Use*
21314 Lassen Street Chatsworth, CA 91311 // Tel. (818) 998-2095 Fax. (818) 998-7807 // www.deltatau.com

Copyright Information

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.

Table of Contents

INTRODUCTION

UMAC's Accessory 59E (ACC-59E) is both an analog data acquisition board as well as an analog output board. Accessory 59E is capable of processing 8 analog inputs as well as 8 general-purpose analog outputs.

The Accessory 59E's design features make it an ideal analog interface board. The analog input data can be used for monitoring and collecting signals from a variety of sensors and transducers. The analog outputs can be used wherever a general-purpose voltage output is needed from –20 to 20 volts. Through simple M-variable assignments, both the input and output values are easily accessible to use in motion programs, PLCs, and data collection.

The Analog-to-Digital Converter (ADC) unit used in ACC-59E is the MAX180 monolithic device manufactured by Maxim Integrated Products. These devices have 12-bit resolution with ± 1/2 LSB linearity specification. For more details of the ADC chips please refer to the data sheet published by the manufacturer:

> Document 19-3950; Rev 0, 6/91
> Complete, 8-Channel, 12-Bit Data Acquisition Systems
> Maxim Integrated Products
> 120 San Gabriel Drive
> Sunnyvale, CA 94086
> Phone: (408) 737-7600

The Digital-to-Analog Converter (DAC) unit used in ACC-59E is the DAC7625 manufactured by Burr-Brown Corporation. These devices are 12-bit quad voltage output digital-to-analog converters with guaranteed 12-bit monotonic performance from –40°C to +85°C. For more details about the DAC chips refer to the data sheet published by the manufacturer:

> DAC7624 and DAC7625
> "12-Bit Quad Voltage Output Digital-To-Analog Convertor"
> Burr-Brown Corporation
> 6730 S. Tucson Blvd.
> Tucson, AZ 85706
> Phone: (520) 746-1111

The A/D converter chips used on this accessory multiplex the data and therefore PMAC must address each channel to read them. Delta Tau has created automatic data transfers for both the 3U Turbo PMAC and the MACRO Station. This automatic method reads addresses for each channel and places the converted data automatically in memory locations accessible by the user.

For UMAC firmware 1.936 and above, 16 channels of analog inputs may be read to specified UMAC memory locations automatically. These registers can be monitored using M-Variables or read into UMAC's encoder conversion table for servo feedback control.

MACRO Station firmware 1.115 and above has implemented a similar procedure for interfacing to a single analog input card (ACC-59E or ACC-36E). When additional cards are used, data transfer methods must be implemented. These are explained in later sections of the manual.

CONNECTIONS

Connectors

Refer to the layout diagram of ACC-59E for the location of the connectors on the board.

P1

This connector is used for interface to UMAC's processor bus via the backplane of 3U rack. The signals, which are brought in through this connector, are buffered on board.

Top J1-J2

Through these connectors the analog signals are brought into ACC-59E. In addition the +/- 12 to 15 volt power supplies are brought out. These power supplies may be used in situations where a separate supply unit is not available for the analog transducers.

Note:

The two fuses limit the current drawn to 0.5 A on each supply line.

Bottom J1-J2

Through these connectors the analog output signals are supplied from the ACC-59E. In addition the +5 volt power supply is brought out.

J5

This connector is used for factory calibration. Do not use this connector.

JP1

This connector is used for factory calibration. Do not use this connector.

JP2

This connector is used for factory calibration. Do not use this connector.

TB1

This is a 4-pin terminal block, which provides the connection for power supply inputs to ACC-59E when it is used in a standalone configuration.

Note:

Do not use this connector when the card is sitting on the 3U rack.

E-Point Jumpers

Jumper	Config	Description	Settings	Default
E1	1-2	Turbo-PMAC/MACRO Select	1-2 for 3U Turbo PMAC and MACRO Stations Revisions 104 and Higher 2-3 for legacy MACRO Stations revisions 103 and earlier	Set by factory
J3	1-2	Bipolar/Unipolar DAC outputs	1-2 for Bi-Polar DAC outputs 2-3 for Uni-Polar DAC outputs	1-2

Address Select DIP Switch SW1

The switch one (SW1) settings will allow the user to select the base address of the ACC-59E. The base address is used for a reference of where to access the ADCs and the DACs in memory. The following two tables show the DIP switch settings for both the UMAC Turbo and the MACRO Station.

UMAC Turbo Switch Settings

Chip Select	UMAC Turbo Address	DIP Switch SW1 Position					
		6	**5**	**4**	**3**	**2**	**1**
CS10	Y:$78C00	ON	ON	ON	ON	ON	ON
	Y:$79C00	ON	ON	ON	OFF	ON	ON
	Y:$7AC00	ON	ON	OFF	ON	ON	ON
	Y:$7BC00	ON	ON	OFF	OFF	ON	ON
CS12	Y:$78D00	ON	ON	ON	ON	ON	OFF
	Y:$79D00	ON	ON	ON	OFF	ON	OFF
	Y:$7AD00	ON	ON	OFF	ON	ON	OFF
	Y:$7BD00	ON	ON	OFF	OFF	ON	OFF
CS14	Y:$78E00	ON	ON	ON	ON	OFF	ON
	Y:$79E00	ON	ON	ON	OFF	OFF	ON
	Y:$7AE00	ON	ON	OFF	ON	OFF	ON
	Y:$7BE00	ON	ON	OFF	OFF	OFF	ON
CS16	Y:$78F00	ON	ON	ON	ON	OFF	OFF
	Y:$79F00	ON	ON	ON	OFF	OFF	OFF
	Y:$7AF00	ON	ON	OFF	ON	OFF	OFF
	Y:$7BF00	ON	ON	OFF	OFF	OFF	OFF

MACRO Station Switch Settings

Chip Select	MACRO Station Address	DIP Switch SW1 Position					
		6	**5**	**4**	**3**	**2**	**1**
CS10	Y:$8800	ON	ON	ON	ON	ON	ON
	Y:$9800	ON	ON	ON	OFF	ON	ON
	Y:$A800	ON	ON	OFF	ON	ON	ON
	Y:$B800 ($FFE0*)	ON	ON	OFF	OFF	ON	ON
CS12	Y:$8840	ON	ON	ON	ON	ON	OFF
	Y:$9840	ON	ON	ON	OFF	ON	OFF
	Y:$A840	ON	ON	OFF	ON	ON	OFF
	Y:$B840 ($FFE8*)	ON	ON	OFF	OFF	ON	OFF
CS14	Y:$8880	ON	ON	ON	ON	OFF	ON
	Y:$9880	ON	ON	ON	OFF	OFF	ON
	Y:$A880	ON	ON	OFF	ON	OFF	ON
	Y:$B880 ($FFF0*)	ON	ON	OFF	OFF	OFF	ON
CS16	Y:$88C0	ON	ON	ON	ON	OFF	OFF
	Y:$98C0	ON	ON	ON	OFF	OFF	OFF
	Y:$A8C0	ON	ON	OFF	ON	OFF	OFF
	Y:$B8C0 ($FFF8*)	ON	ON	OFF	OFF	OFF	OFF
* To implement the alternate addresses $FFE0, $FFE8, $FFF0, and $FFF8, SW1 positions #6 and #5 must be set to OFF. This disables the other addresses and has a limit of four possible locations. It also limits the DAC output functionality on this card.							

Power Supply Connection

If ACC-59E is installed on the 3U-bus through P1, both the +5V supply and the ± 15V supplies are brought in through the bus. If ACC-59E is not powered through the 3U bus, then TB1 is used to bring in external power. The power supply requirements are approximately:

5V Supply	@200mA
+15V Supply	@90mA
-15V Supply	@100mA

Connection to Analog Signal Inputs

The analog signals are brought in from J1 and J2 on the top of the card. For a single-ended connection (using ADCx and GND), the voltage range should be from 0 to 20 volts for unipolar signals and -10V to +10V for bipolar signals. For a differential connection (using ADCx and ADCx/) the voltage range should between -10 and 10V for unipolar signals and -5V to 5V for bipolar signals. The ± 15V power supply is also brought out through this connector.

Note:

The two fuses limit the current drawn to 0.5 A on each supply line. For single-ended inputs, ground the complementary signals.

Power Supply Requirements

ACC-59E draws 100mA from each of its three supply voltages (+12 V, -12V and +5 V).

Power Requirements

5V	12V	-12V	Other 24V etc.
100 mA	100 mA	100 mA	N/A

Note:

Since the analog inputs are not optically isolated on this board, the ±12V supply to this board should not be from the same supply that is used for the UMAC's optically isolated analog outputs (DACs).

ACC-59E Fuse

Manufacturer	Specification	DT Part Number
Little Fuse	125V @ 0.5A	273.500

Adjustment Pots

There are 16 analog offset adjustment pots. These 12-turn pots are located at the top edge of the printed circuit board. From left to right R8 is for ADC 1, R16 is for ADC 2 etc., and R36 is for DAC1, R37 is for DAC2 etc.

R34 and R35 pots are for internal use only; they are used to adjust the reference voltages on the ADC chip.

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 IO cards. The ACC-59E 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 allows 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 IO	12	A
ACC-65E, ACC-66E ACC-67E, ACC-68E ACC-14E	16	General IO	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 just using only the type A UMAC cards or using only the type B UMAC cards in an application, the user does not have to 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 their 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 IO type. If the Type B card is a general IO 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-59E

If using an ACC-11E and ACC-59E, both cards cannot use the same Chip Select because the data from both cards will be overwritten by the other card.

Type A and Type B Example 2: ACC-11E and ACC-65E

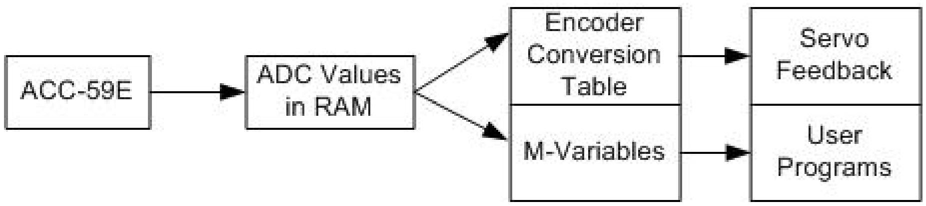
For this example, the two cards are allowed to share the same chip select because the ACC-65E is a general purpose I/O Type B card. The only restriction in doing this is that the ACC-65E must be considered the low-byte addressed card and the ACC-11E must be jumpered to either the middle or high bytes (jumper E6A-E6H).

USING ACC-59E WITH UMAC TURBO PMAC

Reading the analog data through ACC-59E is a simple procedure. There are two possible ways to read the analog inputs. PLCs can be written that monitor the ACC-59E board or UMAC's automatic ADC register read feature can be used.

The A/D converter chips used on this accessory multiplex the data and therefore, UMAC must address each channel for reading. The automatic ADC read addresses each ADC and copies the value into a pre-defined memory address. This copy is done every phase (110 μsec by default) clock for two ADC channels. Manually address the ADC chip and copy the values into memory locations using the manual ADC read method.

The automatic read feature in UMAC allows reading the analog signals as feedback devices or for normal data acquisition by having M-variables pointing to memory locations containing the information received by the automatic read feature. The block diagram below shows the analog data flow for servo feedback and user programs.



The following block diagram shows the information flow from ACC-59E to the user programs using the manual ADC read method.

Enabling UMAC Turbo ADC Transfer (Automatic ADC Read Method)

Just like the standard Turbo PMAC2, the UMAC Turbo allows the use of the automatic copy feature to simplify the reading of the A/D converted data. Using this method to read allows the use of the data for both data acquisition and closing servo loops. To enable the feature, I5060, I5061-I5076, and I5081-I5096 must be set as specified by the Turbo PMAC Software Reference. Up to 32 ADCs or 16 ADC pairs can be read in this fashion when using an ACC-36. However, due to the design of the ACC-59E, only 16 ADCs may be read in this fashion and they are not grouped in pairs.

The multiplexed data from the ADC returns to the PMAC memory address as a 24-bit word. The lower 12-bits contain ADC0 through ADC7 on the 59E. The upper 12 bits of this word are not used by the ACC-59E. After the data is returned it is copied into the upper 12 bits of the addresses shown below. The data is copied automatically, as follows:

I-Variable	"Low" ADC Result	ACC-59E
I5061	Y:$003400 (ADC1)	1st
I5062	Y:$003402 (ADC2)	1st
I5063	Y:$003404 (ADC3)	1st
I5064	Y:$003406 (ADC4)	1st
I5065	Y:$003408 (ADC5)	1st
I5066	Y:$00340A (ADC6)	1st
I5067	Y:$00340C (ADC7)	1st
I5068	Y:$00340E (ADC8)	1st
I5069	Y:$003410 (ADC1)	2nd
I5070	Y:$003412 (ADC2)	2nd
I5071	Y:$003414 (ADC3)	2nd
I5072	Y:$003416 (ADC4)	2nd
I5073	Y:$003418 (ADC5)	2nd
I5074	Y:$00341A (ADC6)	2nd
I5075	Y:$00341C (ADC7)	2nd
I5076	Y:$00341E (ADC8)	2nd

For the ACC-59E, I5060 controls the number of multiplexed A/D converters that are processed and de-multiplexed into individual registers. If I5060 is set to 0, none of these A/D converters is processed automatically.

If I5060 is set to a value greater than 0, it specifies the number of ADCs in the automatic processing ring. Each phase clock cycle, one pair is processed and the values copied into image registers in RAM.

I5061 through I5076 control the addresses of the multiplexed A/D converters read in the A/D ring table, as enabled by I5060. These I-variables contain offsets from the starting Turbo PMAC address $078800 where these ADCs can reside. The base address of the ACC-59E is defined by the SW1 setting.

I5081 through I5096 contain the convert codes written to the multiplexed A/D converters that are read in the A/D ring table, as enabled by I5060. The convert codes control which of the multiplexed ADCs at the address is to be read, and the range of the analog input for that ADC. The ADCs can be on-board the Turbo PMAC with Option 12 and 12A, or off-board with an ACC-36P/V/E or ACC-59E.

I5081-I5096 are 24-bit values, represented by six hexadecimal digits. Legitimate values are of the format $00*m*00*n*, where *m* and *n* can take any hex value from 0 through F.

For the ACC-59E with a 3U Turbo PMAC2, the *m* value is not used; leave it at zero.

For the ACC-59E with a UMAC Turbo, the *n* value determines which of the inputs ANAI00 to ANAI07 and how it is to be converted, according to the following formulas:

```
n = ANAI#              ; 0V to +20V unipolar input
n = ANAI#+8            ; -10V to +10V bipolar input
```

For example, to read ANAI 2 from ACC-59E, as a +/-2.5V differential, bipolar input, into the first slot in the ring, n would be set to A (10), so I5081 would be set to $00000A.

Analog Input Automatic Read Data Acquisition Example

Set up the UMAC Turbo to read channels 0, 1, 2 and 3 as unipolar converted signals and read channels 4, 5, 6 and 7 as bipolar signals. Assume the switch settings are set for a base address of $078C00 ($78800 + $400).

```
I5060=8        ;copy 8 ADCs
I5061=$000400 ;ADC0 and ADC8 are referenced to $078800 + $000400
($078C00)
I5062=$000400 ;ADC1 and ADC9 are referenced to $078800 + $000400
($078C00)
I5063=$000400 ;ADC2 and ADC10 are referenced to $078800 + $000400
($078C00)
I5064=$000400 ;ADC3 and ADC11 are referenced to $078800 + $000400
($078C00)
I5065=$000400 ;ADC4 and ADC12 are referenced to $078800 + $000400
($078C00)
I5066=$000400 ;ADC5 and ADC13 are referenced to $078800 + $000400
($078C00)
I5067=$000400 ;ADC6 and ADC14 are referenced to $078800 + $000400
($078C00)
I5068=$000400 ;ADC7 and ADC15 are referenced to $078800 + $000400
($078C00)
I5081=$000000 ;ADC0 unipolar ADC8 is bi-polar
I5082=$000001 ;ADC1 unipolar ADC9 is bi-polar
I5083=$000002 ;ADC2 unipolar ADC10 is bi-polar
I5084=$000003 ;ADC3 unipolar ADC11 is bi-polar
I5085=$00000C ;ADC4 unipolar ADC12 is bi-polar
I5086=$00000D ;ADC5 unipolar ADC13 is bi-polar
I5087=$00000E ;ADC6 unipolar ADC14 is bi-polar
I5088=$00000F ;ADC7 unipolar ADC15 is bi-polar
```

```
M5061->Y:$003400,12,12,u   ;channel 0 A to D as unipolar
M5062->Y:$003402,12,12,u   ;channel 1 A to D as unipolar
M5063->Y:$003404,12,12,u   ;channel 2 A to D as unipolar
M5064->Y:$003406,12,12,u   ;channel 3 A to D as unipolar
M5065->Y:$003408,12,12,s   ;channel 4 A to D as bipolar
M5066->Y:$00340A,12,12,s   ;channel 5 A to D as bipolar
M5067->Y:$00340C,12,12,s   ;channel 6 A to D as bipolar
M5068->Y:$00340E,12,12,s   ;channel 7 A to D as bipolar
```

Note:

To start the automatic data transfer process, save and restart the 3U Turbo PMAC and then read the m-variables associated with the ADC channel.

ACC-59E Servo Feedback Use Example for UMAC Turbo

To process the A/D information in the encoder conversion table, do the following. For this example, ADC0 will be processed from location Y:$3400,0,12 in the ECT as a parallel unsigned entry.

It is assumed that the ninth entry of the encoder conversion table (ECT) is available.

```
I8008=$203400         ;read location Y:$3400 (ECT location $3509)
I8009=$00C00C         ;read 12 bits shifted 12 bits from Y:$3400
(ECT location $350A)
```

The axis to be used is specified by x.

```
Set Ix03=$350A  ;position feedback address
Set Ix04=$350A  ;velocity feedback address
```

ACC-59E Power-On Position for Turbo PMAC2

As of 9/27/2000, Delta Tau firmware does not support upper 12-bit word power-on position reads. However, the position register can be forced to read the appropriate power-on value using the position offset register.

```
M164->D:$00CC   ;motor 1 offset position register (1/(32*Ix08))
M264->D:$014C   ;motor 2 offset position register (1/(32*Ix08))
M594->D:$01CC   ;motor 3 offset position register (1/(32*Ix08))
M464->D:$024C   ;motor 4 offset position register (1/(32*Ix08))
M564->D:$02CC   ;motor 5 offset position register (1/(32*Ix08))
M664->D:$034C   ;motor 6 offset position register (1/(32*Ix08))
M764->D:$03CC   ;motor 7 offset position register (1/(32*Ix08))
M864->D:$044C   ;motor 8 offset position register (1/(32*Ix08))
M5061->Y:$003400,12,12,u   ;channel 0 A to D as unipolar
M5062->Y:$003402,12,12,u   ;channel 1 A to D as unipolar
M5063->Y:$003404,12,12,u   ;channel 2 A to D as unipolar
M5064->Y:$003406,12,12,u   ;channel 3 A to D as unipolar
M5065->Y:$003408,12,12,u   ;channel 4 A to D as unipolar
M5066->Y:$00340A,12,12,u   ;channel 5 A to D as unipolar
M5067->Y:$00340C,12,12,u   ;channel 6 A to D as unipolar
M5068->Y:$00340E,12,12,u   ;channel 7 A to D as unipolar
```

A PLC could be written to read the ADC into the position offset registers at power up:

```
OPEN PLC 25 CLEAR

I5111=1000*8388608/i10 ;1000 msec delay to ensure data is read properly

While (i5111>0) endwhile
M164=m5061*32*i108    ;set power on position offset to m5061 for mtr1
M264=m5062*32*i108    ;set power on position offset to m5062 for mtr2
M594=m5063*32*i108    ;set power on position offset to m5063 for mtr3
M464=m5064*32*i108    ;set power on position offset to m5064 for mtr4
M564=m5065*32*i108    ;set power on position offset to m5065 for mtr5
M664=m5066*32*i108    ;set power on position offset to m5066 for mtr6
M764=m5067*32*i108    ;set power on position offset to m5067 for mtr7
M864=m5068*32*i108    ;set power on position offset to m5068 for mtr8
Disable plc25
close
```

Manual ADC Read Method with UMAC Turbo

When using the manual ADC read method, address the ADC channel, and then copy the contents of the ADC register into a UMAC memory location (usually an M-variable). This can be accomplished with a PLC. If a PLC program is written to monitor the data, first define up to two M-variables for each ACC-59E board. Once the M-variables have been defined, UMAC's PLC programs may be used to initialize the analog-to-digital conversion process.

M-Variable Definitions for Manual Read Method

Define two M-variables for an eight channel ACC-59E and three M-variables for an ACC-59E with its Option 1. A 24-bit wide unsigned integer M-variable must be pointed to the base address of the board. Next, define two more M-variables, each 12-bits wide and pointed to the same base address. For bipolar signals (-10 to +10V single-ended and +5V to -5V differential), these M-variables must be defined as signed integers. For unipolar signals (0 to +20V single-ended or 0 to +10V differential), define them as unsigned. For example, if the base address is at Y:$78C00 (assuming CS10 is used), then define the three M-variables as follows:

For bipolar signals:

```
M1000->Y:$78C00,24       ;M-variable for Conversion Channel Select
M1001->Y:$78C00,0,12,s   ;M-variable for Read Data for channels 1 to 8
```

For unipolar signals:

```
M1000->Y:$78c00,24       ;M-variable for Conversion Channel Select
M1001->Y:$78c00,0,12,u   ;M-variable for Read Data for channels 1 to 8
```

Note:

The address Y:$78C00 is the same for both M-variables.

Analog Input Manual Read Data Acquisition Example

Initialize the analog-to-digital conversion process by writing into the ADC registers pointed to by Conversion Channel Select M-variable (M1000 in the above example). This process takes a few microseconds and because of the processing speed of Turbo, a small delay may be necessary after initializing the ADC process. Afterwards, the converted data may be read through the Read Data M-variables (M101 & M102 in the above example). The data written into the Conversion Channel Select M-variable determines both the input channel and the conversion type (unipolar Vs bipolar). This is shown in the following table:

Selection of Analog Input Channels for Analog-to-Digital Conversion

Base Address Input[1]	Selected Analog Input Channels	Polarity	Single-ended Range[2] (volts)	Differential Range[3] (Volts)
0	1	Unipolar	0 to 20	0 to 10
1	2	Unipolar	0 to 20	0 to 10
2	3	Unipolar	0 to 20	0 to 10
3	4	Unipolar	0 to 20	0 to 10
4	5	Unipolar	0 to 20	0 to 10
5	6	Unipolar	0 to 20	0 to 10
6	7	Unipolar	0 to 20	0 to 10
7	8	Unipolar	0 to 20	0 to 10
8	1	Bipolar	-10 to 10	-5 to 5
9	2	Bipolar	-10 to 10	-5 to 5
10	3	Bipolar	-10 to 10	-5 to 5
11	4	Bipolar	-10 to 10	-5 to 5
12	5	Bipolar	-10 to 10	-5 to 5
13	6	Bipolar	-10 to 10	-5 to 5
14	7	Bipolar	-10 to 10	-5 to 5
15	8	Bipolar	-10 to 10	-5 to 5

[1] The base address is selected using SW1. The value in this column would be the value given to M100 in the above example.

[2] For single-ended wiring use ADCx input and AGND return.

[3] For differential wiring use ADCx and ADCx/ inputs.

Reading Data through PLC Programs for Manual Read Method

For this example, convert channels 4 and 2 as unipolar inputs and read channel 1 as bipolar in the PLC 10 program:

```
M100 -> Y:$78c00,24       ;M-variable for Conversion Channel Select
M101 -> Y:$78c00,0,12,u   ;M-variable for Read Data for channels 1 to 8
OPEN PLC 10
CLEAR
.
.
.
M100=3     ;convert channel 4 as unipolar
I5111=1
While(i5111>0) endwhile
P104=M101  ;P100 now contains converted chan. 4 data
M100=1     ;convert channel 2 as unipolar
I5111=1
While(i5111>0) endwhile
P110=M102  ;P100 now contains converted chan. 10 dat
M100=8      ;convert channels 1 as bipolar
I5111=1
While(i5111>0) endwhile
P101=M101   ;P101 now contains converted chan. 1 data
.
.
CLOSE
```

UMAC Turbo Analog Output Configuration

To use the general-purpose DAC outputs from the ACC-59E, point an M-Variable to the proper location in memory. From that point, then write a value into memory and output a proportional voltage to the J1 and J2 connectors on the bottom of the rack.

The output addresses for DACs 1-8 are shown below:

DAC Channel #	Address Location[1]	Starting Bit	Bit Width
1	Base + 8	0	12
2	Base + 9	0	12
3	Base + 10	0	12
4	Base + 11	0	12
5	Base + 8	12	12
6	Base + 9	12	12
7	Base + 10	12	12
8	Base + 11	12	12
[1] The Base address is configured through SW1.			

When the ACC-59E is configured for unipolar mode, the voltage range is from 0 to 20V. A value in the address of DAC 1 of 0 will correspond to 0V and a value of 4095 will correspond to about 20V.

When the ACC-59E is configured for bipolar mode, the voltage range is from -20 to 20V. A value in the address of DAC 1 of 0 will correspond to -20V and a value of 4095 ($FFF) will correspond to about 20V.

The following is a list of suggested of M-Variables to access the appropriate DAC output registers for an ACC-59E with a base address of Y:$78c00:

```
M8101->Y:$78c08,0,12        ;DAC #1
M8102->Y:$78c09,0,12        ;DAC #2
M8103->Y:$78c0a,0,12        ;DAC #3
M8104->Y:$78c0b,0,12        ;DAC #4
M8105->Y:$78c08,12,12       ;DAC #5
M8106->Y:$78c09,12,12       ;DAC #6
M8107->Y:$78c0a,12,12       ;DAC #7
M8108->Y:$78c0b,12,12       ;DAC #8
```

ACC-59E Configuration Example for UMAC Turbo

Base Address: $78c00

Using ADC Automatic Read Method

```
I5060=8          ;copy 8 ADC pairs automatically

;59E analog inputs slot pointers
I5061=$000400    ;ADC0 is referenced to $078800 + $00400 ($078C00)
I5062=$000400    ;ADC1 is referenced to $078800 + $00400 ($078C00)
I5063=$000400    ;ADC2 is referenced to $078800 + $00400 ($078C00)
I5064=$000400    ;ADC3 is referenced to $078800 + $00400 ($078C00)
I5065=$000400    ;ADC4 is referenced to $078800 + $00400 ($078C00)
I5066=$000400    ;ADC5 is referenced to $078800 + $00400 ($078C00)
I5067=$000400    ;ADC6 is referenced to $078800 + $00400 ($078C00)
I5068=$000400    ;ADC7 is referenced to $078800 + $00400 ($078C00)

;59E convert codes
I5081=$00008     ;ADC0 is bi-polar
```

```
I5082=$00009     ;ADC1 is bi-polar
I5083=$0000a     ;ADC2 is bi-polar
I5084=$0000b     ;ADC3 is bi-polar
I5085=$0000c     ;ADC4 is bi-polar
I5086=$0000d     ;ADC5 is bi-polar
I5087=$0000e     ;ADC6 is bi-polar
I5088=$0000f     ;ADC7 is bi-polar

;59E M-Variables
M1->Y:$003400,12,12,s      ;channel 0 A to D as bipolar
M2->Y:$003402,12,12,s      ;channel 1 A to D as bipolar
M3->Y:$003404,12,12,s      ;channel 2 A to D as bipolar
M4->Y:$003406,12,12,s      ;channel 3 A to D as bipolar
M5->Y:$003408,12,12,s      ;channel 4 A to D as bipolar
M6->Y:$00340A,12,12,s      ;channel 5 A to D as bipolar
M7->Y:$00340C,12,12,s      ;channel 6 A to D as bipolar
M8->Y:$00340E,12,12,s      ;channel 7 A to D as bipolar
```

DAC Setup Using M-Variable Pointers

```
;Value= 0  Volts= -20
;Value= 2047     Volts= 0
;Value= 4096     Volts= +20

m11->Y:$78c08,0,12         ;Bipolar DAC output   #1
m12->Y:$78c09,0,12         ;Bipolar DAC output   #2
m13->Y:$78c0a,0,12         ;Bipolar DAC output   #3
m14->Y:$78c0b,0,12         ;Bipolar DAC output   #4
m15->Y:$78c08,12,12        ;Bipolar DAC output   #5
m16->Y:$78c09,12,12        ;Bipolar DAC output   #6
m17->Y:$78c0a,12,12        ;Bipolar DAC output   #7
m18->Y:$78c0b,12,12        ;Bipolar DAC output   #8
```

USING ACC-59E WITH UMAC MACRO

Configuring an Acc-59E card in a UMAC MACRO system is similar to configuring the card for a UMAC turbo system with an added layer of communication for the MACRO ring transfers. For the analog outputs, there is really only one convenient method to transfer the data across the ring. Once that is configured, the DAC outputs are accessed through simple M-Variable definitions just like the UMAC turbo. There is more freedom when configuring the analog inputs. The first thing that must be done is transferring the multiplexed data into MACRO Station memory locations. Once that is accomplished, there are basically three methods of transfer that can be chosen to transfer the data. Usually, this decision is made based on the use of the inputs, whether or not the input data is used for servo feedback or low/high priority inputs. In order to get an understanding of how the transfers take place, the analog input configuration will be explained first, followed by the analog output setup.

Analog Input Configuration with UMAC MACRO

Regardless of the method used to transfer to the ADC data across the MACRO ring, the automatic copy method is required to convert the data from the multiplexed form into individual MACRO Station memory locations. Unlike certain freedoms in a UMAC Turbo system, this is the only convenient method to access the data for MACRO.

Note:

To implement the automatic copy method on the ACC-59E, the MACRO Station must have firmware version 1.15 or greater.

Enabling ADC Automatic Read

To enable the MACRO ADC automatic read, set three MI variables at the MACRO Station – MI987, MI988, and MI989. These three variables tell the MACRO Station to copy the multiplexed data to pre-set memory locations in the MACRO Station.

MI987

When the MI987 variable is set to 1, it will enable the automatic ADC transfer. If MI987 is set to zero, the ADC transfer will not take place.

MI988

The MI988 variable controls whether the optional on-board A/D converters are set up for unipolar (0 to +20V) or bipolar (-10 to +10V) inputs. MI988 consists of eight bits; each bit controls the setup of the A/D converters. A value of 0 in the bit sets up the A/D converters for unipolar inputs; a value of 1 in the bits sets up the A/D converters for bipolar inputs.

The following table shows which bits of MI988 control which A/D converters:

MI988 Bit #	Hex Bit Value	ADC
0	1	ANAI00
1	2	ANAI01
2	4	ANAI02
3	8	ANAI03
4	10	ANAI04
5	20	ANAI05
6	40	ANAI06
7	80	ANAI07

MI989

The MI989 variable specifies the memory location to start the ADC transfer from (the address dip switch setting). For example, if the switch settings specified $8800 as the address setting, set MI989 equal to $8800 to properly read the ADC inputs.

Example: At MACRO Station 0, transfer ADC channels 0, 1, 2, 3, 8, 9, 10, and 11 as unipolar and channels 4, 5, 6, 7, 12, 13, 14, and 15 as bipolar from address setting $8840.

```
MS0,MI987=1           ;enable ADC transfer
MS0,MI988=$F0         ;ADC channels 0, 1, 2 and 3 as unipolar and
                      ;channels 4, 5, 6 and 7 as bipolar
MS0,MI989=$8840       ;address dip switch set to $8840
```

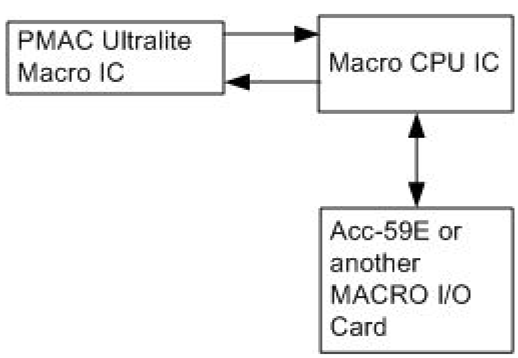
The data from the ADC returns to the MACRO Station memory address as a 24-bit word. The lower 12-bits will contain ADC0 through ADC7 and the upper 12-bits of this word is not used by ACC-59E. The data is copied automatically, as follows:

Channel	Location
ADC0	Y:$0200,0,12
ADC1	Y:$0201,0,12
ADC2	Y:$0202,0,12
ADC3	Y:$0203,0,12
ADC4	Y:$0204,0,12
ADC5	Y:$0205,0,12
ADC6	Y:$0206,0,12
ADC7	Y:$0207,0,12

Transferring Data Across the MACRO Ring

Once the data is translated from the multiplexed form into MACRO Station memory locations, the only thing left to do is transfer the data to the Ultralite so it is available through either the I/O node addresses or through the MACRO encoder conversion table and across the servo node addresses. The latter is used when the analog inputs are used for servo feedback.

For the ACC-59E there are two methods to transfer the data back to the PMAC Ultralite over the I/O nodes. An automatic method could be used with MACRO variables MI173, MI174, and MI,I175 or the data could be transferred using the standard I/O transfer method. The standard data transfer flow for both cases is shown below.



MACRO CPU I/O Node 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:$C0AC	X:$C0AD, X:$C0AE, X:$C0AF
10	X:$C0B0	X:$C0B1, X:$C0B2, X:$C0B3
11	X:$C0B4	X:$C0B5, X:$C0B6, X:$C0B7

PMAC2 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:$078459, 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:$079459, 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:$07A459, 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
(IC3) 10	58	X:$07B430	X:$07B431, X:$07B432, X:$07B433
(IC3) 11	59	X:$07B434	X:$07B435, X:$07B459, X:$07B437

To read the inputs from the MACRO Station of the first 24-bit I/O node address of node 2 (X:$C0A0), 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) can be used to monitor the inputs for either the Ultralite or Turbo Ultralite respectively.

MACRO ACCESSORY I/O DATA TRANSFER

There is two methods that can be used to implement I/O data transfer on MACRO: automatic transfer and self-configured transfers.

Automatic Transfer

The automatic data transfer uses MACRO Station I-variables to send the information from the ACC-59E to the MACRO Station I/O node address. Once the information is at the MACRO Station node address, the information is used at the PMAC Ultralite. There are three MACRO Station I-variables to set up this automatic transfer:

MSn,MI173	Up to six 12-bit transfers for A/D inputs 1-6
MSn,MI174	Could use interchangeably with MSn,MI173
MSn,MI175	Up to four 12-bit transfers for A/D inputs 7-8, must be consecutive.

Use these MI-variables to send A/D information to the node addresses automatically, as described in the following paragraphs.

MI173

MI173 specifies the registers used in A/D transfer between two MACRO nodes. It transfers the lower A/D's, three at a time, to three 16-bit nodes. The upper four bits are set to zero. The individual digits are specified as follows:

Digit #	Valid Values	Description
1	1 – 2	Specifies the number of nodes to be used.
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	(Reserved for future use)
8	0	(Reserved for future use)
9-12	$200 … $205	MACRO Station Y A/D Address (Bits 00 … 11)

When this function is active, the MACRO Station copies values from the Y: Address specified in digits 9 - 12 into X: Address specified in digits 3-6, three at a time, up to a total of six. The move to the specified node address assumes it to be the first X: memory 16 bit node register.

MI73 Example: Transfer the first six A/D converter channels (channels 1-6) using MI73.

```
I173 = $20C0A1000200
```

A/D Channel	A/D Location	MACRO Node Address	Ultralite M-Variable	Turbo Ultralite M- Variable
ADC1	Y:$200,0,12	X:$C0A1,8,12	M1001->X:$C0A1,8,12	M1001->X:$078421,8,12
ADC2	Y:$201,0,12	X:$C0A2,8,12	M1002->X:$C0A2,8,12	M1002->X:$078422,8,12
ADC3	Y:$202,0,12	X:$C0A3,8,12	M1003->X:$C0A3,8,12	M1003->X:$078423,8,12
ADC4	Y:$203,0,12	X:$C0A5,8,12	M1004->X:$C0A5,8,12	M1004->X:$078425,8,12
ADC5	Y:$204,0,12	X:$C0A6,8,12	M1005->X:$C0A6,8,12	M1005->X:$078426,8,12
ADC6	Y:$205,0,12	X:$C0A7,8,12	M1006->X:$C0A7,8,12	M1006->X:$078427,8,12

Now the M-variables defined can be used to process the data in the PMAC Ultralite.

I175

I175 specifies the registers used in A/D transfer between one to two MACRO nodes. It transfers the upper and lower A/Ds to a 24-bit node with a maximum of two 24-bit nodes. The individual digits are specified as follows:

Digit #	Valid Values	Description
1:	1 - 2	Specifies the number of nodes to be used.
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 24-bit register
7	0	(Reserved for future use)
8	0	(Reserved for future use)
9-12	$200 … $207	MACRO Station Y A/D Address (Bits 00 … 23)

When this function is active, the MACRO Station copies values from the Y: Address specified in digits 9 - 12 into X: Address specified in digits 3-6 one at a time up to a total of two. The move to the specified node address assumes it to be the X: memory of a 24-bit node register.

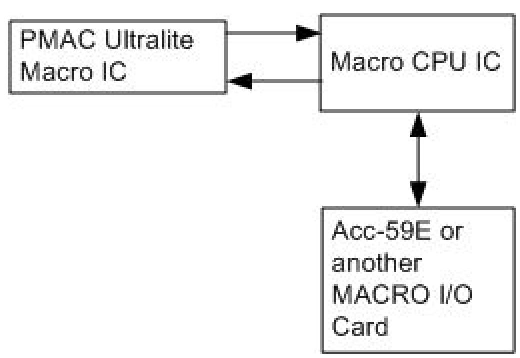
MI175 Example: Transfer the A/D converter channels 7 and 8 using MI75.

```
I175 = $20C0A0000206
```

A/D Channel	A/D Location	MACRO Node Address	Ultralite M- Variable	Turbo Ultralite M- Variable
ADC7	Y:$206,0,12	X:$C0A0,0,12	M1007->X:$C0A0,0,12	M1007->X:$078420,0,12
ADC8	Y:$207,0,12	X:$C0A4,0,12	M1008->X:$C0A4,0,12	M1008->X:$078424,0,12

Self Configured Data Transfer via the I/O Nodes

The MACRO Station also transfers data back to the Ultralite from any MACRO station memory location. This function is useful to read the 12-bit A/D converters, transferring data from either Gate1B or Gate 2B, which are not transferred automatically, or any other location for verification or troubleshooting purposes.



The data transfer process uses MI20 and MI21-MI68 to enable this function. Since the I/O nodes are used, MI975, MI19, and the Ultralite I/O node activation I-variables must be set to appropriate values. (See the MACRO Station Software Reference manual.)

MI20

MI20 controls which of 48 possible data transfer operations are performed at the data transfer period set by MI19. MI20 is a 48-bit value; each bit controls whether the data transfer specified by one of the variables MI21 through MI68 is performed.

```
MI20 = $1  ;transfer MI21
MI20 = $3  ;transfer MI21 and MI22
MI20 = $F  ;transfer MI21, MI22, MI23, and MI24
```

Hex	0	0	0	0	0	0	0	0	0	0	0	F

MI21 through MI68

MI21 through MI68 are 48-bit addresses describing the transfer of data from the desired memory location to the MACRO Station I/O node location. This transfer can be done on a bit-by-bit basis, but typically, this data transfer process is done as a 24-bit transfer. MI20 tells the MACRO Station how many of these data transfers will take place for a given period.

Hex Digit #	1	2	3	4	5	6	7	8	9	10	11	12
Contents	From Register Format Code		From Register Address				To Register Format Code		To Register Address			

The first 24 bits (six hex digits) specify the address of the register on the Compact MACRO Station from which the data is to be copied; the second 24-bits (six hex digits) specify the address on the Compact MACRO Station to which the data is to be copied. In each set of six hex digits, the last four hex digits specify the actual address. The first two digits (8-bits) specify what portion of the address is to be used.

The following table shows the 2-digit hex format codes and the portions of the address that each one selects.

Code	X or Y	Bit Width	Bit Range	Notes
$40	Y	8	0-7	
$48	Y	8	8-15	
$50	Y	8	16-23	
$54	Y	12	0-11	Lower 12-bit ADC registers
$60	Y	12	12-23	Upper 12-bit ADC registers
$64	Y	16	0-15	
$6C	Y	16	8-23	16-bit MACRO Servo Node Registers
$78	Y	24	0-23	24-bit MACRO Servo Node Registers
$B0	X	8	0-7	
$B8	X	8	8-15	
$C0	X	8	16-23	
$C4	X	12	0-11	
$D0	X	12	12-23	
$D4	X	16	0-15	
$DC	X	16	8-23	16-bit MACRO I/O Node Registers
$E8	X	24	0-23	24-bit MACRO I/O Node Registers

Example:

```
MI21=$780200E8C0A0
```

copies 24-bit data from Station address Y:$0200 to X:$C0A0

ACC-59E Self-Configured Data Transfer Example for MACRO

This example uses the I/O transfer method. This method will transfer data from any MACRO station memory location to the I/O transfer node. This could be done on a bit-by-bit basis or as a 24-bit transfer. For ACC-59E with automatic copy implemented, the ADC locations are found at locations Y:$0200 through Y:$0207. These 24-bit registers contain the information for the ADC channels of data. The lower 12 bits contain ADC value for channels 1-8.

If ACC-59E is used in conjunction with I/O accessories ACC-3E, ACC-9E, ACC-10E, ACC-11E, ACC-12E, or ACC-13E, use the three 16-bit read/write method (48-bit per node) of I/O transfer. This will free up the 24-bit I/O registers for the eight channels of A/D. The 24-bit I/O registers could then be mapped back 12 bits at a time and four of the six 24-bit registers could be used to read eight ADCs. If more I/O registers are available, then the three 16-bit registers could be used also. If more nodes are needed, then the ADC 16-bit registers for the current loop feedback could be used to transfer the information back to the Ultralite when the axis node used is not in PWM mode.

Note

Remember, if there is only one master (Ultralite) in the system, then node 14 could be used for I/O transfer (two 24-bit registers and six 16-bit registers).

Data Acquisition Example uses two nodes with multiple reads and the address DIP switch settings are set for $8800.

```
m960->x:$c0a0,0,24          ;uses node 2 24-bit register
m961->x:$c0a4,0,24          ;uses node 3 24-bit register

m72->x:$0701,0,24,s         ;for PMAC2 Ultralite
                            ;Use Isx11 and Isx12 for Turbo Ultralite
ms0,mi975=$ccc
ms0,mi19=4
ms0,mi20=3
ms0,mi987=1
ms0,mi988=0
ms0,mi989=$8800

open plc10 clear

cmd "ms0,i21=$780200E8C0a0";ADC1
cmd "ms0,i22=$780201E8C0a4";ADC2
m72=100*(8388608/i10)       ;for PMAC2 Ultralite
while (m72>0) endwhile
;i5111=100*(8388608/i10)    ;for Turbo
;while(i5111>0) endwhile
p601=m960&$000fff
p602=m961&$000fff

cmd "ms0,i21=$780202E8C0a0";ADC3
cmd "ms0,i22=$780203E8C0a4";ADC4
m72=100*(8388608/i10)
while (m72>0) endwhile
;i5111=100*(8388608/i10)    ;for Turbo
;while(i5111>0) endwhile
p603=m960&$000fff
p604=m961&$000fff

cmd "ms0,i21=$780204E8C0a0";ADC5
cmd "ms0,i22=$780205E8C0a4";ADC6
m72=100*(8388608/i10)
while (m72>0) endwhile
;i5111=100*(8388608/i10)    ;for Turbo
;while(i5111>0) endwhile
p605=m960&$000fff
p606=m961&$000fff
```

```
cmd "ms0,i21=$780206E8C0a0";ADC7
cmd "ms0,i22=$780207E8C0a4";ADC8
m72=100*(8388608/i10)
while (m72>0) endwhile
;i5111=100*(8388608/i10)    ;for Turbo
;while(i5111>0) endwhile
p607=m960&$000fff
p608=m961&$000fff

close
```

ACC-59E Servo Feedback Use Example for MACRO

Using 4-axis servo at MACRO Station, the fourth axis closes servo loop on A/D converted value from the ACC-59E.

The information is sent from ADC1 at MACRO Station (Y:$0200,0,12) to the encoder conversion table entry number four (MS0, MI123) at the station. Conversion table entry fifth (MS0, MI124) will be the mask and the result. The information is sent to the Ultralite conversion table based on MS0, MI104. MI104 should contain the address of the fifth entry of the conversion table ($14).

```
MS0,MI987=1             ;ENABLES A/D INPUTS.
MS0,MI988=0             ;unipolar inputs
MS0,MI989=$8800         ;board address on backplane
MS0,MI123=$200200       ;Feeding ANAI00 information into the end of
MS0,MI124=$000FFF       ;macro station conversion table with parallel
MS0,MI104=$14           ;Y-word data, no filtering conversion method.
                        ;the conversion table result will be in X:$14
                        ;of the macro station
```

This information will be sent to the Ultralite's encoder conversion table (fourth encoder entry) as a parallel word. By default, the position and velocity loop address should be correct.

Analog Output Configuration with UMAC MACRO

The method used to set up general use of the analog outputs over the MACRO ring is similar to that for UMAC Turbo, but a communication method must be set up first. The most straightforward way of doing this is by using MACRO data transfer through the I/O nodes as demonstrated for the ADC inputs above. Use the self-configured data transfer because there is not an automatic transfer with the correct configuration parameters included. Therefore, configure the transfers similar to that shown previously for the ADC transfers.

To implement the self-configured data transfers across I/O nodes, first set up the variables MI20 and MI21-68. Since the ACC-59E has eight general-purpose DAC outputs, set up MI20 and MI21-MI36. The reason for using 16 data transfers is to set up both transferring of data from the I/O nodes to the MACRO address and then back from MACRO memory to the I/O node. If the transfer was not set up to return to I/O node, then what was written to the DAC could not be read. The command would still get the proper location and voltage would be present on the DAC, however, it could not be read in any of the user programs. Remember that we could have chosen any set of 16 variables within MI21-MI68.

Example: Configuring ACC-59E general DAC outputs over the MACRO ring using self-configured data transfers with base address of Y:$8800

```
ms0,mi19=4                   ;enable node transfer
ms0,mi975=$ccc               ;enable I/O channels
```

```
ms0,mi20=$ffff
ms0,mi21=$C4C0A8548808      ;DAC1 node 6 24 bit register
ms0,mi22=$548808C4C0A8      ;DAC1 node 6
ms0,mi23=$CCC0A9548809      ;DAC2 node 6 16 bit register
ms0,mi24=$548809CCC0A9      ;DAC2
ms0,mi25=$CCC0AA54880A      ;DAC3      node 6 16 bit register
ms0,mi26=$54880ACCC0AA      ;DAC3
ms0,mi27=$CCC0AB54880B      ;DAC4 node 6 16 bit register
ms0,mi28=$54880BCCC0AB      ;DAC4
ms0,mi29=$C4C0AC608808      ;DAC5 node 7 24 bit register
ms0,mi30=$608808C4C0AC      ;DAC5
ms0,mi31=$CCC0AD608809      ;DAC6 node 7 16 bit register
ms0,mi32=$608809CCC0AD      ;DAC6
ms0,mi33=$CCC0AE60880A      ;DAC7 node 7 16 bit register
ms0,mi34=$60880ACCC0AE      ;DAC7
ms0,mi35=$CCC0AF60880B      ;DAC8 node 7 16 bit register
ms0,mi36=$60880BCCC0AF      ;DAC8
```

ACC-59E Configuration Example for UMAC Turbo

Base Address: $8800

Using ADC Automatic Read Method

Macro Communication Setup with a Turbo Ultralite

```
I6840=$4030       ;sets up  master/master ASCII communications
I6841=$fcfff      ;enable nodes 0-7
I70=$0033         ;takes place of i1000 on non-turbo Ultralite
I71=$0033         ;takes place of i1002 on non-turbo Ultralite
I78=32            ;master/slave comm timeout- takes place of i1003
i79=32            ;master/master communications timeout
I80=100           ;in place of i1001 on non-turbo Ultralite
I81=2             ;in place of i1004 on non-turbo Ultralite
I82=2             ;takes place of i1005 on non-turbo Ultralite
```

Automatic I/O Transfer Method

```
ms0,mi19=4                  ;enable node transfer
ms0,mi975=$ccc              ;enable I/O channels
ms0,mi987=1                 ;enable automatic copy of ADC registers
ms0,mi988=$ff               ;sets either bipolar or unipolar
ms0,mi989=$8800             ;sets the copying from specified incoming
                            ;ADC address
ms0,mi173 = $20C0A1000200   ;sets up the nodes 2,3 transfer to the
                            ;Ultralite
ms0,mi175 = $20C0A0000206   ;sets up the nodes 10,11 transfer to
                            ;the Ultralite

ms0,mi20=$ffff
ms0,mi21=$C4C0A8548808      ;DAC1 node 6 24 bit register
ms0,mi22=$548808C4C0A8      ;DAC1 node 6
ms0,mi23=$CCC0A9548809      ;DAC2 node 6 16 bit register
ms0,mi24=$548809CCC0A9      ;DAC2
ms0,mi25=$CCC0AA54880A      ;DAC3      node 6 16 bit register
ms0,mi26=$54880ACCC0AA      ;DAC3
ms0,mi27=$CCC0AB54880B      ;DAC4 node 6 16 bit register
ms0,mi28=$54880BCCC0AB      ;DAC4
ms0,mi29=$C4C0AC608808      ;DAC5 node 7 24 bit register
ms0,mi30=$608808C4C0AC      ;DAC5
```

```
ms0,mi31=$CCC0AD608809      ;DAC6 node 7 16 bit register
ms0,mi32=$608809CCC0AD      ;DAC6
ms0,mi33=$CCC0AE60880A      ;DAC7 node 7 16 bit register
ms0,mi34=$60880ACCC0AE      ;DAC7
ms0,mi35=$CCC0AF60880B      ;DAC8 node 7 16 bit register
ms0,mi36=$60880BCCC0AF      ;DAC8
```

M-Variables for the Ultralite

```
m1->x:$78421,8,12,s         ;channel 2 node transfer -ADC1
m2->x:$78422,8,12,s         ;channel 2 node transfer -ADC2
m3->x:$78423,8,12,s         ;channel 2 node transfer -ADC3
m4->x:$78425,8,12,s         ;channel 3 node transfer -ADC4
m5->x:$78426,8,12,s         ;channel 3 node transfer -ADC5
m6->x:$78427,8,12,s         ;channel 3 node transfer -ADC6
m7->x:$78420,0,12,s         ;channel 2 node transfer -ADC7
m8->x:$78424,0,12,s         ;channel 3 node transfer -ADC8

m11->x:$78428,0,12          ;channel 6 node transfer -DAC1
m12->x:$78429,8,12          ;channel 6 node transfer -DAC2
m13->x:$7842a,8,12          ;channel 6 node transfer -DAC3
m14->x:$7842b,8,12          ;channel 7 node transfer -DAC4
m15->x:$7842c,0,12          ;channel 7 node transfer -DAC5
m16->x:$7842d,8,12          ;channel 7 node transfer -DAC6
m17->x:$7842e,8,12          ;channel 2 node transfer -DAC7
m18->x:$7842f,8,12          ;channel 3 node transfer -DAC8
```

ACC-59 PINOUTS

TB1 (4-Pin Terminal Block)

Pin #	Symbol	Function	Description
1	GND	Common	Digital Ground
2	+5V	Input	External Supply
3	+15V	Input	External Supply
4	-15V	Input	External Supply

DB15 Breakout Option

J1 – Top of Card– ADC1 through ADC4 (DB15 Connector)

Pin #	Symbol	Function	Description
1	+ADC1	Input	+Analog Input #1
2	+ADC2	Input	+Analog Input #2
3	+ADC3	Input	+Analog Input #3
4	+ADC4	Input	+Analog Input #4
5	Open	N/A	
6	Open	N/A	
7	AGND	Common	Ground*
8	-15V	Output	Neg. Supply**
9	-ADC1	Input	-Analog Input #1
10	-ADC2	Input	-Analog Input #2
11	-ADC3	Input	-Analog Input #3
12	-ADC4	Input	-Analog Input #4
13	Open	N/A	
14	Open	N/A	
15	+15V	Output	Pos. Supply**
* This common point is connected to the digital ground of the UMAC board. **The supply voltages are for output from the board to supply the sensors connected to ACC-59E. The drawn current should not exceed 0.5 A.			

J2 - Top of Card– ADC5 through ADC8 (DB15 Connector)

Pin #	Symbol	Function	Description
1	+ADC5	Input	+Analog Input #1
2	+ADC6	Input	+Analog Input #2
3	+ADC7	Input	+Analog Input #3
4	+ADC8	Input	+Analog Input #4
5	Open	N/A	
6	Open	N/A	
7	AGND	Common	Ground*
8	-15V	Output	Neg. Supply**
9	-ADC5	Input	-Analog Input #1
10	-ADC6	Input	-Analog Input #2
11	-ADC7	Input	-Analog Input #3
12	-ADC8	Input	-Analog Input #4
13	Open	N/A	
14	Open	N/A	
15	+15V	Output	Pos. Supply**

* This common point is connected to the digital ground of the UMAC board.

**The supply voltages are for output from the board to supply the sensors connected to ACC-59E. The drawn current should not exceed 0.5 A.

J1 – Bottom of Card – DAC 1 through DAC 4 (DB15 Connector)

Pin #	Symbol	Function	Description
1	+DAC1	Output	+DAC Output #1
2	+DAC2	Output	+DAC Output #2
3	+DAC3	Output	+DAC Output #3
4	+DAC4	Output	+DAC Output #4
5	Open	N/A	
6	Open	N/A	
7	AGND	Common	Ground*
8	Open	N/A	
9	-DAC1	Output	-DAC Output #1
10	-DAC2	Output	-DAC Output #2
11	-DAC3	Output	-DAC Output #3
12	-DAC4	Output	-DAC Output #4
13	Open	N/A	
14	Open	N/A	
15	+5V	Output	Digital Supply

* This common point is connected to the digital ground of the UMAC board.

J2 – Bottom of Card– DAC 5 through DAC 8 (DB15 Connector)

Pin #	Symbol	Function	Description
1	+DAC5	Output	+DAC Output #5
2	+DAC6	Output	+DAC Output #6
3	+DAC7	Output	+DAC Output #7
4	+DAC8	Output	+DAC Output #8
5	Open	N/A	
6	Open	N/A	
7	AGND	Common	Ground*
8	Open	N/A	
9	-DAC5	Output	-DAC Output #5
10	-DAC6	Output	-DAC Output #6
11	-DAC7	Output	-DAC Output #7
12	-DAC8	Output	-DAC Output #8
13	Open	N/A	
14	Open	N/A	
15	+5V	Output	Digital Supply
* This common point is connected to the digital ground of the UMAC board.			

Terminal Block Option (Top)

Connector TB1 Top – ADC1 through ADC4

Pin #	Symbol	Function	Description	Notes
1	ADC1+	Input	Analog Input #1	
2	ADC1-	Input	Analog Input #1/	
3	ADC2+	Input	Analog Input #2	
4	ADC2-	Input	Analog Input #2/	
5	ADC3+	Input	Analog Input #3	
6	ADC3-	Input	Analog Input #3/	
7	ADC4+	Input	Analog Input #4	
8	ADC4-	Input	Analog Input #4/	
9	NC	NC		
10	NC	NC		
11	NC	NC		
12	NC	NC		

Connector TB2 Top – ADC5 through ADC8

Pin #	Symbol	Function	Description	Notes
1	ADC5+	Input	Analog Input #5	
2	ADC5-	Input	Analog Input #5/	
3	ADC6+	Input	Analog Input #6	
4	ADC6-	Input	Analog Input #6/	
5	ADC7+	Input	Analog Input #7	
6	ADC7-	Input	Analog Input #7/	
7	ADC8+	Input	Analog Input #8	
8	ADC8-	Input	Analog Input #8/	
9	NC	NC		
10	NC	NC		
11	NC	NC		
12	NC	NC		

Connector TB3 Top – Power Supply Outputs

Pin #	Symbol	Function	Description	Notes
1	AGND	Input/Output	Common Reference for ADC1-16	
2	+15V	Output	+15V from UMAC Power Supply	Fused (1/2 A)
3	-15V	Output	-15V from UMAC Power Supply	Fused (1/2 A)

Terminal Block Option (Bottom)

Connector TB1 Bottom – DAC1 through DAC4

Pin #	Symbol	Function	Description	Notes
1	DAC1+	Output	DAC Output #1+	
2	DAC1-	Output	DAC Output #1-	
3	DAC2+	Output	DAC Output #2+	
4	DAC2-	Output	DAC Output #2-	
5	DAC3+	Output	DAC Output #3+	
6	DAC3-	Output	DAC Output #3-	
7	DAC4+	Output	DAC Output #4+	
8	DAC4-	Output	DAC Output #4-	
9	NC	NC		
10	NC	NC		
11	NC	NC		
12	NC	NC		

Connector TB2 Bottom – ADC13 through ADC16

Pin #	Symbol	Function	Description	Notes
1	DAC5+	Output	DAC Output #5+	
2	DAC5-	Output	DAC Output #5-	
3	DAC6+	Output	DAC Output #6+	
4	DAC6-	Output	DAC Output #6-	
5	DAC7+	Output	DAC Output #7+	
6	DAC7-	Output	DAC Output #7-	
7	DAC8+	Output	DAC Output #8+	
8	DAC8-	Output	DAC Output #8-	
9	NC	NC		
10	NC	NC		
11	NC	NC		
12	NC	NC		

Connector TB3 Top – Power Supply Outputs

Pin #	Symbol	Function	Description	Notes
1	GND	Input/Output	Common for +5V Output	
2	+5V	Output	Digital Output	
3	NC			