OPERATION AND USE OF THE ANIMATICS AIO-100
ANALOG INPUT OUTPUT MODULE

Table of Contents

1. Introduction

1.1 Product Description

The AIO-100 is a four-input, one-output analog I/O peripheral module for the AniLink bus. This module allows an Animatics controller to read analog signals from external devices, or control an external system with an analog signal. Speed control drives, joysticks, and process controls are common examples. The range of signal voltage for both input and output is 0 to +5 VDC, with eight bits of resolution.

Upper and lower limits of the A/D or D/A functions are independently adjustable, allowing the full eight bit accuracy to be applied to small signal ranges. Each input channel is can be centered independently. These adjustments help the AIO-100 card adapt to almost any application requirement.

The AIO-100 is especially useful on Series 5000 systems for creating multi-axis joystick applications. Such applications make use of the AniLink bus and the Series 5000's joystick firmware command set. Either digital or proportional joysticks can be easily integrated. The AIO-100 has digital inputs wired in parallel with the Series 5000's auxiliary inputs 5 and 6. These inputs are dedicated to speed select an joystick engage/ disengage when the controller is operating in JOY1ON mode.

The AIO-100 is cable/plug compatible with either the Series 5000 or the SmartMotor control systems. Up to four AIO-100 modules can be addressed by a single Series 5000 controller, up to eight by a SmartMotor. Powered by the +5 V and ground from the AniLink cable, these units are equipped with a voltage regulator allowing support from an external +6 V to +24 V source. Most applications do not need external support.

The AniLink Network is a proprietary serial based, high speed component network shared by the Series 5000 and SmartMotor lines of motion control products.

1.2 Features

- One eight-bit analog output, ranging from 0 to 5 VDC.
- Four eight-bit analog inputs, ranging from 0 to 5 VDC.
- Parallel inputs for Series 5000 Auxiliary Inputs 5 and 6.
- Simple plug in operation
- Convenient size and mounting, DSUB connector
- Direct firmware support under the Series 5000 and SmartMotor command sets; special multi-axis joystick support in the Series 5000 language.
- +5 V DC operation
    - Can be drawn directly from AniLink network cable
    - Alternate power supply port available
- AniLink Network Addressable (3-bit)
    - High speed serial communications (100K BPS)
    - Multi-drop addressing,

1.3 Part Numbering

AIO-100 refers specifically to the one-in, four-out, eight bit analog product. Produced as a general purpose peripheral, other similar peripherals have been produced within the AIO-1XX family of peripherals.

Please contact your applications engineer for specifics about our special products.

4 Theory of Operation

The AIO-100 card is equipped with a voltage comparitor, a D/A converter and number of track and hold buffers. The D/A converter can be thought of a resistor divider chain connected to the external reference voltage and ground. There are 256 taps in the chain, each with a selection switch and supporting logic.

The AIO-100 communicates serially over the AniLink bus with a Animatics controller. The controller can write to the AIO-100 module (D/A), or read from it (A/D).

Writing to the AIO-100 is carried out as a three byte string. The first byte of a write command contains a three bit addressing scheme. The last four available bit patterns of this scheme are reserved for functions internal to the Series 5000 control. This is the reason that a maximum of four AIO-100 cards may be uniquely addressed on a Series 5000 based AniLink network. Since the SmartMotor does not have this limitation, all eight bit address patterns are available, allowing up to eight uniquely addressed cards.

The second byte of a write enables the analog output and performs housekeeping functions, the third byte contains the output value. When a write is performed, the output level is matched by a auto-zeroing unity gain amplifier. This buffering amplifier allows the voltage divider to be used for other functions without disturbing the AIO-100's output level.

A read is carried out a string of two bytes, and it may cause the AIO-100 to return up to five bytes since the channels must be converted in order. When a conversion of a channel is started, a an input voltage sample is stored on the chip, and converted to an eight bit binary code. The conversion rate is determined by the actual speed of the AniLink bus.

5 Simplified Schematic
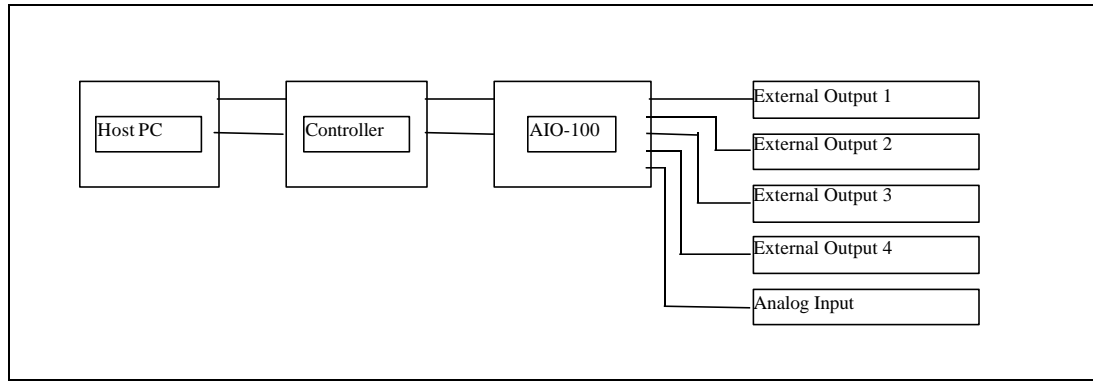
System Block Diagram

2.   Specifications

All listed specifications are correct as of the date of printing. See errata for latest details. Any and all product specifications are subject to change without notice by the manufacturer.
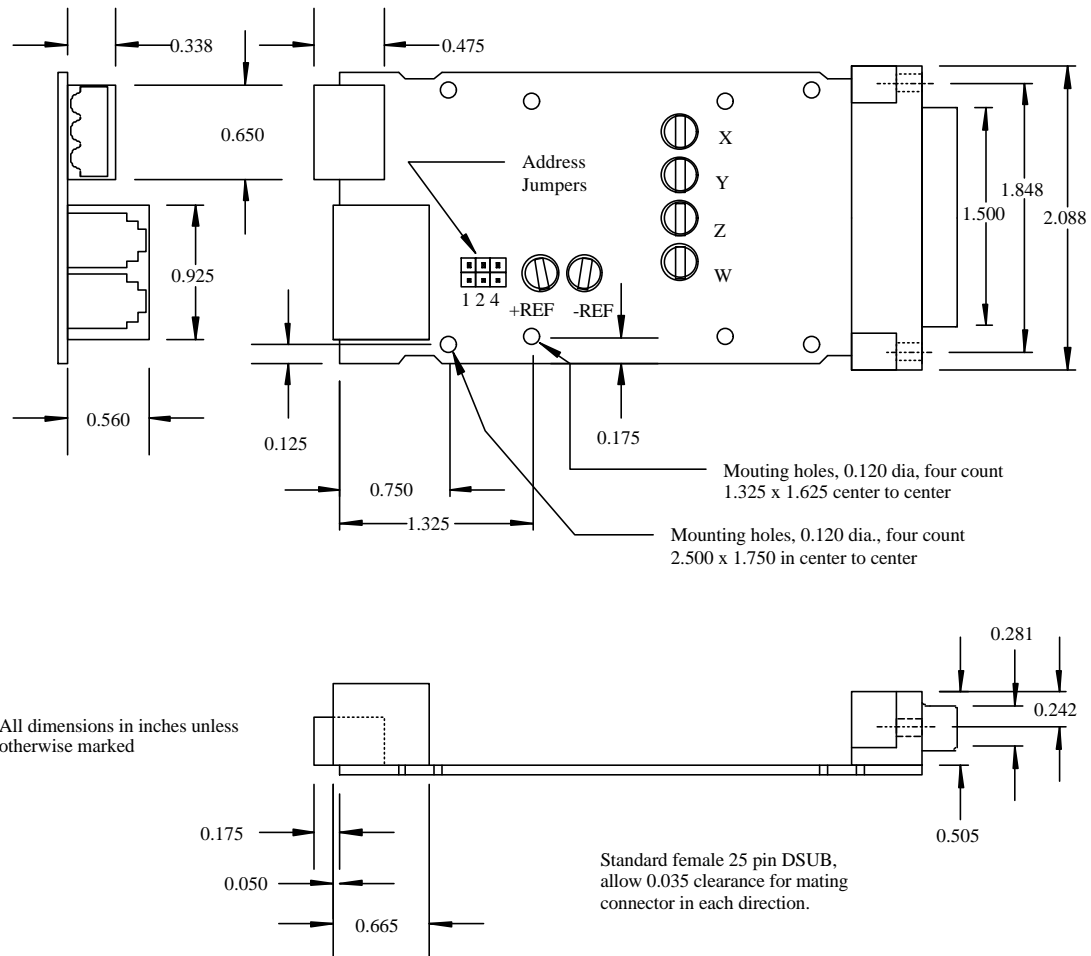
2.1   Electrical

Bus DC line voltage	5V DC
Aux. Power Connector voltage	+6 to 24 V DC
Normal Maximum DC current	100 ma

2.2 Mechanical

Dimensions: See Figure
Weight: 1 oz

Address Jumpers

1 2 4

+REF   -REF

X
Y
Z
W

0.338

0.475

0.650

0.925

0.560

0.125

0.750

1.325

0.175

1.500

1.848

2.088

Mouting holes, 0.120 dia, four count
1.325 x 1.625 center to center

Mounting holes, 0.120 dia., four count
2.500 x 1.750 in center to center

All dimensions in inches unless
otherwise marked

0.175

0.050

0.665

0.281

0.242

0.505

Standard female 25 pin DSUB,
allow 0.035 clearance for mating
connector in each direction.

2.3 Environmental

Operating temperature	0°C to 50°C
Storage temperature	-20°C to 70°C
Humidity	0 % to 90 % (non-condensing)

<u>**3. Installation**</u>

3.1 Unpacking and Inspection

Upon receipt of the equipment, carefully inspect to ensure that no damage has occurred during shipment. If damage is detected, notify the carrier immediately. Equipment should be stored in its original shipping container until ready for use.

3.2 Mounting

The AIO-100 module should be mounted inside a cabinet or suitable enclosure to protect it from physical and environmental damage. It must be kept free of combustible or flammable materials, oil vapor, steam, excessive moisture, corrosives and general debris.

Mounting holes for standard 4-40 screws are located in eight places on the board. The board can also be secured using the two threaded 4-40 nuts in the 25-pin DSUB connector. Jack screws are suggested for this purpose.
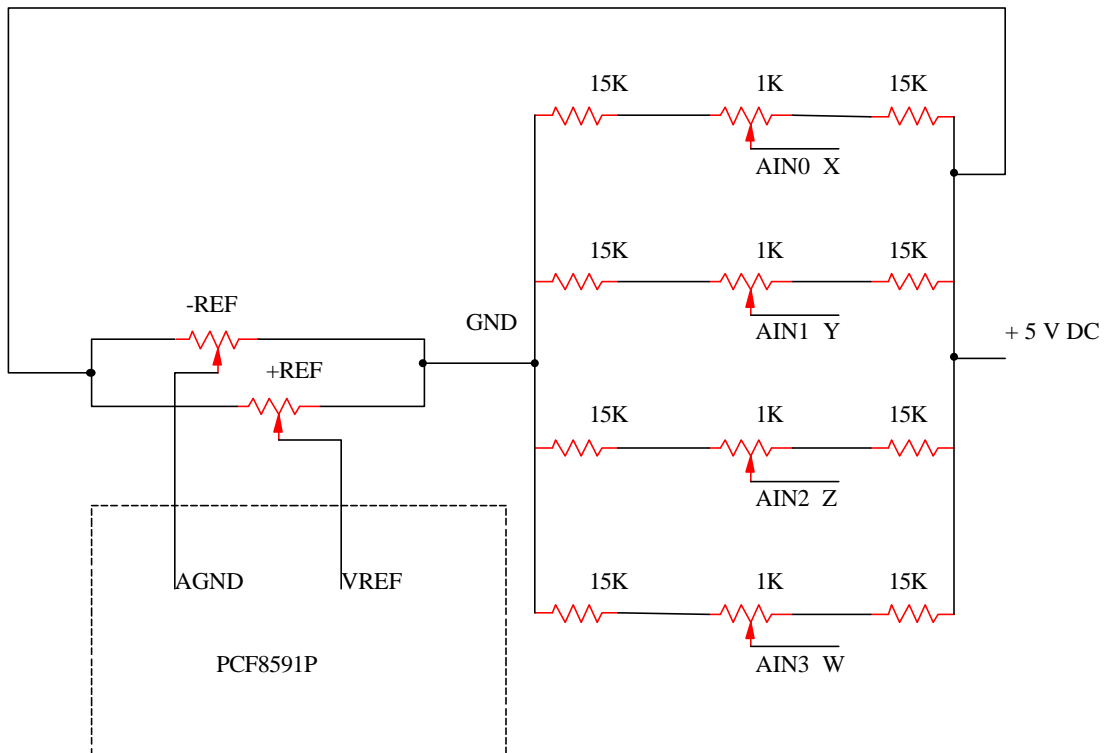
3.3 Wiring

Wiring the AIO-100 is often as simple as plugging in the supplied "phone cable" into the RJ11-6 type connector, and plugging the opposite end of the cable into the controller. By using the second RJ11-6 (wired in parallel), additional AniLink modules can be plugged in to the network.

Longer runs of AniLink cable are possible. Maximum tested runs for the "phone cable" wiring and RJ11-6 type connectors is about 3 feet. Use of higher efficiency shielded cable and better connectors will allow much longer runs.

Users desiring industrial-type communications connections often remove the RJ11 jack and solder shielded cable directly to the p.c. board. While this is a generally accepted practice, understand that poor workmanship will void any warranty on this product.

In the following schematic, normally open push buttons can be connected to the module to create an inexpensive jog function. Notice the +5 and GND taps are taken from the outputs in series with 1K resistors. This prevents a short if both the positive and negative buttons are pushed simultaneously.

15K 1K 15K

AIN0 X

-REF

15K 1K 15K

+REF AIN1 Y

GND

AGND VREF + 5 V DC

15K 1K 15K

AIN2 Z

PCF8591P

15K 1K 15K

AIN3 W

+ 5 V DC

1   Connector Pin Out

The I/O connector is a standard female 25 pin DSUB. Peripheral systems can be powered from the AIO-100's + 5 VDC and GND up to the maximum available current on the AniLink network.

13   1

25   14
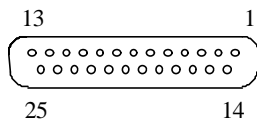
PIN No. Function
1   Analog input #1
2   Analog input #2
3   Analog input #3
4   Analog input #4
5   No Connection
6   Parallel to 5000's Aux. Input 5
7   Parallel to 5000's Aux. Input 6
8   No Connection
9   No Connection
10   Analog Output
11   +5 DC
12   +5 DC
13   + 5 DC with 1K resistor
14 - 24 GND
25   GND with 1 K resistor

1   6   1   6
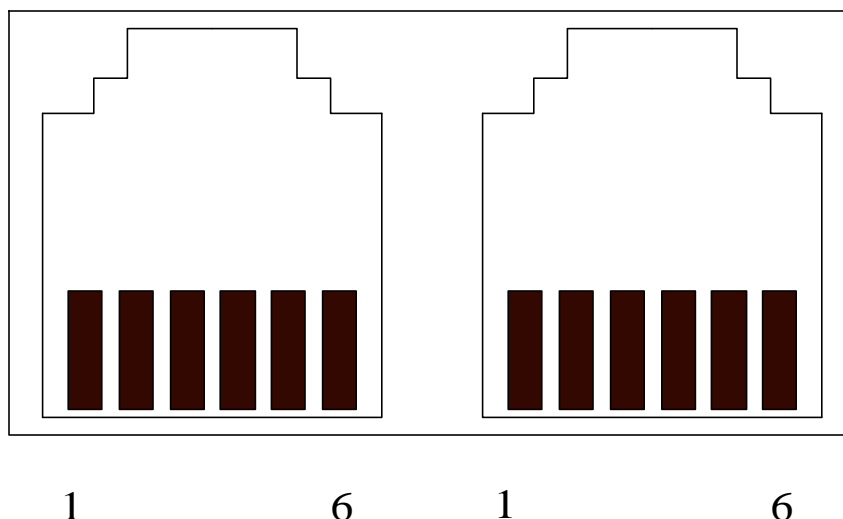
Connector C1 (RJ11-6 connector, x 2 in parallel)

PIN	SIGNAL	DESCRIPTION
1	IN #5	Reserved for other AniLink peripheral devices
2	GND	To Controller
3	+5 V DC	AniLink Power from controller (limited to about 150 Ma)
4	CLOCK	AniLink Clock
5	DATA	AniLink Data
6	IN #6	Reserved for other AniLink peripheral devices

The AniLink connector is a standard RJ-11-6. The two female RJ-11-6 sockets on the AIO-100 board are wired in parallel. This allows the AniLink network to be extended by plugging one module into the next.

When several AniLink devices are connected to the same network, the available controller power supply may not to maintain operating voltage to the peripherals. In this case, an additional power source can be added to the network at the screw terminals provided at connector C2. If additional power is fed into on module of an AniLink network that power will be distributed to the other modules over the +5V line of the AniLink network. Attention should be paid to the 7805 voltage regulator for any unit receiving external power: if the regulator gets too hot to touch, connect the additional power to the individual screw terminal ports of all the AniLink

modules.
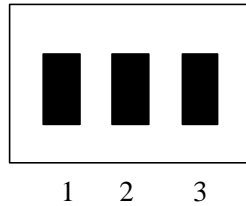
Connector C2, External Power Connector

PIN	SIGNAL	Description
1	+5-+24 VDC	
2	GND	
3	+5-+24 VDC	Internally connected to 1

The power connector is a three socket phoenix type connector.

3.4   Power-up and Checkout

No particular power up procedure is necessary for the AIO-100.

A checkout procedure can be derived from the programming examples found later in this manual. Use a 5 K pot and a DMM or oscilloscope to check the expected ranges and returns if desired.

Be sure to apply common safety practices when working on any motion based system: make sure that their is no possibility of personal injury or machine damage before first time power up.

4. Adjustments

1   Adjusting +REF, -REF and the Input Centers
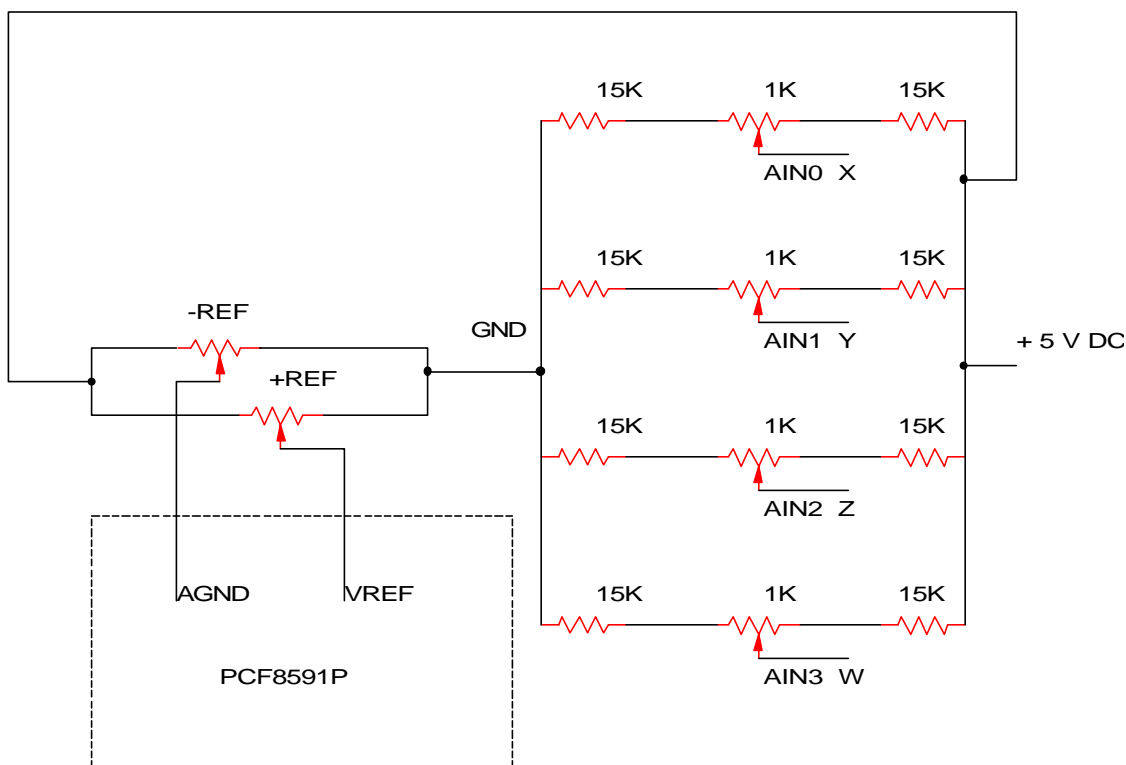
The input and output signal ranges are tied to the PCF8591's AGND and VREF signal levels. The +REF and -REF potentiometers on the AIO-100 set the signal maximum and minimum.

The AIN# potentiometers are independent channel adjustments for the inputs. These are used to center the input range.

When building a proportional joystick, start by adjusting the +REF and -REF to allow the full 0 - 5V swing. Next adjust the joystick center point and the X, Y, Z, and W channel potentiometers such that the neutral stick position reads 128 +/- 2 and the full 0 to 255 swing is available to the stick.

4.2   AniLink Bus Module Address

Peripherals on a AniLink network must have non-conflicting addresses to function properly. This normally means a unique address for each AniLink peripherals. The AniLink peripherals use a three bit address scheme based on the of jumpers 1, 2 and 4.

Module Address	Jumper State			X = jumper in place O = jumper absent
	1	2	3	
A	**O**	**O**	**O**	**Bold values only valid on Series 50000**
B	**X**	**O**	**O**	
C	**O**	**X**	**O**	
D	**X**	**X**	**O**	
E	O	O	X	
F	X	O	X	
G	O	X	X	
H	X	X	X	

An AIO-100 module must use address A to be addressed by the JOY1ON firmware support.

5. Maintenance and Repair

5.1   Maintenance

There are no user serviceable components on the AIO-100. The only periodic maintenance requirement is to keep the board clean.

5.2   Indications

A failure of the AIO-100 module would be indicated by observably incorrect command returns at the host controller, or by invalid signals appearing at the output. These conditions would present themselves as a loss of control in an application.

5.3   Trouble Shooting

As these units have no user serviceable parts, trouble shooting is usually limited to checking for power and ground, and checking for communications signal.

As a part of documenting your application, you should record acceptable test levels for future use during the development process. Should questions arise later about the serviceable condition of an AIO-100 card, comparison levels can be a tremendous asset.

The commands and programming techniques found in the programming section of this document will be useful in the troubleshooting process.

If your unit is not working:

      0.   Check cabling for unplugged connectors or cable cuts

      1.   Check for power using a DMM
                A.   On connector C1: +5 between pins 2 and 3
                B.   On IC 7805: +5 between pins 1 and 2
                C.   External power source (if applicable)

           If not receiving power, locate cause.

      2.   Check for signal on AniLink Clock and Data lines using a logic probe or oscilloscope. Logic and Data lines are normally high between data transmissions.

      3.   If power, data and clock all show correct signals, and your unit is still not working, return your module to Animatics for inspection and repair.

6. Programming

The languages and programming techniques for the Series 5000 and the SmartMotor are radically different. Consistent between these languages is the direct command support and module addressing scheme.   Each AIO-100 card on an AniLink network can be addressed by a letter address, A-D or A-H, and each module has four channels, 1 through 4.

Additional information about the command sets and languages demonstrated in this section can be found in the Series 5000 and SmartMotor User's Manuals.

6.1   Series 5000 Command Set

The AniLink bus on a Series 5000 controller runs throughout the controller, and is ported to the outside in two locations. Addresses E, F, G, and H are used for processes within the controller and are therefore not available to the outside network. See the Series 5000 User's manual for details on this feature.

The Series 5000 command set provides two levels of communication to the AIO-100: firmware level joystick support, and direct communications commands. To be addressed by the firmware level joystick commands, the target AIO-100 must use address A.

Command	Comments

AIN(let)	Upload the value of the corresponding analog input
	    (let) valid from A through D – module address
	    # valid from 1 to 4 – channel number

AOUT(let)#	Set the level of the analog output for the corresponding module
	    (let) valid from A through D – module address
	    # valid from 0 though 255 – output level

JOYVH#	Set the high speed max. deflection velocity for the JOY1ON mode, #'s units determined by the system's SPU values
JOYVL#	Set the low speed max. deflection velocity for the JOY1ON mode #'s units determined by the system's SPU values
JOY1OFF	Disable the controller joystick mode
JOY1ON	Enable the controller joystick mode
JOY1?	Returns the current channel levels of the AIO-100 module A in XYZW format

Joystick Programming

   The first program that most designers and technicians want is a test loop. This program uses GOTO statements and increments variables to set up a test loop displaying modes. Variations of the program are often used in the integration of AIO-100 applications. Run the program from TERM5000 in UNTERM mode.

```
VART500
VARA0
100
JOY1?
VARA?
AOUT[VARA]
WAIT[VART]
IF([VARA]=0)
GOTO(105)
IF([VARA]=128)
GOTO110
IF([VARA]=225)
GOTO(115)
200
GOTO(100)
END
105
VARA128
GOTO(200)
110
VARA225
GOTO(200)
115
VARA0
GOTO(200)
END
```

This program demonstrates a joystick subroutine. Auxiliary input #1 is used as a joystick enable for this application. Note that JOY1ON mode uses Auxiliary inputs #5 and #6 as speed select and joystick enable respectively. Auxiliary output #6 is used as an enable confirmation.

```
100
ACC40000
X200Y300F500
WAIT
ACC10000
X0Y0F200
WAIT
IF(1ON)
GOSUB105
GOTO100
END
105
JOY1ON
106
IF(1ON)
GOTO106
JOY1OFF
RETURN
END
```

Direct Command Programming

The 100 subroutine in the first joystick program relies on the JOY1? Command to format the return from the AIO-100 module addressed as A. We could address other modules by slightly changing the program.

```
…                          Replaces from above
AINB1?                     JOY1?
AINB2?                     "
AINB3                      "
AINB4                      "
VARA?
AOUTB[VARA]                AOUTA[VARA]
…
```

The Series 5000 language can evaluate the complicated expressions used in some feedback control algorithms.

```
VARA0                      Initialization for first iteration
VARN0                      Initialization for first iteration
100                        Label statement
VARN[VARN+1]
VARB[AINA1/255]
VARB([VARB]+[AINA2/255])
VARB([VARB]+[AINA3/255])
VARB([VARB]+[AINA4/255])
VARB[VARB/4]
VARA(([VARA]*([VARN-1]))+[VARF]/[VARN])
AOUTA[VARAl*255]           Output the moving average
…
…
GOTO(100)
```

6.2   SmartMotor Command Set

The SmartMotor has direct command software support for the AIO-100 module. There are two major differences in firmware support between the SmartMotor and the Series 5000:

> The SmartMotor uses a different self-diagnostic design than the Series 5000. This different makes it possible to address up to eight unique AIO-100 modules on a single SmartMotor AniLink network.

> The SmartMotor does not have firmware level joystick support.

An example of a stored user program joystick program can be found in the SmartMotor User's Manual (JOY.SRC on page 234).

Command	Comments
	Port is valid from A through H
	Input is valid from 1 to 4
	Exp. expressions must evaluate to an integer
AIN{port}input	Fetch analog byte from port, input
AOUT{port}, exp.	Output an analog byte on channel port
RAIN{port}input	Fetch analog byte from port, input: report ASCII integer value on RS-232

This program demonstrates the use of these commands.

```
MV              'Set mode velocity
A=500           'Set acceleration to 500 =  63.13 rev. / sec.²
V=0             'Set velocity = to 0
O0              'Set current position equal to the origin
UAI
UBO
RAINA1          'Report channel 1 value
RAINA2          'Report channel 2 value
RAINA3          'Report channel 3 value
RAINA4          'Report channel 4 value
WHILE 1
a=AINA1
V=a
b=AINA2
c=AINA3
d=AINA4
G
IF UAI==1 GOSUB 0 ENDIF
LOOP
END
C0                              'Subroutine 0 reports on RS-232 in format
UB=1                            'User output B signals you to get your finger off of A
WHILE UAI==0 LOOP
UB=0
PRINT("INPUT A1 = ",a,#13)      'Notice the formatting of this statement
PRINT("INPUT A2 = ",b,#13)
PRINT("INPUT A3 = ",c,#13)
PRINT("INPUT A4 = ",d,#13)
RETURN
END
```