



Artisan Technology Group is your source for quality new and certified-used/pre-owned equipment

- FAST SHIPPING AND DELIVERY
- TENS OF THOUSANDS OF IN-STOCK ITEMS
- EQUIPMENT DEMOS
- HUNDREDS OF MANUFACTURERS SUPPORTED
- LEASING/MONTHLY RENTALS
- ITAR CERTIFIED SECURE ASSET SOLUTIONS

SERVICE CENTER REPAIRS

Experienced engineers and technicians on staff at our full-service, in-house repair center

*InstraView*SM REMOTE INSPECTION

Remotely inspect equipment before purchasing with our interactive website at www.instraview.com ↗

WE BUY USED EQUIPMENT

Sell your excess, underutilized, and idle used equipment. We also offer credit for buy-backs and trade-ins. www.artisanng.com/WeBuyEquipment ↗

LOOKING FOR MORE INFORMATION?

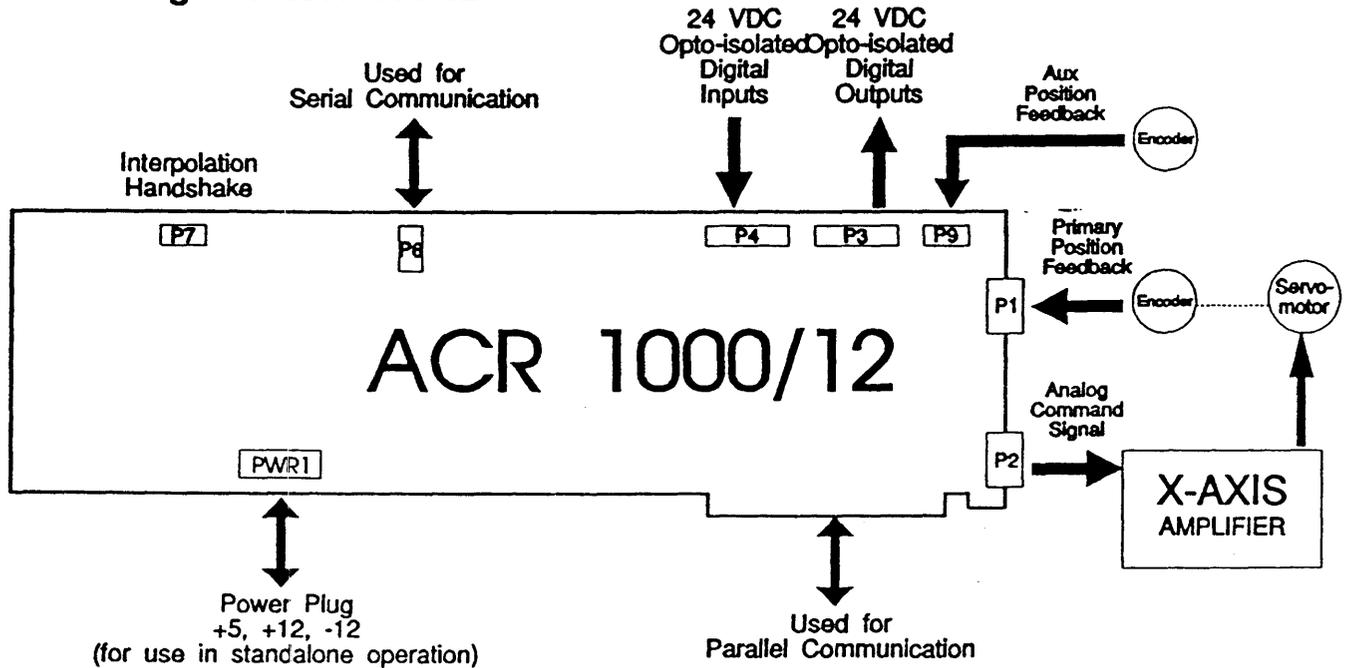
Visit us on the web at www.artisanng.com ↗ for more information on price quotations, drivers, technical specifications, manuals, and documentation

Contact us: (888) 88-SOURCE | sales@artisanng.com | www.artisanng.com

Chapter 3 System Interface

This chapter describes the location and pinouts for the various interface connectors on the ACR1000.

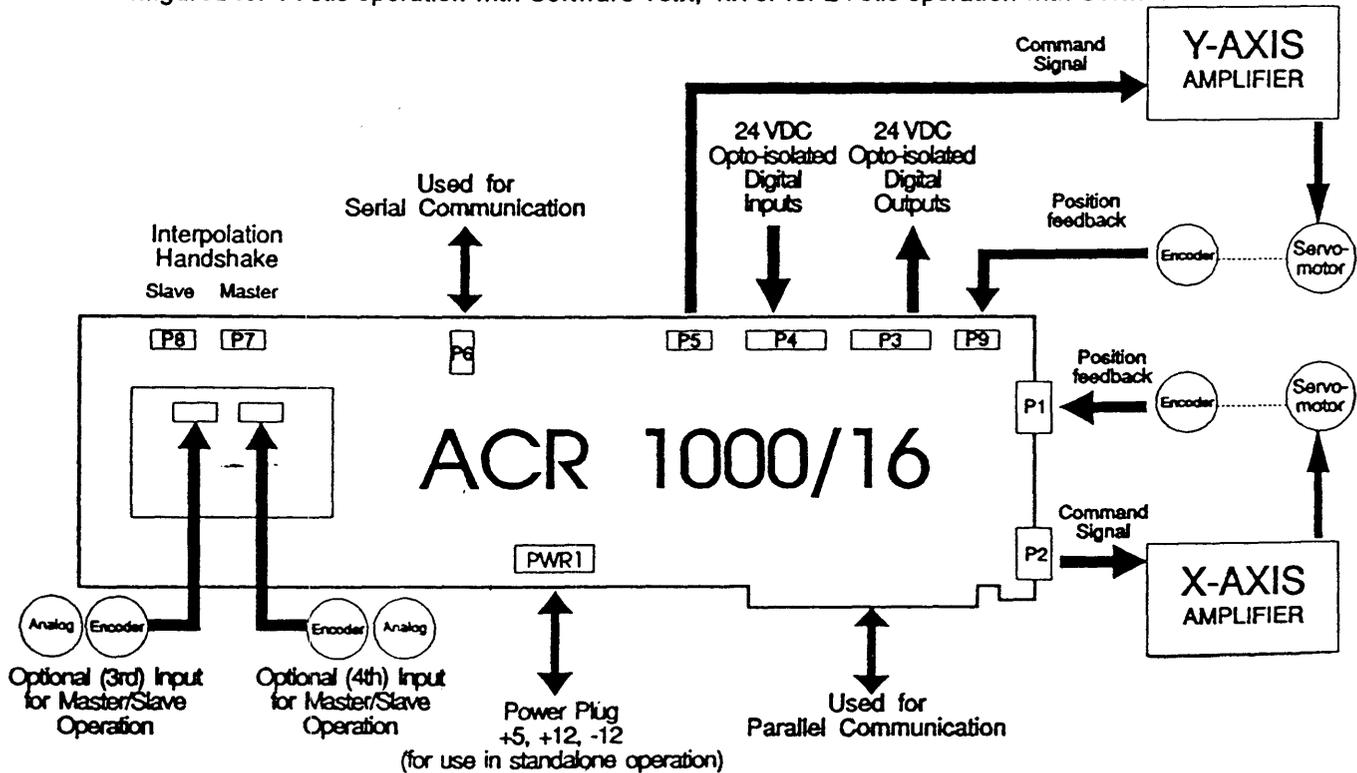
Block Diagram ACR1000/12



ACR1000/12 PLUG INTERFACE

Block Diagram ACR1000/16

The following diagram shows the plug interface for the ACR1000/16 controller. This board can be configured for 1-Axis operation with Software v3.x, 4.x or for 2-Axis operation with Software v5.x



ACR1000/16 PLUG INTERFACE

Plug Definitions

There are several plugs and option jumpers located on the board that have to be considered when installing the board in a system.

They are as follows:

P1	Encoder Signals
P2	Drive Analog/Watchdog, User Relays
P3	12 User Outputs
P4	12 User Inputs
P5	Aux DAC Output. (ACR1000/16 Only)
P6	RS232
P7	Master Axis for Interpolation Handshake
P8	Slave Axis of Interpolation. (ACR1000/16 Only)
P9	Aux Encoder, Handwheel Input
PWR	+5v, GND, +12, -12 (Not used with PC BUSS (ISA bus) interface)
JS1-1	Serial/Parallel Port Selection Jumper
JS1:2-4	BAUD Rate Selection Jumpers
JS1:5-8	Card Address Selection Jumpers
JS2:1-8	PC BUSS (ISA bus) Address (ACR1000/12 Only)

P1. Encoder Signals.

1. Channel A (chA)
2. Channel A not (chA not)
3. Channel B (chB)
4. Channel B not (chB not)
5. Marker (M)
6. Marker not (M not)
7. +5v for encoder.
8. Ground for encoder.
9. Key(n/c)

P2. Analog Drive Signal

1. Drive ± 10 volt analog
2. Drive analog ground.
3. USER11 (RELAY# 117) N.C.
4. USER11 (RELAY# 117) COMM.
5. Watchdog relay N.O.
6. Watchdog relay COMM.
7. Key(n/c)
8. USER11 (RELAY# 117) N.O.
9. Watchdog N.C.

P3: User-Defined Outputs

2. output#12
 4. output#13
 6. output#14
 8. output#15
 10. output#16
 12. output#17
 14. output#18
 16. output#19
 18. output#20
 20. output#21
 22. output#22
 24. output#23
 - 25,26. EXTERNAL ISOLATED VOLTAGE (+24VOLTS).
- 1,3,5,7,9,11,13,15,17,19,21,23. [ISOLATED GROUND.]

P4. User-Defined Inputs

2. INPUT#0
4. INPUT#1
6. INPUT#2
8. INPUT#3
10. INPUT#4
12. INPUT#5
14. INPUT#6
16. INPUT#7
18. INPUT#8
20. INPUT#9
22. INPUT#10
24. INPUT#11
- 25,26 External Isolated Voltage [+24 VOLTS].
- 1,3,5,7,9,11,13,15,17,19,21,23 [ISOLATED GROUND.]

P5. ACR1000/16 AUX DAC Output

1. +/-10Volt Signal.
2. N.C.
3. Analog GND.
- 4-10. N.C.

NOTE: This plug can be used in conjunction with a P5D Cable assembly. In this case, the pin outs are
P5D-1. +/- 10V Signal.
P5D-2. Signal GND.
P5D-3-9. N.C.

P6. RS-232 Port Plug

1. Receive Data In (Data To ACR1000)
2. Transmit Data Out (Data From ACR1000)
3. Ground
4. Transmit Data In -For Multiaxis Applications. [Transmit Data From Next Higher Address Board Should Be Wired To This Pin.]

P7. Interpolation Handshake Plug

Use Supplied INTERPOLATION CABLE assembly.

ACR1000/12:

- PLUG CABLE assembly Swap Pins x and x and x x.
- Goes from P7 of Master Axis to P7 of all Slave Axes.

ACR1000/16:

- PLUG CABLE ASSY is a 1 to 1 Cable.
- Goes from P7 of Master Axis to P8 of all Slave Axes.

P8. Interpolation Handshake For Slave Axes

ACR1000/16 only

Use Supplied INTERPOLATION CABLE ASSY.

P9. Aux Encoder Handwheel Input Plug. (PC-ACRIBM-03 and later)

1. chA
2. MARKER Complement
3. chA Complement
4. +5V
5. chB
6. GND
7. chB Complement
8. N.C
9. MARKER
10. N.C

NOTE: P9 is designed to work in conjunction with a 10 pin ribbon cable terminated in a standard 9 pin female 'D' plug. When used in this manner, the D-plug pinout is identical to encoder plug_P1.

P10. PC BUSS (ISA bus) Interface

This is a standard 8-bit PC-XT/AT bus connector. Refer to PC-XT/AT technical manual for details. See Chapter 5 for more information regarding the PC BUSS (ISA bus) interface.

Power Plug

Note: this plug is not used for PC BUSS (ISA bus) applications.

1. GROUND.
2. +5 VOLTS.
3. -12VOLTS.
4. +12VOLTS.

For the following jumpers, orient the card so that plugs P1, P2 are on the right and the PC BUSS (ISA bus) connector is on the bottom. jumpers are considered in the ON position when they are shorting the top two pins. Jumpers are considered OFF when they are shorting the bottom two pins. If necessary, refer to the drawing at the end of the manual for more detail.

JS1-1 Serial/PC BUSS (ISA bus) Selection

JS1-1	MODE
ON	PC BUSS (ISA bus)
OFF	RS-232

JS1:2-4 BAUD Rate Jumpers

JS1-2	JS1-3	JS1-4	BAUD
ON	ON	ON	9600
ON	ON	OFF	9600 **
ON	OFF	ON	9600
ON	OFF	OFF	4800
OFF	ON	ON	2400
OFF	ON	OFF	1200
OFF	OFF	ON	600
OFF	OFF	OFF	300

** This setting also signals the N 8 1 mode on newer /16 firmware versions.

JS1:5-8: Card Address Jumpers

JS1-5	JS1-6	JS1-7	JS1-8	CARD ADDR. #
ON	ON	ON	ON	15
ON	ON	ON	OFF	14
ON	ON	OFF	ON	13
ON	ON	OFF	OFF	12
ON	OFF	ON	ON	11
ON	OFF	ON	OFF	10
ON	OFF	OFF	ON	9
ON	OFF	OFF	OFF	8
OFF	ON	ON	ON	7
OFF	ON	ON	OFF	6
OFF	ON	OFF	ON	5
OFF	ON	OFF	OFF	4
OFF	OFF	ON	ON	3
OFF	OFF	ON	OFF	2
OFF	OFF	OFF	ON	1
OFF	OFF	OFF	OFF	0

JS2:1-8 PC BUSS (ISA bus) Base Address Select (ACR1000/12 Only)

JS2	BASE ADDRESS
8	0300H
7	0304H
6	0308H
5	030CH
4	0310H
3	0314H
2	0318H
1	031CH

Chapter 4 The ACR1000 Serial (RS232) Port

The user has the option of communicating with the ACR1000 through either the RS-232 serial port or the PC BUSS (ISA bus) parallel interface. The RS-232 port provides a convenient interface to the ACR1000, requiring only a simple three-wire connection to a dumb terminal or a PC running communication software. This interface allows the ACR1000 chassis to be mounted up to 50 feet from the terminal or host computer.

For applications requiring an intelligent controller, a single RS-232 port can control up to 16 axes. Multiple axes may be connected to a single RS-232 line by daisy-chaining the serial ports of each ACR1000 card (see chapter 3 for signal details), creating a serial bus. Commands may be specifically directed to a single axis or broadcast to a group of axes. The ACH30X0 chassis has the necessary wiring to daisy-chain up to 3 axes. Multiple ACH-3000's can be wired together for up to 16 axes on a single line.

RS-232 Configuration

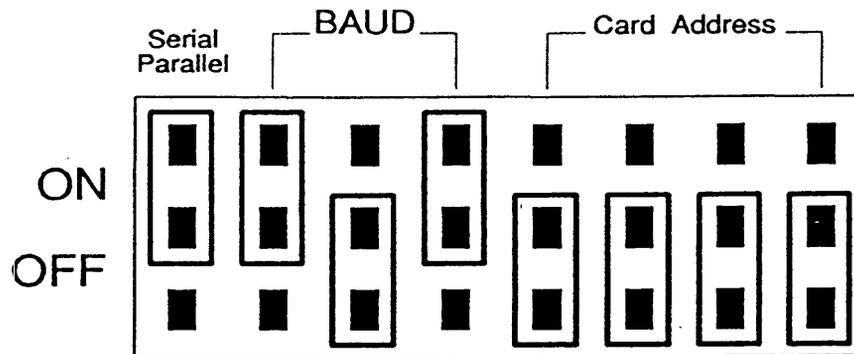
The RS-232 port is set to a standard configuration when the card is shipped. This standard configuration is as follows:

BAUD	9600	(jumper configurable)
DATA BITS	7	(fixed*)
STOP BITS	1	(fixed)
PARITY	ODD	(fixed*)

* one of the 9600 baud selections also selects N 8 1 mode on newer /16 versions.

Before going further, make sure all the JS1 jumpers are set properly. For serial communication JS1:1 must be off. If needed, the baud rate can be changed via JS1:2-4. The card address, which is factory set for 0 on a 1-axis system, can be changed via JS1:5-8. These are set at 0000 for the first axis, 0001 for the second axis...etc.

JS1- Jumper Selection



JS1 Jumper Settings

- JS1-1 = 1 ON selecting the card for PC BUSS (ISA bus) operation.
- JS1-2 = 1 ON
- JS1-3 = 0 OFF
- JS1-4 = 1 ON JS1-2,3,4 HAVE 9600 BAUD SELECTED.
- JS1-5 = 0 OFF
- JS1-6 = 0 OFF
- JS1-7 = 0 OFF
- JS1-8 = 0 OFF JS1-5,6,7,8 HAVE CARD ADDRESS 0 SELECTED.

Echo Control

In the parameter table, P1 controls whether the ACR1000 will echo characters received by it. Setting P1 to 0 (P1=0) will cause the ACR1000 to suppress the echo (half-duplex) but allow the ACR1000 to output any necessary messages. P1=1 enables echo (full duplex). P1=2 will disable any output from the card; only responses from status requests will be output. If the ACR1000 is being commanded with a terminal, set the terminal to full duplex and the ACR1000 to full duplex (P1=1). For applications involving a host computer, general output from the card should be disabled (P1=2) since information coming from the card (but not expected by the host program) will fill up communications buffers and cause confusion at the receiving

end.

If the card address is set to 0, the card will power-up in the full duplex mode (P1=1). For all other addresses, the cards power up in the output disabled (P1=2) mode and will only echo once P1 is set equal to 1.

For details on setting parameters, refer to chapter 6.

Communication Buffer

As the commands are received by the ACR1000 they are stored in a circular buffer. This buffer is 256 bytes long. If commands are entered into the ACR1000 board faster than the board can process them, the buffer may become full. Once this happens, before the buffer can overflow, the board sends out an XOFF character (hex 13). The host should stop sending data immediately. When buffer space becomes available, the board issues an XON character (hex 11), and the host can resume sending data. It is important that any communication software used with the ACR1000 be capable of properly responding to these XON/OFF characters. Failure to do so may result in lost characters and erratic operation.

When the ACR1000 card receives a command it is queued in the communications buffer until the processor can execute it. Commands involving moves are quickly processed and then execution of the move is done in the background. The CPU will then process the next command. If it is an immediate command, i.e. SET 12, then this command will be done and the next command in the queue will be processed. If another move command is encountered then the first move will be completed before the second is started. For example, if the commands

```
MOV 10000
SET 12
MOV 0
```

were sent to the ACR1000, the ACR1000 would start the move to position 10000 and almost immediately turn on output 12. Immediately after the move to 10000 was completed the move back to 0 would be started. If, on the other hand the commands

```
MOV 10000
MOV 0
SET 12
```

were sent, the output 12 would be set immediately after the move back to zero had been started. This can be a handy technique for signaling a host computer that a move has been completed.

Common Problems With Getting An RS232 Connection Working

There can be difficulties commonly encountered when setting up RS-232 communications. To resolve a problem, ask yourself the following questions:

Are the XMIT, RCV, and GND connections made properly?

A quick way to check this is to verify the voltages on these lines. XMIT from the terminal/host should connect to RCV on the ACR1000 and vice versa. An RS-232 signal at idle should sit at approximately -10V with respect to GND. Some devices such as the TRS-80 Model 100 only produce a -5V signal, so look for -5V if using one. If XMIT and RCV are swapped, the voltage on one or both of these signals will be lower than normal. If all else fails, just reverse XMIT and RCV and try again.

Does your communications setup require handshake lines to be used?

Many RS-232 ports require that the signals DCD, DSR, and CTS are active (pulled high) in order for communication to occur. Since the ACR1000 provides only a 3-wire interface, these signals are not provided. If this presents a problem, wire pins 5,6,8 and 20 together on the end of the RS-232 cable that connects to the PC.

Are the communications parameters the same as the ACR1000 settings?

If these are incorrectly set, you may see garbage echoed, or no communication at all. The terminal settings for these parameters MUST exactly match those set on the ACR1000. Refer to the previous section on ACR1000 configuration for details.

Are some characters occasionally dropped (not echoed?)

If your communication software does not support the XON/XOFF protocol properly, lost characters may be the result. When just typing in commands there may be no problem, but for uploading/downloading programs to/from the ACR1000, failure to support this protocol will result in disaster. Reducing baud rate may help, but it is best to ensure that your software supports this function properly.

Loading And Listing Of Programs On The ACR1000

With larger programs, it may be preferable to use the editing facilities of a word processor to write a program and then send the program to the ACR1000 via the RS-232 port. Some word processors, i.e. Word Star, include control characters within the text of files. These control characters may be harmless when uploaded to the ACR1000, but could potentially cause trouble. To be on the safe side save the program as text or ASCII only. If a program is already loaded in the ACR1000 card, type `NEW` to clear it before loading a new program (see `NEW`, chapter 7). Once the file exists and the RS-232 connection is established, uploading it to the ACR1000 is just a matter of telling the communications software to upload the file. To capture a program in the ACR1000 memory, perhaps to save it to disk, select "download" or "capture" within your communications program, then type `LIST`. The ACR1000 will begin sending the contents of program memory out the serial port. When the listing has completed, terminate the download or capture using the method your system requires. You will now have a text file suitable for archiving, re-editing, or printing.

Chapter 5 The ACR1000 PC BUSS (ISA bus) Interface

The user has the option of communicating with the ACR1000 through either the RS-232 serial port or the PC BUSS (ISA bus) parallel interface. The PC BUSS (ISA bus) interface has a definite advantage in speed of communication and overall throughput of command information. For interpolated motion requiring very high throughput, several commands have been adapted that will only function with the PC BUSS (ISA bus) interface. Also, since each card represents a discrete group of I/O addresses to the PC, each card can be addressed directly. Multiaxis systems benefit from not having to use the control character ON and OFF commands (ctrl-A and ctrl-B) required with the RS-232 port.

To use the card in the IBM PC/XT/AT environment requires more work on the PC end of things. However, AMCS provides an excellent demo program, AcroGEN, that will allow the user to simply install the ACR1000 cards in the PC and issue commands. The program has many of the same features of standard communications programs as well as a continuous display of card status, and support for a subset of the RS-274D standard.

Setting Jumpers Before Getting Started

Before using the card make sure all the jumpers are set properly. If the card is shipped from the factory for PC operation, there should be no need to make any changes.

In order to use the PC BUSS (ISA bus) at all, JS1:1 must be in the on position. The card address, which is factory set for 0 on a 1-axis system, is set by JS1:5-8. These are set at 0000 for the first axis, 0001 for the second axis...etc. For all jumpers, OFF = 0 and ON = 1. On then ACR1000/12 Card, in addition to the card address, JS2:1-8 must be set properly. The following table explains:

CARD ADDRESS	JS2 JUMPER POSITION	PORT ADDRESS
0 (First)	8	300-303
1 (Second)	7	304-307
2 (Third)	6	308-30B
3 (Fourth)	5	30C-30F
4 (Fifth)	4	310-313
5 (Sixth)	3	314-317
6 (Seventh)	2	318-31B
7 (Eighth)	1	31C-31F

For the ACR1000/16 the port addresses are automatically derived from the card address setting on JS1.

Echo Control

In the parameter table, P1 controls whether the Acroloop will echo characters received by it. Setting P1=0 will disable echo (HALF-DUPLEX), but allow the Acroloop to output any necessary messages. P1=1 enables echo (FULL DUPLEX). P1=2 will disable any output from the card; only responses from status requests will be output. For PC BUSS (ISA bus) applications, it is strongly recommended that general output from the card be disabled (P1=2).

If the card address is 0, the card will power-up in the full duplex mode (P1=1). For all other addresses, the cards power up in the output disabled (P1=2) mode and will only echo once P1 is set equal to 1.

For more details on setting parameters, refer to chapter 6.

Communications Buffer

As commands are entered, they go into a circular buffer. This buffer is 256 bytes long. If commands are entered into the ACR1000 board faster than the board can process them, the buffer may become full. Once this occurs, before the buffer can overflow, the board sends out an XOFF character (hex 13). The host should stop sending data immediately. When buffer space becomes available, the board issues an XON character (hex 11), and the host can resume sending data. It is important that the program written to command the ACR1000 be capable of properly responding to these control characters. Failure to do so may result in lost characters and erratic operation.

PC BUSS (ISA bus) Program Interface

In the IBM PC/XT/AT environment, the ACR1000 card occupies 4 PORT addresses. There is an 8255A-5 (on the ACR1000/12) the card that is used as a an interface between the ACR1000 and the IBM PC. CARD0 represents addresses 0300(PORTA), 0302(PORTC) and 0303 (CONTROL). Data is written and read from 0300(PORTA), status is read from 0302(PORTC).

The first thing the PC must do before attempting communication with the ACR1000 is to send C3h to address 303. This sets up the 8255A-5 properly for bi-directional communication. You will have no need to write to this port again. Also, doing this is entirely optional on the ACR1000/16.

From this point on, characters may be sent to the ACR1000 by polling the status port (portC) until the OK_TO_SEND signal is set, then writing the character to the data port (portA). Writing to the data port will clear the OK_TO_SEND bit.

To read a character from the ACR1000, simply poll the status port until DATA_AVAILABLE is set, then read the data port (portA). Reading from the data port will clear the DATA_AVAILABLE bit.

The status port (portC) is bit-mapped as follows:

BIT POSITION	USAGE
7	OK_TO_SEND
6	NOT USED
5	DATA_AVAILABLE
4	NOT USED
3	NOT USED
2	RESET UP/DN COUNT (ACR1000/12 Only)
1	COUNT UP (ACR1000/12 Only)
0	COUNT DOWN (ACR1000/12 Only)

Note: that the RESET UP/DOWN COUNT, COUNT UP and COUNT DOWN are inputs to the ACR1000/12. These can be commanded to set or clear by the PC software.

By sending a HDWON 2 or HDWON 4 command (ACR1000/12 Only), the axis can be made to follow the COUNT UP and COUNT DOWN pulses from the PC software. (See chapter 7 HDWON command.)

Preliminary I/O Procedures

The best way to work with the PC BUSS (ISA bus) interface is to write the low-level routines that will interface with the ACR1000 on the character level. Once these character routines have been written, then create routines that will send and receive strings to and from these routines. Once this is set up, any command may be sent to, and any status information may be received from, the ACR1000 easily.

See the following pages for an example of "Get Actual Status" implementation. (Turbo C)

```
/* Get Actual Status */
```

```
#define ACRO_STAT_ACTUAL 21
#define ACRO_NO_ERROR 0
#define ACRO_TIMEOUT_ERROR -1

#define acro_data_available(axis) (inportb(acro_statport[axis])&0x20)
#define acro_clear_to_send(axis) (inportb(acro_statport[axis])&0x80)

/*****
/*Port assignments      axis0 axis1 axis6 axis7      */
*****/

int acro_dataport[] = { 0x300,0x304,0x308,0x30C,0x310,0x314,0x318,0x31C };
int acro_xtraport[] = { 0x301,0x305,0x309,0x30D,0x311,0x315,0x319,0x31D };
int acro_statport[] = { 0x302,0x306,0x30A,0x30E,0x312,0x316,0x31A,0x31E };
int acro_ctrlport[] = { 0x303,0x307,0x30B,0x30F,0x313,0x317,0x31B,0x31F };

/*****
/* acro_putchar      */
/*      */
/* Sends a character to an axis after waiting for      */
/* the transmit register to empty out.      */
/*      */
/* Returns ACRO_TIMEOUT_ERROR if unable to send the      */
/* character, otherwise returns the character c.      */
*****/

int acro_putchar(int c, int axis)
{
    int count=0;

    while (!acro_clear_to_send(axis))
        if (count++ >= ACRO_TIMEOUT) return ACRO_TIMEOUT_ERROR;

    outportb(acro_dataport[axis],c);
    return c;
}

/*****
/* acro_getchar      */
/*      */
/* Gets a character from an axis after making sure      */
/* that something is in the receive register.      */
/*      */
/* Returns ACRO_TIMEOUT_ERROR if no character is      */
/* available, otherwise returns the character read      */
/* from the axis.      */
*****/

int acro_getchar(int axis)
{
    int count=0;

    while (!acro_data_available(axis))
        if (count++ >= ACRO_TIMEOUT) return ACRO_TIMEOUT_ERROR;
    return inportb(acro_dataport[axis]) & 0xFF;
}
```

```

/*****
/*  acro_gets                                     */
/*                                             */
/*  Gets a string from an Acroloop.  Terminates on a  */
/*  carriage return or after receiving n characters.  */
/*  The carriage return is not stored with the string.  */
/*                                             */
/*  Buffer area pointed to by s must be at least n+1  */
/*  in length in order to hold '\0' at end of string.  */
/*                                             */
/*  Returns ACRO_TIMEOUT_ERROR if unable to receive  */
/*  the string, otherwise returns length of string.  */
*****/

```

```
int acro_gets(char *s, int n, int axis)
```

```
{
    int code, count=0;
    char *p = s;

    while (count < n) {
        code = acro_getchar(axis);
        if (code == '\r') break;
        if (code < 0) return code;
        *p++ = (char)code;
        count++;
    }
    *p = '\0';
    return count;
}

```

```

/*****
/*  acro_get_actual                             */
/*                                             */
/*  Fetches "actual" position from an Acroloop card.  */
/*  The actual position is where the card actually is  */
/*  and is derived directly from the encoder feedback.  */
/*  Returns 0L if unable to read actual position.  */
*****/

```

```
long acro_get_actual(int axis)
```

```
{
    long actual, dummy;
    int code;
    char s[20];

    code = acro_putchar(ACRO_STAT_ACTUAL, axis);
    if (code < 0) return 0L;
    code = acro_gets(s, sizeof(s)-1, axis);
    if (code < 0) return 0L;
    sscanf(s, "%08lx%08lx", &actual, &dummy);
    return actual;
}

```

Chapter 6 System Parameters

The ACR1000 firmware uses several system variables to configure operation. By manipulating the values of these system parameters the user may customize how the ACR1000 responds to the machine that it is controlling. These system parameters control such things as how the machine homes, how encoder signals are interpreted and maximum machine speed. Several parameters are redefined for use with the WEB command. In addition to these system parameters there are several undefined parameters that the user may use as they wish.

Parameter Definitions

	DESCRIPTION	SIZE
P1	Serial Port Control	16 BITS
P2	Axis Feedback Select And Multiplier	16 BITS
P3	Aux Feedback Select And Multiplier	16 BITS
P4	JOG Feedrate	16 BITS
P5	JOG ACCEL	16 BITS
P6	JOG DECEL	16 BITS
P7	Maximum Velocity	16 BITS
P8	Unidirectional Approach / Y Axis Max Velocity v5.x	16 BITS
P9	Backlash Compensation / Web-Offset / 3rd encoder v5.x	16 BITS
P10	3rd Encoder Multiplier v3.x / 4th encoder v5.x	16 BITS
P11	4th Encoder Multiplier v3.x	16 BITS
P12	Spare	16 BITS
P13	Spare	16 BITS
P14	Spare	16 BITS
P15	Spare	16 BITS
P16	Spare	16 BITS
P17	(v5.x)Y-Axis Velocity Damping Gain (Software Tach)	16 BITS
P18	(v5.x)Y-Axis Feed-Forward Gain	16 BITS
P19	X-AXIS Velocity Damping Gain (Software Tach)	16 BITS
P20	X-AXIS Feed-Forward Gain	16 BITS
P21	HOME ACC/Spare	16 BITS
P22	HOME DEC/Spare	16 BITS
P23	Velocity To Home Switch/Spare	16 BITS
P24	Velocity Away From Switch/Spare	16 BITS
P25	Home Limit Switch Input/Spare	16 BITS
P26	Direction To Home To/Spare	32 BITS
P27	Move Cmd Floating Zero	32 BITS
P28	v2.7.x, v4.x Master Move In Interpolation	32 BITS
P29	v2.7.x, v4.x Destination In Arc Move	32 BITS
P30	v5.x Y Floating Zero/Spare	32 BITS
P31	Spare	32 BITS
P32	Spare	32 BITS
P33	Spare	32 BITS
P34	Spare	32 BITS
P35	Spare	32 BITS
P36	Spare	40 BITS
P37	Spare	40 BITS
P38	Spare	40 BITS
P39	Spare	40 BITS
P40	Spare	40 BITS
P41	Spare	40 BITS
P42	Spare	40 BITS
P43	Spare	40 BITS
P44	Spare	40 BITS
P45	Spare	40 BITS
P46	Spare	40 BITS
P47	Spare	40 BITS
P48	Spare	40 BITS
P49	Spare	40 BITS
P50	Spare	40 BITS

Entering System Parameters

There are 50 system parameters available. They are referenced by a 'P' followed immediately by the parameter number, i.e. P1, P2, P3...P50. Parameters P1 through P25 will only hold values in the range of ± 32767 (16 bits). Parameters P26 through P35 will contain values ± 2147483647 (32 bits). Parameters 36 through 50 are floating point parameters with 40 bit precision and can have a mantissa of ± 2147483647 and exponent of ± 128 . Even though these variables can hold different values, this does not mean that all values are *valid*. For example: the variable P1 can hold ± 32767 , but only the values 0, 1 & 2 are defined as *valid*. Using values outside of a parameters defined domain can have unpredictable results. It is the responsibility of the programmer to insure that proper values are assigned to these parameters before they are used.

To enter a value into a parameter, the assignment command, "=", is used. This command can be executed either in the immediate mode or within a program. For example, to load parameters 1,2,3 with the values 1000, 2599, 3000 use the following commands:

```
P1=1000
P2=2599
P3=3000
```

P1: Serial Port Control

P1 controls the RS232 serial port. P1 can be set to a value of 0,1 or 2 to get half duplex, full duplex or disable output modes respectively. This parameter is set to it's default value during a reset or power-up. If the card address is set to zero (the first card address) the default value for this parameter is 1, otherwise it is 2.

P1=0: HALF DUPLEX

In this mode the ACR1000 will not echo the characters being sent to it, but will respond to all queries and commands as usual. Error messages will also be sent. This mode should be used when the card is being commanded by a host computer and the host needs to have access to the PRINT capabilities of the ACR1000. (If the card, in this mode, is being commanded by the user from a terminal, the terminal ECHO should be turned ON.)

P1=1: FULL DUPLEX

In this mode the ACR1000 will echo all characters sent to it as well as any response it might have. This mode should be used when the card is being programmed or commanded by the user from a terminal. (In this case the terminal should have the ECHO turned OFF.)

P1=2: Disable Output.

In this mode, the card will suppress character echo as well as any messages. It will only respond to status requests from the host. This mode is most often used when the ACR1000 is being controlled by a host computer. Since the host only needs specific information, requested via status commands, and does not want any extraneous information cluttering up it's input buffers, the output of the ACR1000 is turned off.

P2,P3: Main/Aux (Handwheel) Feedback Multipliers

The following table explains how to select times 1, times 2 or times 4 multipliers on the main Axis and Handwheel encoder:

VALUE	RESULTING MULTIPLIER
0000	x1
0001	x2
0002	x4
0003	DISABLE ENCODER Note that this is useful in some applications where it is desirable to shut off the encoder pulses at certain times.
0004	(/16 only) x1, use 12 bit (0-1024) A/D for feedback
0005	(/16 only) x2, use 12 bit (0-2048) A/D for feedback
0006	(ACR1000/16) x4, use 12 bit (0-4096) A/D for feedback
0007	(ACR1000/16) DISABLE ENCODER
>0007	These values are reserved. Do not use.

Example: If the encoder shaft turns 5 times for every inch of linear motion and the encoder has 500 line resolution, to get 10000 pulses per inch axis feedback, P2 should be set to 2 as 500 lines times 5 turns times 4 equals 10000. For velocity, distance, acc/dec, IPR, and maximum pulse rate calculations, the number of lines/rev of the encoder AND the multiplier must be taken into account. As far as the ACR1000 is concerned, the number of pulses/rev (PPR) of the encoder is equal to lines/rev x MULTIPLIER.

NOTE: Changing P2 or P3 has no effect on feedback resolution until either a RES or REN or possibly RES 2 command has been executed. (See command descriptions for details.)

Jog Parameters

P4, P5 and P6 select the JOG VELOCITY, ACCEL and DECEL respectively. For example

P4 = 500: REM This sets the jog velocity to 500

P5 = 10: REM This sets the jog accel to 10

P6 = 10: REM This sets the jog decel to 10

If the JOG parameters are left at zero, the system uses the currently programmed VELOCITY, ACCEL and DECEL values. If the programmed ACCEL or DECEL values are 0, JOG will default to using values of '1'.

P7, P8: Maximum Velocity

P7 controls the maximum velocity at which the ACR1000 will permit the motor to move. For example, if a motor has a maximum speed of 2500 RPM it would be dangerous to try to move it faster. In this case it might be wise to clamp the maximum speed to 2500. Setting P7 to 2500 (P7=2500) will do this. If P7 is 0, the velocity is not clamped at all. P8 is the maximum velocity parameter for the Y axis in v5.x.

P8: Unidirectional Approach (v2.6.x & v3.x)

Because some machine movement destinations are more reproducible if the machine is moving to the destination from the same direction each time, it may be desirable to ensure that this happens. For example, a drill head is moved over an x-y table to a series of points where holes are drilled. Because of backlash in the ball screws that move the drill, there is a variation in the accuracy of the positioning of the drill. This variation is dependent on the direction of approach for each axis. Using UNIDIRECTIONAL APPROACH, this variation can be eliminated because each destination is approached from the same sides for each hole, regardless of where the drill started from.

This parameter allows control the final approach direction for all movement. To use this feature, set P8 equal to the distance the axis should move beyond the target before the final approach—generally a bit larger than the maximum backlash. The sign of P8 determines the final approach direction. Output 119 must be turned on (SET 119) to enable this feature. Executing a CLR 119 will disable this mode and return P8 to general purpose use. Relay 119 is normally clear, to retain compatibility with older applications that may use P8 for other purposes.

P9: Backlash Compensation (v2.6.x & v3.x)

For precise applications where backlash is a problem and UNIDIRECTIONAL APPROACH cannot be used, backlash can be compensated for by setting P9 equal to the amount of backlash. Execute a SET 120 to enable this feature. Executing a CLR 120 will disable this feature and return P9 to general purpose use. Relay 120 is normally clear.

P10,P11: 3rd & 4th Encoder Multipliers (v3.x & v5.x)

When the piggy back adapter for 3rd and 4th Encoder inputs is plugged into the ACR1000/16 card, (P10 & P11 v3.x), (P9 & P10 v5.x) control the scaling of the encoders. In either case, (P10 & P11 v3.x), (P9 & P10 v5.x) work the same as P2 & P3 in being able to allow x1, x2 and x4 multiplication to the encoder lines.

VALUE OF P10, P11	v3.x	RESULTING MULTIPLIER
P9, P10	v5.x	
0000		x1
0001		x2
0002		x4
0003		DISABLE ENCODER
>0003		These values are reserved. Do not use.

P17,P19: Velocity Damping Gain (Software Tachometer) (v2.6.x, 3.x & 5.x)

For applications that use current mode amplifiers, the ACR1000 can modify the amplifier signal by adding a velocity feedback term. This adds a damping effect. P19 controls the gain of this simulated tachometer. The valid range for this parameter is 0 through 255. This parameter must be set to zero for applications that have a stabilized velocity loop. Current Mode operation is explained in more detail in chapter 13. P17 controls the damping gain of the Y axis in v5.x.

P18,P20: Feed-Forward Gain (v2.6.x, 3.x & 5.x)

Often it is desirable to have zero following error. The ACR1000 executive has provisions to generate Feed-Forward based on the command velocity. This allows anticipating the velocity before any following

error is generated. P20 controls the amount of "anticipation" and can be any value between ± 255 . A positive value causes the ACR1000 to increase its signal as velocity increases, decreasing the amount of following error. A negative value acts in an opposite manner, increasing the following error. If P20 is set too high the motor will overshoot its target position. To check the following error, use the ctrl-P status command. P20, P18 must be set to zero if no feed forward gain is desired. P20 affects the primary axis and P18 affects the 2nd axis. P18 works as feed forward gain in version 5.x software. P18 is the Feed-Forward parameter for the Y axis in v5.x.

Home Sequence Parameters

P21, P22, P23, P24, P25, P26 are used in the HOME sequence. See the HOME command for more detail.

P27,P30: Floating Zero

P27 is used by the FLZ command and will contain the value of the Target Position minus the FLZ command's argument after the FLZ command is executed. Once and FLZ is done, the value in P27 is added to each subsequent MOV. See FLZ in chapter 7 for more details. P30 is the floating zero for the Y Axis in v5.x. HOME clears the FLZ values in versions 4.072 and above.

Interpolation Parameters (v2.7.x & v4.x)

P28 and P29 are used in conjunction with interpolated moves. P28 must be loaded with the maximum number of pulses any axis will move with all axes moving together. P29 is only used for circular interpolated moves, and must be loaded with the absolute destination for each axis. For details on these parameters see chapter 8.

Array Variables

The ACR1000 allows users the ability to use memory as an array to store data. One useful application would be a "teachable" system that would capture axis positions, store them in an array, and later use the array to reproduce the series of movements. Also, having movements stored on the card eliminates delays associated with sending large quantities of data from a host computer.

Only one array is available on the ACR1000. Once the array is DIMensioned (see Chapter 7 commands CLEAR, DIM and MEM), array elements may be used anywhere standard system parameters can. This includes math and arguments for any of the other commands.

Using the DIM command, the user can select the variables to be one of the following sizes:

- 16 bits ± 32767
- 32 bits ± 2147483647
- 40 bits Floating Point.

An array element is accessed by a P followed by an *index* in parentheses, where index 0 is the first element of the array. The legal range for *index* is from 0 to (# of elements-1).

The standard method of assigning a value to a parameter has been extended for array usage. If a list of data separated by commas is assigned to a array variable, the data will be place in the array sequentially starting at the specified array index. Remember that a maximum of 256 characters may sent on a single line.

Examples:

```
P(30)=14 : P(0)=1,2,3,4,5,6,7,8,9,10  
ADD 12,P(20) : MOVE P(55)
```



Artisan Technology Group is your source for quality new and certified-used/pre-owned equipment

- FAST SHIPPING AND DELIVERY
- TENS OF THOUSANDS OF IN-STOCK ITEMS
- EQUIPMENT DEMOS
- HUNDREDS OF MANUFACTURERS SUPPORTED
- LEASING/MONTHLY RENTALS
- ITAR CERTIFIED SECURE ASSET SOLUTIONS

SERVICE CENTER REPAIRS

Experienced engineers and technicians on staff at our full-service, in-house repair center

*InstraView*SM REMOTE INSPECTION

Remotely inspect equipment before purchasing with our interactive website at www.instraview.com ↗

WE BUY USED EQUIPMENT

Sell your excess, underutilized, and idle used equipment. We also offer credit for buy-backs and trade-ins. www.artisanng.com/WeBuyEquipment ↗

LOOKING FOR MORE INFORMATION?

Visit us on the web at www.artisanng.com ↗ for more information on price quotations, drivers, technical specifications, manuals, and documentation

Contact us: (888) 88-SOURCE | sales@artisanng.com | www.artisanng.com