UNIDEX 100

IE-488

OPTION MANUAL

PN:EDU129

AEROTECH, INC., 101 Zeta Drive, Pittsburgh, PA 15238
(412) 963-7470 • TWX 710-795-3125 • FAX (412) 963-7459

IDENTIFICATION

for

Unidex 100 Series
IE-488 Option Manual

by

Aerotech, Inc.
101 Zeta Drive
Pittsburgh, PA 15238-2897

Manual Part Number - EDU129

Version 1.1

Released - November 23, 1993

1st Printing -July 29, 1993

Supplementary Material -

Table of Contents

SECTION PAGE

CHAPTER 1
INTRODUCTION

CHAPTER 2
GETTING STARTED

CHAPTER 3
PROGRAMMING THE U100

CHAPTER 4
HARDWARE

SECTION **PAGE**

CHAPTER 5

TROUBLE SHOOTING

APPENDICES

INDEX

List of Figures

CHAPTER 1: INTRODUCTION

The IE-488 option for the UNIDEX 100 permits the user to control the UNIDEX 100 through the IE-488 bus. Using the HOST command set, the IE-488 offers many control capabilities. These include: executing programs and commands, transferring files, and reading or modifying data.

Before continuing, it is best that the user reviews the *Unidex 100 User's Manual*.

This Page Intentionally Left Blank.

CHAPTER 2: GETTING STARTED

In this chapter you will find information about the initial setup of the IE-488 option. Included is information describing the interface connections, IE-488 parameter setup, and the initial IE-488 power up.

2.1: IE-488 INTERFACE CONNECTIONS

The IE-488 interface is an 8 bit parallel data bus along with eight control lines. It can accommodate up to fifteen devices on the bus. This bus also provides a method of requesting attention from the Host controller. We will refer to this method as the Service Request (see Section 4.2.5 in the *Unidex 100 User's Manual*). To properly connect the IE-488 device, all connections made by the user must include approved IE-488 cables. The total cable length must not exceed twenty meters (see IE-488 cable list in Appendix A).

2.2: SETTING THE UNIDEX 100 IE-488 PARAMETERS

To use the IE-488 Mode it is necessary to configure the UNIDEX 100. To do this the user must set the IEEE address parameters (PRM:008 and PRM:009) and the RS-232-C/IE-488 parameter (PRM:019) for the IE-488 mode of operation. For details involving these parameters, see the *Unidex 100 User's Manual.*

PRM:008

This parameter is the default value that the IEEE Address Register #0 will use. This parameter sets the mode 0 address, the mode 0 talker enable bit, and the mode 0 listener enable bit. The lower 5 bits are the device address bits (default is 4). Bit #6 is the listener bit (default is enabled). Bit #7 is the talker bit (default is disabled). The parameter default is 0x24.

PRM:009

This parameter is the default value that the IEEE Address Register #1 will use. This parameter sets the mode 1 address, the mode 1 talker enable bit, and the mode 1 listener enable bit. The lower 5 bits are the device address bits (default is 4). Bit #6 is the listener bit (default is disabled). Bit #7 is the talker bit (default is enabled). The parameter default is 0x44.

PRM:019

This parameter allows the user to select either the RS-232 or the IE-488 mode of operation. Setting this parameter to a 1 will enable the RS-232 mode of operation while setting it to a 2 selects the IE-488 mode. Set the value of this parameter last. That is after setting all other IE-488 parameters. The parameter default is a 1 (RS-232).

NOTE: Parameter 008 and 009 should normally use the same address (lower 5 bits). Also, normally the upper bits do not change, nor should they change. For instance, if the user wishes to change the IE-488 device to #7, Parameter #008 should equal 0x27 and Parameter #009 equal 0x47.

NOTE: To get back all the default parameter values the user must perform a setup. See the *Unidex 100 User's Manual.*

2.3: INITIAL IE-488 POWER UP

By now, the user should have 1.) completely connected the IE-488 system controller using the standard IE-488 cables and 2.) set the parameters for the IE-488 operation. Turn on power (allowing approximately 10 seconds for the UNIDEX 100 to initialize) and the UNIDEX 100 will be ready to receive its first IEEE command. To verify that the IE-488 mode is the current operating mode, do not disconnect the RS-232 device from the UNIDEX 100. Then, verify that the RS-232 is not operational. If it is possible to attain RS-232 operation, parameter #019 may contain an incorrect setting. Verify that the setting for this parameter (PRM:019) is a two. To test the IEEE operation the user must send a host mode command to the UNIDEX 100 (recommend an immediate motion command). See Chapter 3 in this manual for information concerning how to program the UNIDEX 100.

NOTE: If the LED's on the UNIDEX 100 front panel are blinking, this may indicate that the UNIDEX 100 IEEE interface is not operational.

This Page Intentionally Left Blank.

CHAPTER 3: PROGRAMMING THE U100

While in the IEEE mode of operation, this Chapter shows how to program the UNIDEX 100 through the Host Mode Command Set. Along with these commands the UNIDEX 100 incorporates a Service Request. The following sections describe how to program the UNIDEX 100 while in this mode.

3.1: HOST MODE OPERATION

To control the UNIDEX 100 in IEEE mode it is necessary that the user use the Host Mode commands. These commands are special code formats that the UNIDEX 100 translates into commands. For information regarding the Host Mode commands, refer to Appendix B in this manual.

3.2: SERVICE REQUEST

The UNIDEX 100 uses the Service Request line to request attention from the Host Controller when operating in IE-488 communication mode. Upon activating the Service Request, it is necessary for the Host Controller to serial poll the UNIDEX 100. Failure to poll the UNIDEX 100 will result in the appearance of a locked up UNIDEX 100.

The Service Request consists of an SRQ status code. The UNIDEX 100 sends multiple SRQ's on an individual basis with each one requiring its own serial poll. The SRQ status code provides information concerning the Service Request. To obtain the true SRQ status code, the user must remove the SRQ bit by subtracting 64 (0x40) from the received SRQ status byte. If the SRQ status is equal to 3 (after removing the SRQ bit) the user must read in up to 3 additional bytes to obtain the error code before any commands can be sent to the UNIDEX 100. Failure to read the error codes can result in a lockup of the UNIDEX 100. See the *Unidex 100 User's Manual* for additional information concerning Service Request codes. For example:

A received SRQ = 67 (0x43)
= 67-64 = 3 (true SRQ code).

Since 3 represents an error, the user must read in up to 3 bytes to obtain the error code.

Task 0 Service Request Status Codes

0x00	Reserved
0x01	Program Uploading in Progress
0x02	Program Downloading in Progress
0x03	Host Mode Command Processing Error
0x04	Memory Allocation Error
0x05	"Line #" Update for "Task 1 Window" Running in Host Mode
0x06	"Line #" Update for "Task 2 Window" Running in Host Mode
0x07	Ready for Commands in Host Mode (RS-232 only)

Task 1 Service Request Status Codes

0x10	Reserved
0x11	User Defined #1
0x12	User Defined #2
0x13	User Defined #3
0x14	User Defined #4
0x15	User Defined #5
0x16	User Defined #6
0x17	User Defined #7
0x18	User Defined #8
0x19	User Defined #9
0x1A	User Defined #10
0x1B	User Defined #11
0x1C	User Defined #12
0x1D	User Defined #13
0x1E	User Defined #14
0x1F	User Defined #15

Task 2 Service Request Status Codes

0x20	Reserved
0x21	User Defined #1
0x22	User Defined #2
0x23	User Defined #3
0x24	User Defined #4
0x25	User Defined #5
0x26	User Defined #6
0x27	User Defined #7
0x28	User Defined #8
0x29	User Defined #9
0x2A	User Defined #10
0x2B	User Defined #11
0x2C	User Defined #12
0x2D	User Defined #13
0x2E	User Defined #14
0x2F	User Defined #15

Kernel Service Request Status Codes

0x30	Reserved
0x31	Initiate "Xon" (allow RS-232 Host to transmit)
0x32	Initiate "Xoff" (stop RS-232 Host from transmitting)
0x33	SRQ Exception Mask encountered a "true" condition

NOTE: IEEE Service Request Status Codes may contain Bit 6 set. (e.g., Status Code 0x01 may be received as 0x41).

3.3: ERROR AND STATUS CODES

Descriptions on the Error and Status codes for the UNIDEX 100 appear in Section 4.7 of the UNIDEX 100 User's Manual. To obtain these codes the user may use the Host mode commands to read the associated registers. Below you will find a list of the most important UNIDEX 100 Status Registers.

Register 015	-	Compiler Errors
Register 016	-	Task 1 Run Time Error
Register 017	-	Task 2 Run Time Error
Register 018	-	System Library Access Error
Register 302	-	Error Status
Register 303	-	Axis Status
Register 304	-	System Status

3.4: PROGRAM EXAMPLES

In this section you will find IEEE program examples used with the UNIDEX 100. These examples are very basic and serve as an aide in the understanding of the IE-488 interface. The following examples demonstrate an immediate mode move, writing and reading a BV variable, and reading a program from the UNIDEX 100.

3.4.1: HP85 Immediate Mode Command

The following program is an HP85 program used to issue an immediate mode motion command to the UNIDEX 100.

```
10 !                               ; HP85 Immediate Mode command to move
                                   ; axis.
20 ! Move Axis
30 !
40 IMAGE #, "#CBD(1000)", "↑"      ; Host Mode Immediate Mode command
50 OUTPUT 704 USING 40 ;           ; Send command to U100
60 END                             ; End of Program
```

3.4.2: HP85 Write & Read BV Variable

The following program is an HP85 program used to load BV:28 with a value of 10 and also read the value of BV:28 back.

```
10 !
20 ! Write & Read BV:28             ; HP85 Write & Read BV:28
30 !
40 IMAGE #, "#FCCA", 2Z, "↑B"       ; Host Mode command for BV:28 Write
50 IMAGE #, 2Z, "↑"                 ; Specify # value digits and LF
60 OUTPUT 704 USING 40 ; 28         ; Send Host Mode command for BV:28
                                    ; Write
70 OUTPUT 704 USING 50 ; 10         ; Send new value for BV:28
80 IMAGE #, "#FCCA", 2Z, "↑A"       ; Host Mode command for BV:28 Read
90 IMAGE #%, #%K                    ; Input Specifications
100 OUTPUT 704 USING 80 ; 28        ; Send Host Mode command for BV:28
                                    ; Read
110 ENTER 704 USING 90 ; X$         ; Input BV:28 value
120 DISP X$                         ; Display BV:28 value
130 END                             ; End of Program
```

3.4.3: HP85 Retrieve Program from UNIDEX 100

This program is an HP85 program used to acquire a program from the UNIDEX 100. This program also contains a serial poll example.

```
10 !
20 !                                              ; HP85 get program #2
30 !
40 DIM Z$[1000]                                   ; Provide program buffer
50 IOBUFFER Z$                                    ; Input/Output buffer
55 B=0                                            ; Serial poll flag
60 ON INTR 7 GOSUB 1000                           ; Setup interrupt conditions
70 ENABLE INTR 7 ; 8                              ; Enable interrupt
80 REMOTE 704                                     ; Device #4 remote
90 IMAGE #, #FDBA", 1Z, "↑"                       ; Host Mode command to request
                                                  ; program
100 OUTPUT 704 USING 90 ; 2                       ; Request program #2 for U100
110 IF B=0 THEN GOTO 110                          ; Wait until serial poll is performed
120 ON EOT 7 GOTO 200                             ; Set up end of transmission entry
                                                  ; line
130 TRANSFER 704 to Z$ INTR ; DELIM 126           ; EOF terminates
140 RESUME 7                                      ; Continue
150 ! Loop Till Data Terminated                   ; Loop again
160 GOTO 150                                      ; Display program #2
200 DISP Z$                                       ; End of Main Program
210 END

1000 !
1010 ! Service Request
1020 !
1030 S=SPOLL (704)                                ; Do serial poll
1040 STATUS 7, 1 ; C0                             ; IEEE status
1050 B=1                                          ; Set serial poll flag
1060 ENABLE INTR 7 ; 8 @ RETURN                   ; Enable serial poll interrupt
1100 END                                          ; End of Program
```

3.4.4: QB Immediate Mode Command

The following program is a Quick Basic program used to issue an immediate mode motion command to the UNIDEX 100. Initially, Aerotech wrote this program for the National Instruments GPIB board using the Universal language interface "HP-Style Calls".

```
/IEEE BASIC PROGRAM #2, IMMEDIATE MOVE COMMAND
/INITIALIZE THE INPUT & OUTPUT CHANNELS
OPEN "gpib0" FOR OUTPUT AS #1                      ; IEEE output channel
OPEN "gpib0" FOR INPUT AS #2                       ; IEEE input channel
/INITIALIZE THE BUS & RESET TO DEFAULT PARAMETERS
PRINT #1, "ABORT"                                  ; Initialize IEEE port
PRINT #1, "RESET"                                  ; Reset IEEE port
PRINT #1, "GPIBEOS OUT CR"                         ; IEEE output string terminator
PRINT #1, "TIMEOUT 0"                              ; Timeout

/PLACE THE DEVICE IN THE REMOTE STATE
PRINT #1, "REMOTE 4"                               ; IEEE device #4 REMOTE

PRINT #1 "CLEAR 4"                                 ; Clear device #4

/SEND THE COMMANDS TO THE U100                     ; Immediate Mode Host Command
PRINT #1, "OUTPUT 4 ; #CBD(1000)"                  ; "MOTION"

END                                                ; End of Program
```

3.4.5: QB Write & Read BV Variable

The following program is a Quick Basic program used to load BV:28 with a value of 10 and also read the value of BV:28 back. Aerotech wrote this program for the National Instruments GPIB board using the Universal Language. "HP-style Calls".

```
/ IEEE BASIC PROG, WRITE, READ BV:28
/INITIALIZE THE INPUT & OUTPUT CHANNELS
OPEN "gpib0" FOR OUTPUT AS #1                     ; IEEE Output Channel
OPEN "gpib0" FOR INPUT AS #2                      ; IEEE Input Channel
/INITIALIZE THE BUS & RESET TO DEFAULT PARAMETERS
PRINT #1, "ABORT"                                 ; Initialize IEEE Port
PRINT #1, "RESET"                                 ; Reset IEEE Port
PRINT #1, "GPIBEOS OUT CR"                        ; IEEE Output String Terminator
PRINT #1, TIMEOUT 0"                              ; Timeout
/PLACE THE DEVICE IN THE REMOVE STATE
PRINT #1, "REMOTE 4"                              ; IEEE Device #4 Remote

PRINT #1, "CLEAR 4"                               ; Clear IEEE Device #4

/SEND THE COMMANDS TO THE U100
PRINT #1, "OUTPUT 4 ; #FCCA28"                    ; Send Host Command for BV:28
PRINT #1, "OUTPUT 4 ; A"                          ; Read
PRINT #1, "ENTER 4"                               ; IEEE for Input
INPUT #2, A$                                      ; Get BV:28 value
PRINT A$                                          ; Display BV:28 value
END                                               ; End of Program
```

3.4.6: Retrieve Program from UNIDEX 100

The following program is a Quick Basic program used to get a program from the UNIDEX 100. Aerotech wrote this program for the National Instruments GPIB board using the Universal language Interface "HP-Style Calls". This program will acquire program No. 2 from the UNIDEX 100.

```
/IEEE QBASIC PROG, GET PROGRAM #2
/INTERRUPT SERVICE SETUP & ENABLE
ON PEN GOSUB INTSRVC                              ; Set up serial poll interrupt
PEN ON                                            ; Enable Interrupt
/INITIALIZE THE INPUT & OUTPUT CHANNELS
OPEN "gpib0" FOR OUTPUT AS #1                     ; IEEE output channel
OPEN "gpib0" FOR INPUT AS #2                      ; IEEE input channel
/INITIALIZE THE BUS & RESET TO DEFAULT PARAMETERS
PRINT #1, "ABORT"                                 ; Initialize IEEE
PRINT #1, "RESET"                                 ; Reset IEEE
PRINT #1, "GPIBEOS IN CHR(/x7E)"                  ; IEEE input string terminator EOF
PRINT #1, "GPIBEOS OUT CR"                        ; IEEE output string terminator
/PLACE THE DEVICE IN THE REMOTE STATE
PRINT #1, "REMOTE 4"                              ; Remote device #4
PRINT #1, "CLEAR 4"                               ; Reset device 4
/GET PROG #2 FROM U100
A = 0                                             ; Clear serial poll interrupt flag
CLS                                               ; Clear screen
PRINT #1, "OUTPUT 4 ; #FDBA2"                     ; Host command to get program #2
PRINT "WAIT FOR INTERRUPT"                        ; Display wait for interrupt
WHILE A = 0                                       ; Loop till serial poll performed
WEND                                              ; Loop back to previous line
PRINT #1, "GPIEBOS IN CHR(/x07E) CHR(/x07E)       ; Change terminator to 2 EOF's
PRINT #1, "ENTER 4"                               ; Enable input
INPUT #2, P$                                      ; Read in program #2
PRINT P$                                          ; Display program
END                                               ; End of Main Program
INTSRVC:                                          ; Serial poll interrupt subroutine
PRINT #1, "SPOLL 4"                               ; Do serial poll
INPUT #2, SP%                                     ; Get serial poll
A = 1                                             ; Set serial interrupt flag
PRINT #1, "STATUS"                                ; Request IEEE status
INPUT #2, IBSTA%, IBERR%, IBCNT%                  ; Get status
PRINT SP%                                         ; Print status
RETURN                                            ; Return to Main Program
END                                               ; End of Entire Program.
```

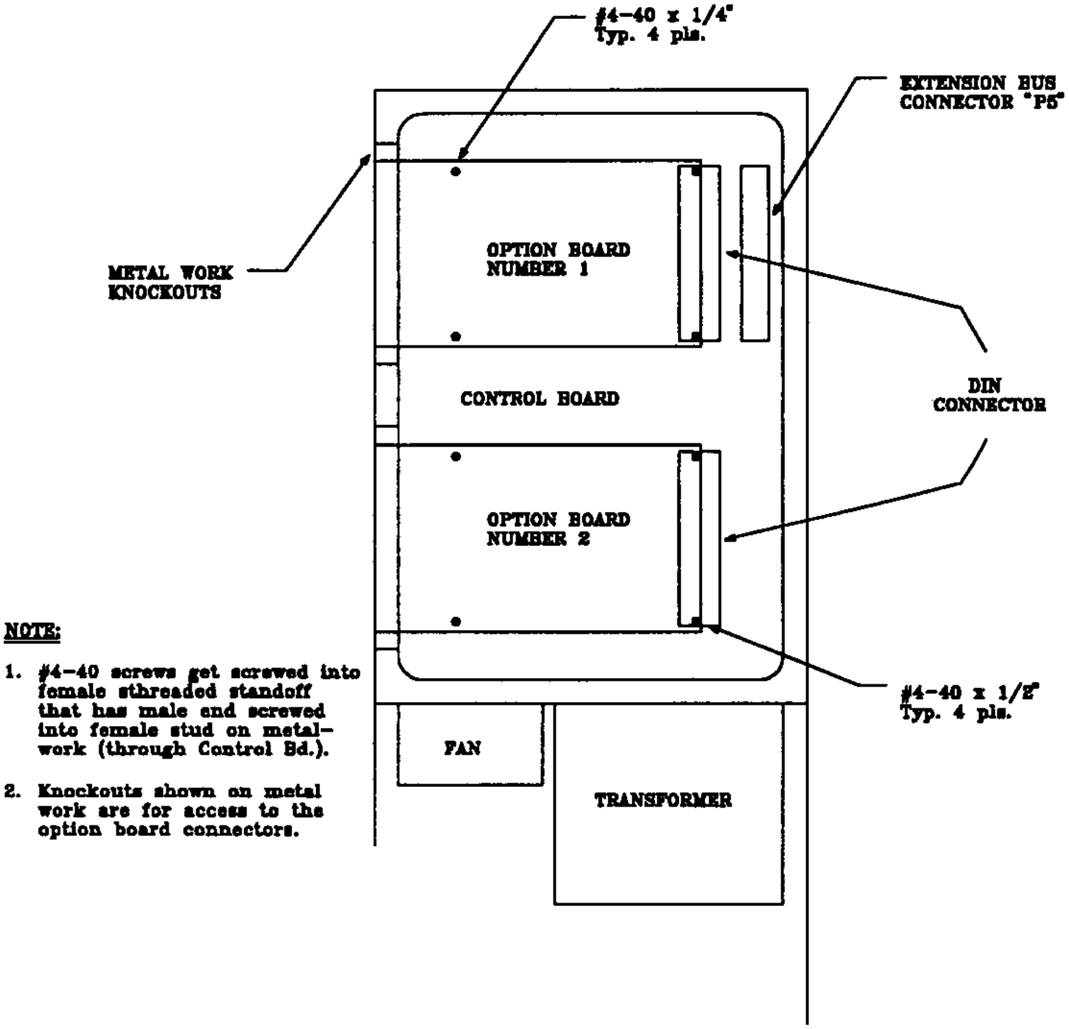
This Page Intentionally Left Blank.

CHAPTER 4: HARDWARE

This chapter contains information that concerns the IEEE board and related hardware. It provides information on both electrical and mechanical features.

4.1: UNIDEX 100 IEEE OPTION LOCATION

The Unidex 100 IE-488 Interface Board mounts to the UNIDEX 100 Control Board with four #4-40 x 3/4" standoffs. These standoffs get screwed into the female studs located behind the mounting holes on the Control Board. There are two #4-40 x 1/4" screws that go through the mounting holes at the front of the IEEE Interface Board. Then, the screws get screwed into the remaining two standoffs. For an example on how to mount the IEEE Interface Board to the UNIDEX 100 Control Board, see ***Figure 4-1*** on the following page.



NOTE:

1. #4-40 screws get screwed into female sthreaded standoff that has male end screwed into female stud on metal-work (through Control Bd.).
2. Knockouts shown on metal work are for access to the option board connectors.

Figure 4-1: Mounting of IEEE Interface Board to the UNIDEX 100 Control Board

NOTE: This option can be field installed, however it requires option wiring (Consult Factory). The UNIDEX 100 Control Board can mount 2 option boards.

4.2: IE-488 INTERFACE BOARD DESCRIPTION

The IE-488 Interface Board is a 3U size board that provides the UNIDEX 100 with IEEE capability. ***Figure 4-2*** shows the IEEE Interface Board.

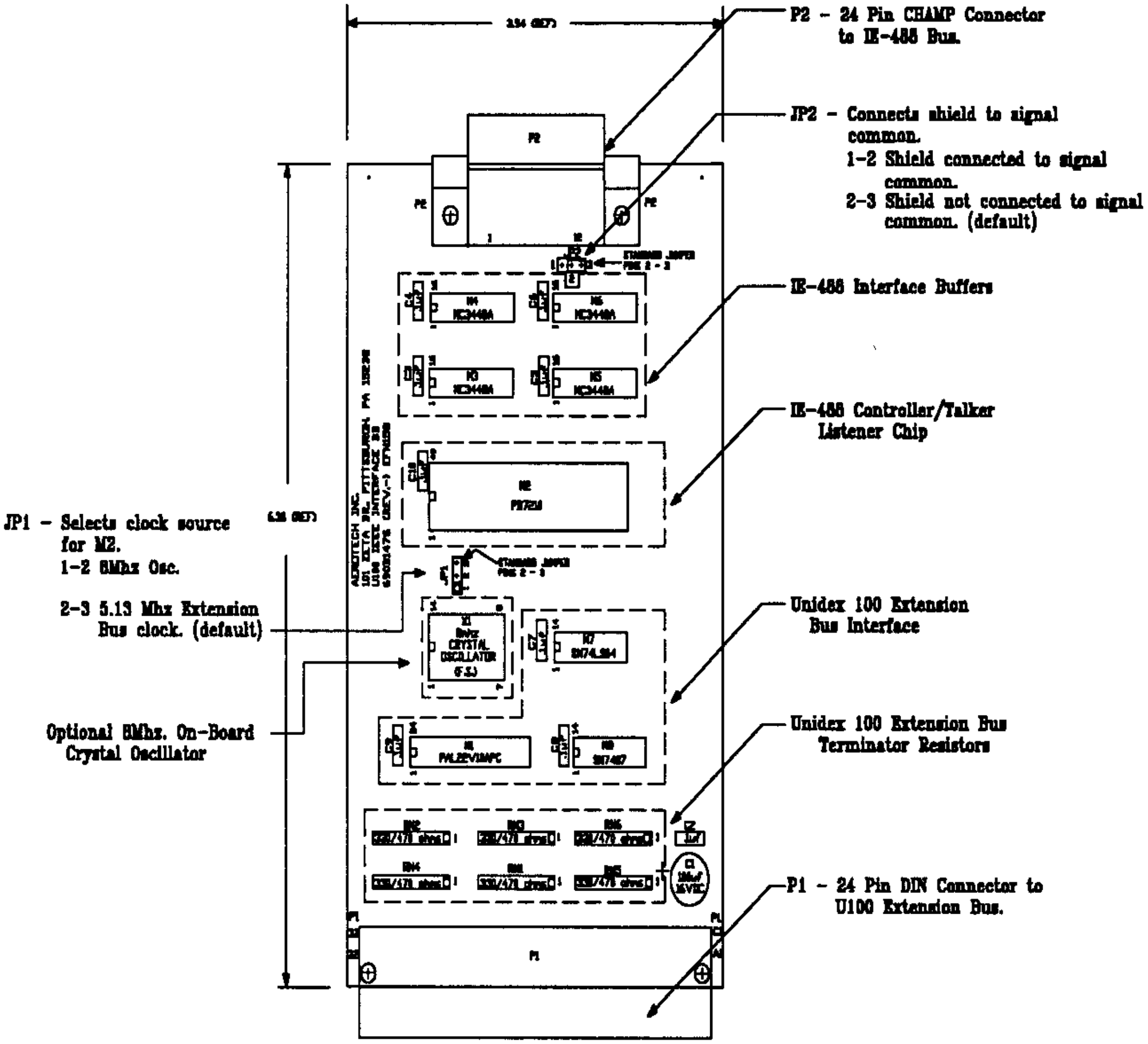

P2 - 24 Pin CHAMP Connector to IE-488 Bus.
JP2 - Connects shield to signal common.
1-2 Shield connected to signal common.
2-3 Shield not connected to signal common. (default)
IE-488 Interface Buffers
IE-488 Controller/Talker Listener Chip
Unidex 100 Extension Bus Interface
Unidex 100 Extension Bus Terminator Resistors
P1 - 24 Pin DIN Connector to U100 Extension Bus.
JP1 - Selects clock source for M2.
1-2 8Mhz Osc.
2-3 5.13 Mhz Extension Bus clock. (default)
Optional 8Mhz. On-Board Crystal Oscillator
P2
STANDARD JUMPER PINS 2 - 3
MC3448A
M2
P8726
JP1
X1
8Mhz CRYSTAL OSCILLATOR (F.S.)
M7
SN74LS04
PAL22V10APC
SN7407
330/470 ohms
C1
P1
AEROTECH INC.
101 ZETA DR. PITTSBURGH, PA 15238
U100 IEEE INTERFACE BD
690D1476 (REV.-) ETCHING

Figure 4-2: IE-488 Interface Board

4.3: IE-488 INTERFACE BOARD JUMPERS

There are two jumpers on the IEEE Interface Board. The first jumper is JP1 and selects the clock for the IEEE board. This jumper normally selects a 5.12 MHz. external clock (JP1, 2-3). To select the on board clock source of 8 MHz set JP1 to 1-2.

The second jumper is JP2. This jumper connects the IEEE connector shield to common. In normal use, there is no connection between the shield and common (JP2, 2-3). To connect the shield to common, place JP2 to 1-2.

The following are recommended Jumper Positions:

JP1 - 2-3 (select external 5.12 MHz clock)

JP2 - 2-3 (shield not connected to common)

4.4: IE-488 INTERFACE BOARD POWER REQUIREMENTS

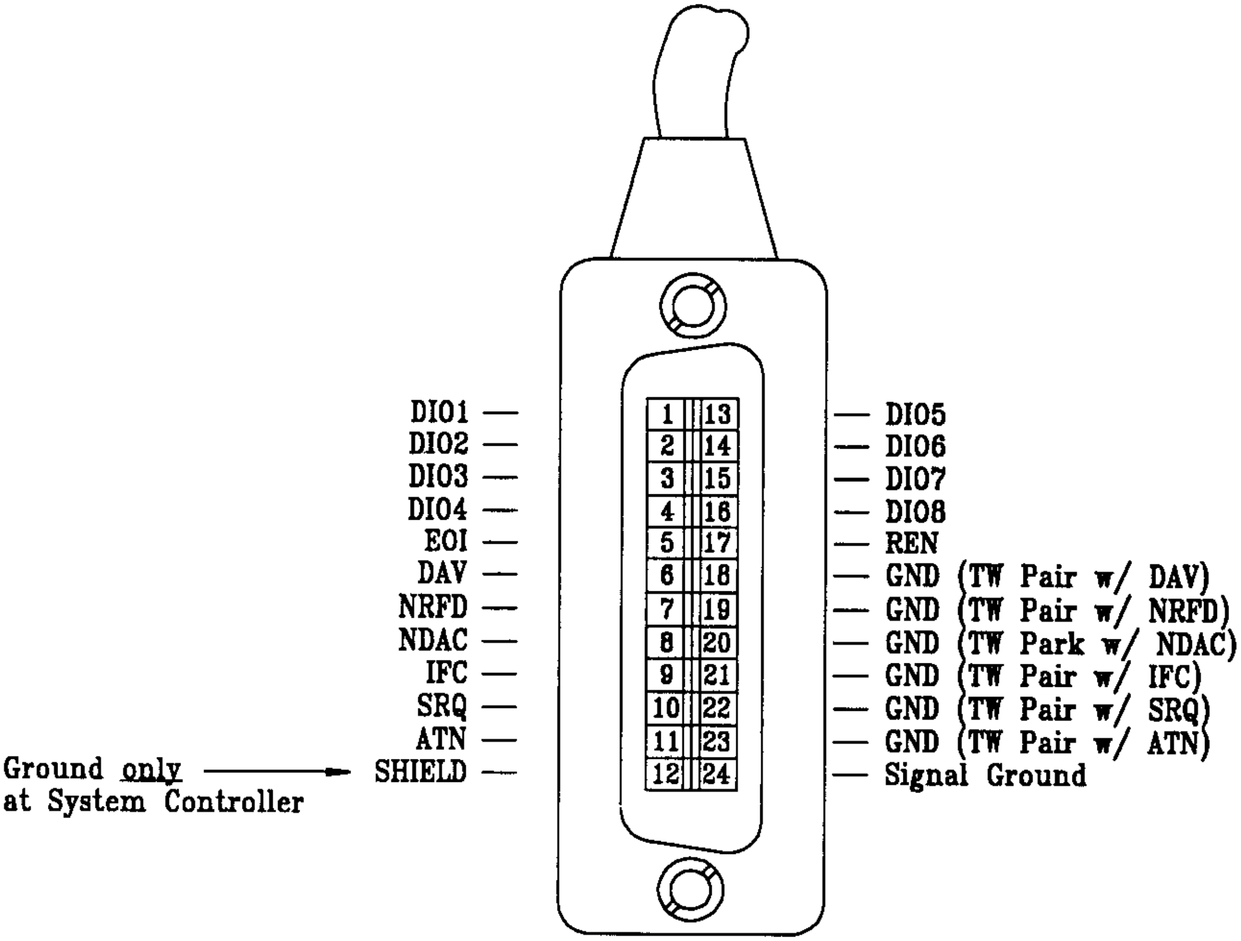
The IE-488 Interface Board requires +5 volts, all of which the Extension Bus connector P1 provides. By no means should this voltage be greater than 5.25 volts or less than 4.75. Voltages not falling within the operating range may cause damage or communication failures.

NOTE: The Extension Bus connector supplies ±12v. The IEEE, however, does not use these supplies.

NOTE: The IEEE Interface Board uses all 5v. logic level.

4.5: IE-488 CONNECTOR

The IE-488 connector is a standard 24 pin IE-488 connector. The total cable length may not exceed 20 meters. It is necessary to use a standard IE-488 cable. Examples of these cables appear in Appendix A of this manual. Below, ***Figure 4-3*** shows the IE-488 connector. Descriptions of the pins shown in this figure follow.



UNIDEX 100 4-3: IE-488 Pin Connections

PIN # DESCRIPTION

1 DIO1 - Data line #1.

2 DIO2 - Data line #2.

3 DIO3 - Data line #3.

4 DIO4 - Data line #4.

5 EOI (End of Identify) - This control line indicates the last data byte and will Parallel Poll devices using the ATN line.

6 DAV (Data Valid) - This control line, used by the talker device, will indicate to the listener devices that the data is valid.

7 NRFD (Not Ready for Data) - This control line indicates that one or more devices are not ready for data.

8 NDAC (Not Data Accepted) - This control line indicates that one or more devices have not accepted the data.

9 IFC (Interface Clear) - The system controller activates this control line and place all devices in the unaddressed state. This line will also cause the system controller to be the active device.

10 SRQ (Service Request) - This control line indicates that one or more devices require attention. Following activation of this line the controller performs a poll of the devices to determine which one is requesting service.

A Serial Poll will clear the SRQ.

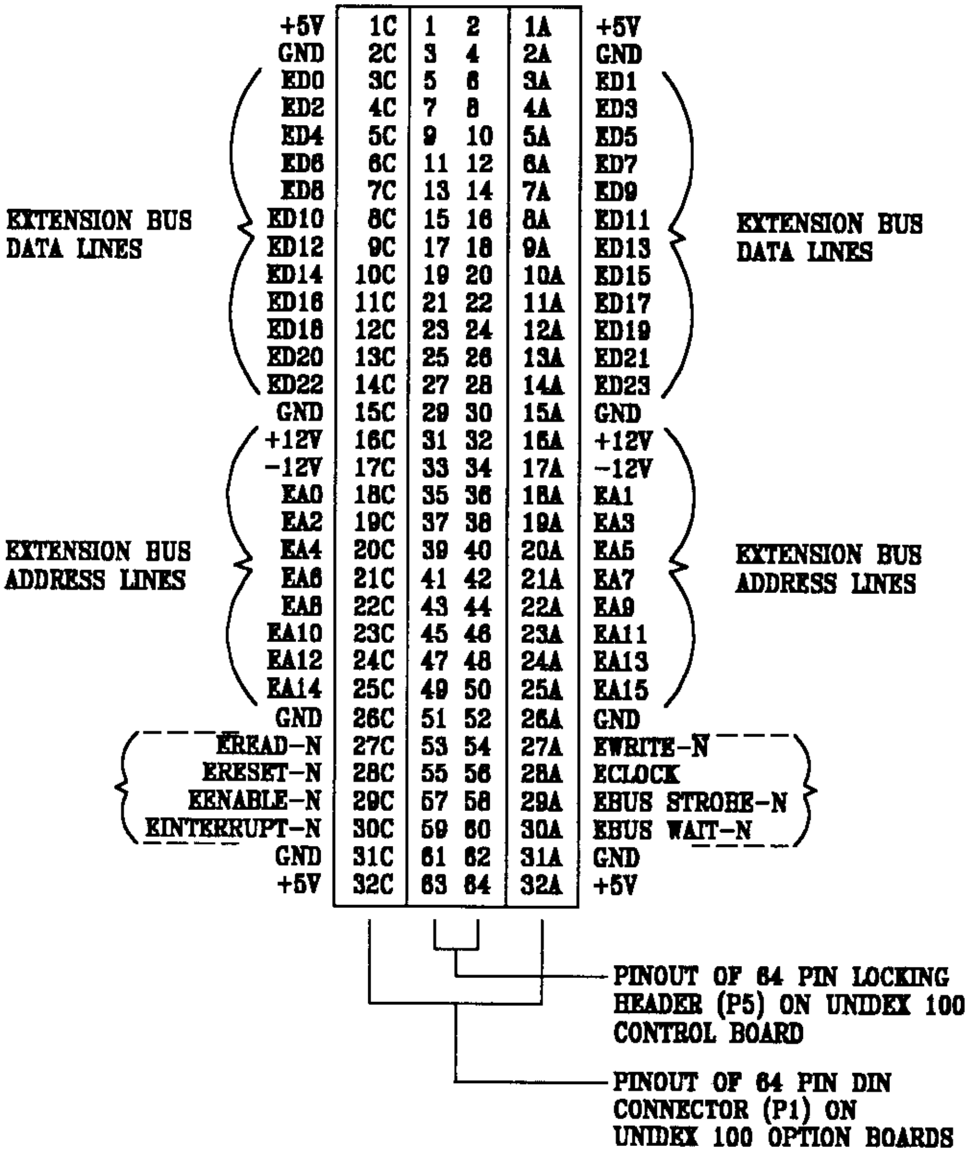
11 ATN (Attention) - This control line when asserted true sends bus interface messages on the data bus. This line, asserted with EOI, will do a parallel poll. When ATN is false, it is possible to send data over the bus by a designated talker.

12 Shield - This pin is the cable shield that normally connects to ground at the system controller. The U100 IE-488 board contains a shield jumper JP2 (1-2 is grounded and 2-3 is unconnected).

PIN#	DESCRIPTION
13	DIO5 - Data line #5.
14	DIO6 - Data line #6.
15	DIO7 - Data line #7.
16	DIO8 - Data line #8.
17	REN (Remote Enable) - This control line will place any addressed listening device into the remote mode, upon asserting this line.
18	GND (Ground) - Typically twisted wire pair with DAV line.
19	GND (Ground) - Typically twisted wire pair with NRFD line.
20	GND (Ground) - Typically twisted wire pair with NDAC line.
21	GND (Ground) - Typically twisted wire pair with IFC line.
22	GND (Ground) - Typically twisted wire pair with SRQ line.
23	GND (Ground) - Typically twisted wire pair with ATN line.
24	Signal Ground - This is signal common.

4.6: IEEE EXTENSION BUS CONNECTOR (P1)

The IEEE Extension Bus connector (P1) serves two purposes. It supplies power to the IEEE Interface board and transfers data between the Control Board and the IEEE Interface board. Followed by ***Figure 4-4*** below is a description of all the Extension Bus pins.



+5V	1C	1	2	1A	+5V
GND	2C	3	4	2A	GND
ED0	3C	5	6	3A	ED1
ED2	4C	7	8	4A	ED3
ED4	5C	9	10	5A	ED5
ED6	6C	11	12	6A	ED7
ED8	7C	13	14	7A	ED9
ED10	8C	15	16	8A	ED11
ED12	9C	17	18	9A	ED13
ED14	10C	19	20	10A	ED15
ED16	11C	21	22	11A	ED17
ED18	12C	23	24	12A	ED19
ED20	13C	25	26	13A	ED21
ED22	14C	27	28	14A	ED23
GND	15C	29	30	15A	GND
+12V	16C	31	32	16A	+12V
−12V	17C	33	34	17A	−12V
EA0	18C	35	36	18A	EA1
EA2	19C	37	38	19A	EA3
EA4	20C	39	40	20A	EA5
EA6	21C	41	42	21A	EA7
EA8	22C	43	44	22A	EA9
EA10	23C	45	46	23A	EA11
EA12	24C	47	48	24A	EA13
EA14	25C	49	50	25A	EA15
GND	26C	51	52	26A	GND
EREAD-N	27C	53	54	27A	EWRITE-N
ERESET-N	28C	55	56	28A	ECLOCK
EENABLE-N	29C	57	58	29A	EBUS STROBE-N
EINTERRUPT-N	30C	59	60	30A	EBUS WAIT-N
GND	31C	61	62	31A	GND
+5V	32C	63	64	32A	+5V

EXTENSION BUS DATA LINES
EXTENSION BUS DATA LINES
EXTENSION BUS ADDRESS LINES
EXTENSION BUS ADDRESS LINES
PINOUT OF 64 PIN LOCKING HEADER (P5) ON UNIDEX 100 CONTROL BOARD
PINOUT OF 64 PIN DIN CONNECTOR (P1) ON UNIDEX 100 OPTION BOARDS

Figure 4-4: Extension Bus Pinout

IEEE (P1)	U100 (P5)	DESCRIPTION
1A	2	+5V
2A	4	Ground
3A	6	ED1 - Extension Bus Data line #1.
4A	8	ED3 - Extension Bus Data line #3.
5A	10	ED5 - Extension Bus Data line #5.
6A	12	ED7 - Extension Bus Data line #7.
7A	14	ED9 - Extension Bus Data line #9.
8A	16	ED11 - Extension Bus Data line #11.
9A	18	ED13 - Extension Bus Data line #13.
10A	20	ED15 - Extension Bus Data line #15.
11A	22	ED17 - Extension Bus Data line #17.
12A	24	ED19 - Extension Bus Data line #19.
13A	26	ED21 - Extension Bus Data line #21.
14A	28	ED23 - Extension Bus Data line #23.
15A	30	Ground
16A	32	+12V
17A	34	-12V
18A	36	EA1 - Extension Bus Address line #1.
19A	38	EA3 - Extension Bus Address line #3.
20A	40	EA5 - Extension Bus Address line #5.
21A	42	EA7 - Extension Bus Address line #7.

IEEE (P1)	U100 (P5)	DESCRIPTION
22A	44	EA9 - Extension Bus Address line #9.
23A	46	EA11 - Extension Bus Address line #11.
24A	48	EA13 - Extension Bus Address line #13.
25A	50	EA15 - Extension Bus Address line #15.
26A	52	Ground
27A	54	EWRITE-N - Extension Bus Write line.
28A	56	ECLOCK - Extension Bus Clock line.
29A	58	EBUS STROBE-N - Extension Bus Strobe line.
30A	60	EBUS WAIT-N - Extension Bus Wait line.
31A	62	Ground
32A	64	+5V
1C	1	+5V
2C	3	Ground
3C	5	ED0 - Extension Bus Data line #0.
4C	7	ED2 - Extension Bus Data line #2.
5C	9	ED4 - Extension Bus Data line #4.
6C	11	ED6 - Extension Bus Data line #6.
7C	13	ED8 - Extension Bus Data line #8.
8C	15	ED10 - Extension Bus Data line #10.
9C	17	ED12 - Extension Bus Data line #12.
10C	19	ED14 - Extension Bus Data line #14.

IEEE (P1)	U100 (P5)	DESCRIPTION
11C	21	ED16 - Extension Bus Data line #16.
12C	23	ED18 - Extension Bus Data line #18.
13C	25	ED20 - Extension Bus Data line #20.
14C	27	ED22 - Extension Bus Data line #22.
15C	29	Ground
16C	31	+12V
17C	33	-12V
18C	35	EA0 - Extension Bus Address line #0.
19C	37	EA2 - Extension Bus Address line #2.
20C	39	EA4 - Extension Bus Address line #4.
21C	41	EA6 - Extension Bus Address line #6.
22C	43	EA8 - Extension Bus Address line #8.
23C	45	EA10 - Extension Bus Address line #10.
24C	49	EA12 - Extension Bus Address line #12.
25C	49	EA14 - Extension Bus Address line #14.
26C	51	Ground
27C	53	EREAD-N - Extension Bus Read line.
28C	55	ERESET-N - Extension Bus Reset line.
29C	57	EENABLE-N - Extension Bus Enable line.
30C	59	EINTERRUPT-N - Ext. Bus Interrupt line.
31C	61	Ground
32C	63	+5V

This Page Intentionally Left Blank.

CHAPTER 5: TROUBLESHOOTING

This section provides troubleshooting information concerning the IEEE Interface. Communication problems with the IEEE may be related to hardware, the communications program, or the incorrect setup of a UNIDEX 100 parameter. Following is a list of problems and possible causes.

SYMPTOM: Cannot establish communications

POSSIBLE CAUSE: Check IEEE cable

Verify that the U100 is in the IEEE mode (see Initial IEEE-488 Power up in Section 2.3).

Check communications program.

Try another Host command such as an Immediate mode command (motion command). Test command in RS-232 mode to verify that the command is valid.

Verify that the U100 is not waiting for a Serial Poll to clear a Service Request.

SYMPTOM: UNIDEX 100 locks up

POSSIBLE CAUSE: UNIDEX 100 may be in a Service Request and waiting for a Serial Poll.

Incomplete or incorrect command sent to the UNIDEX 100. (e.g., a carriage return used with a line feed may cause a lock up.)

SYMPTOM: Will not transfer files

POSSIBLE CAUSE: Service Request must be serviced before file is transferred.

Files do not exist

Incomplete or incorrect command sent to the UNIDEX 100. (e.g., a carriage return used with a line feed may cause a problem.)

SYMPTOM: Some commands do not execute

POSSIBLE CAUSE: Task is busy. Check status of UNIDEX 100.

Incomplete or incorrect command sent to the UNIDEX 100. (e.g., a carriage return used with a line feed may cause the command to not be recognized.)

Command error. Check UNIDEX 100 error status.

SYMPTOM: UNIDEX 100 will not transmit

POSSIBLE CAUSE: Check UNIDEX 100 parameters PRM:008 and PRM:009.

Incomplete or incorrect command sent to the UNIDEX 100.

Waiting for Service Request to be serviced.

APPENDIX A: IE-488 CABLE LIST

The following recommended cables exist for the IE-488. The vendors who supply these include: Hewlett-Packard and Belden Corporation.

Hewlett-Packard

PN	Length
HP 10833D	.5 Meter
HP 10833A	1 Meter
HP 10833B	2 Meter
HP 10833C	4 Meter
HP 10834A	Adapter

Belden Corporation

PN	Length
9642	1 Meter
9643	2 Meter
9644	4 Meter
9645	8 Meter
9646	16 Meter

NOTE: This cable list is for reference only and does not imply that these are the only cables that will work with the IEEE.

This Page Intentionally Left Blank.

APPENDIX B: IEEE HOST MODE OPERATION

It is necessary for the user to use the Host Mode when operating the UNIDEX 100 in the IE-488 Mode. The intent of this appendix is to list the specifications for the "HOST" command set.

The Host Mode operates in both RS-232-C and IE-488 communications mode. IE-488 automatically selects the Host Mode. For RS-232 Host Mode operation the *Unidex 100 User's Manual* contains a more detailed Host Mode description.

The following pages describe the IEEE Host Command Set in detail.

NOTE: The keyword <LF> is a line feed character.

NOTE: Possible errors may occur when executing these commands. If this happens, a Service Request number 0x3 gets sent to the Host (applies to both IE-488 and RS-232-C interface modes). The UNIDEX 100 sends the Host Error Code following the SRQ code.

To determine the specific error code, refer to the *Unidex 100 User's Manual*.

All Service Requests must be acknowledged by performing a serial poll to get the Service Request code. The user must also remove the SRQ bit by subtracting 64 (0x40) to obtain the true SRQ code. If the result is 3, the user must read up to 3 additional bytes to obtain the error code.

Running a "PGM:xx" File

Use the following syntax for running programs while in the IEEE Host Mode.

#B<task number><run mode><program number><LF>

where:

<task number> = "A" for Task 1

= "B" for Task 2

<run mode> = "A" for Auto Mode

= "B" for Block Mode

NOTE: The Block Mode does not implement for host interface in Revision 1.0 Software. Use the immediate command mode shown on the following page.

<program number> = number of the desired program (in memory).

EXAMPLE:

#BAA1<LF>	Run program PGM1 in Task 1 while in the Auto Mode.

NOTE: If an error occurs a SRQ occurs after the command is sent. To allow a Service Request to immediately follow execution, set PRM:023 and PRM:024.

Sending an Immediate Command

Use the syntax below to send an immediate command while in the IEEE Host Mode.

#C<task number><command string><LF>

where:

<task number>	= "A" for Task 1
	= "B" for Task 2
<command string>	= ASCII string containing the specified command. Appendix C lists the available U100 command set. See the *Unidex 100 User's Manual* for additional information on these commands.

NOTE: However, certain U100 commands are not applicable for immediate commands (e.g., GOTO:1).

EXAMPLE:

#CBD(100)<LF>	Execute the command D(100) in Task 2. Note that when executing an immediate command for motion (e.g., D( ), V( ), A( ), or T( )) that the proceeding GO statement is inferred.
#CAFV:1=1.5<LF>	Execute the command FV:1=1.5 in Task 1.

NOTE: If an error occurs a SRQ will be sent after the command is sent. To allow a Service Request to immediately follow execution, set PRM:023 and PRM:024.

Making a Copy of a File

Use the following syntax to make a copy of a file while in the IEEE Host Mode.

#FB<file type><file number 1><LF><file number 2><LF>

where:

<file type> = "A" for "PGM" file
= "B" for "DEF" file
= "C" for "MAC" file
= "D" for "CAM" file
= "E" for "LST" file
= "F" for "DIR" file

<file number 1> = number of selected file to copy.

<file number 2> = number of file to be copied to.

EXAMPLE:

#FBA1<LF>2<LF>	Copy file PGM1 to PGM2.

NOTE: A SRQ will be sent if an error occurs.

Erasing a File

Use the following syntax to erase a file while in the IEEE Host Mode.

#FAA<file type><file number><LF>

where:

<file type> = "A" for "PGM" file

= "B" for "DEF" file

= "C" for "MAC" file

= "D" for "CAM" file

= "E" for "LST" file

= "F" for "DIR" file

<file number> = number of selected file to erase.

EXAMPLES:

#FAAB3<LF>	Erase the DEF3 file from program memory.

#FAB<LF>	Erase all files from program memory.

NOTE: A SRQ will be sent if an error occurs.

Generating a File Directory

While in the IEEE Host Mode, use the following syntax to generate a directory of files that are currently in program memory (e.g., PGM:xx, DEF:xx, etc.).

#AB<LF>

EXAMPLE:

#AB<LF>	Generate a file directory.

NOTE: The UNIDEX 100 will send an IEEE Service Request if an error occurs.

Transfer a File from the Host to the UNIDEX 100

The following syntax is required to transfer a file from the Host to the UNIDEX 100 while in the IEEE Host Mode.

#FDA<file type><file number><LF>

where:

<file type>	= "A" for "PGM" file
	= "B" for "DEF" file
	= "C" for "MAC" file
	= "D" for "CAM" file
	= "E" for "LST" file
	= "F" for "DIR" file
<file number>	= number of selected file to transfer.

EXAMPLE:

#FDAC4<LF>	Transfer the MAC4 file from the Host to the UNIDEX 100's program memory.

NOTE: The UNIDEX 100 will send an IEEE Service Request (must be acknowledged) before transferring the program.

Transfer a File from the UNIDEX 100 to the Host

Use the following syntax to transfer a file from the UNIDEX 100 to the Host while in the IEEE Host Mode.

#FDB<file type><file number><LF>

where:

<file type> = "A" for "PGM" file

= "B" for "DEF" file

= "C" for "MAC" file

= "D" for "CAM" file

= "E" for "LST" file

= "F" for "DIR" file

<file number> = number of selected file type to transfer.

EXAMPLE:

#FDBA2<LF>	Transfer the program 2 file from the UNIDEX 100's program memory to the Host.

NOTE: The UNIDEX 100 sends the SRQ (must be acknowledged) before transferring the program.

Modify the Value of a Parameter

Use the syntax below to modify the value of a specific parameter while in the IEEE Host Mode.

#FCA<parm type><parm number><LF>B<parm value><LF>

where:

<parm type> = "A" for Communication parameters (0xx)

= "B" for Motion parameters (1xx)

= "C" for Drive parameters (2xx)

= "D" for System parameters (3xx)

NOTE: The parameter type is determined by MSD of parameter number (e.g., PRM:202 is drive parameter "C").

<parm number> = number of PRM:xx selected for retrieval. Parm number is the 2 LSD's of the parameter number (e.g., the parm number for PRM:202 is 2).

<parm value> = value to be passed to the specified parameter. This value can be either a floating point or an integer type depending on the specific parameter selected.

EXAMPLE:

#FCAB39<LF>B3<LF>	Modify parameter #139 to the value 3.

NOTE: A SRQ will be sent if an error occurs.

Retrieve the Value of a Parameter

Use the following syntax to retrieve the value of a specific parameter while in the IEEE Host Mode.

#FCA<parm type><parm number><LF>A

where:

<parm type> = "A" for Communication parameters (0xx)

= "B" for Motion parameters (1xx)

= "C" for Drive parameters (2xx)

= "D" for System parameters (3xx)

NOTE: Parameter type is determined by MSD of parameter number (e.g., PRM:202 is drive parameter "C").

<parm number> = number of PRM:xx selected for retrieval. Parameter number is the 2 LSD's of the parameter number (e.g., the parm number to use for PRM:202 is 2).

EXAMPLE:

#FCAA33<LF>A	Retrieve the value of parameter PRM:033.

NOTE: A SRQ will be sent if an error occurs.

Modify the Value of a "Write Only" Register

Use the syntax below to modify the value of a "write only" register while in the IEEE Host Mode.

#FCB<reg type><reg number><LF><reg value><LF>

where:

<reg type> = "A" for Communication registers (0xx)

= "B" for Motion registers (1xx)

= "C" for Drive registers (2xx)

= "D" for System registers (3xx)

NOTE: Register type is determined by MSD of the register number (e.g., REG:308 is the system register "D").

<reg number> = number of REG:xx selected for modification. The reg number is the 2 LSD's of the register number (e.g., the reg number for REG:308 is 8).

<reg value> = value to be passed to the specified write only register. This value can be either an integer or long type depending on the register selected.

EXAMPLE:

#FCBD8<LF>0x3C0<LF>	Modify the value of the write only register REG:308 to the hexadecimal value of 0x3C0.

NOTE: A SRQ will be sent if an error occurs.

Retrieve the Value of a "Read Only" Register

Use the syntax below to retrieve the value of a read only register while in the IEEE Host Mode.

#FCB<reg type><reg number><LF>

where:

<reg type> = "A" for Communication registers (0xx)

= "B" for Motion registers (1xx)

= "C" for Drive registers (2xx)

= "D" for System registers (3xx)

NOTE: The register type is determined by MSD of register number (e.g., REG:015 is communication register "A").

<reg number> = number of REG:xx selected for modification. The reg number is the 2 LSD's of the register number (e.g., the reg number for REG:015 is 15).

EXAMPLE:

#FCBA15<LF>	Retrieve the value of the read only register REG:015.

NOTE: A SRQ will be sent if an error occurs.

Modify the Value of a "Read/Write" Register

Use the following syntax to modify the value of a read/write register while in the IEEE Host Mode.

```
#FCB<reg type><reg number><LF>B<reg value><LF>
```

where:

<reg type> = "A" for Communication registers (0xx)

= "B" for Motion registers (1xx)

= "C" for Drive registers (2xx)

= "D" for System registers (3xx)

NOTE: The register type is determined by MSD of the register number (e.g., REG:202 is drive register "C").

<reg number> = number of REG:xx selected for modification. The register number is the 2 LSD's of the reg number (e.g., the register number to use for REG:202 is 2).

<reg value> = value to be passed to the specified write only register. This value can be either an integer or long type depending on the register selected.

EXAMPLE:

#FCBC2<LF>B0<LF>	Modify the value of a read/write register REG:202 to the value 0.

NOTE: A SRQ will be sent if an error occurs.

Retrieve the Value of a "Read/Write" Register

Use the syntax below to retrieve the value of a read/write register while in the IEEE Host Mode.

#FCB<reg type><reg number><LF>A

where:

<reg type> = "A" for Communication registers (0xx)

= "B" for Motion registers (1xx)

= "C" for Drive registers (2xx)

= "D" for System registers (3xx)

NOTE: The register type is determined by the MSD of the register number (e.g., REG:202 is drive register "C").

<reg number> = number of REG:xx selected for retrieval. The reg number is the 2 LSD's of the register number (e.g., the register number to use for REG:202 is 2).

EXAMPLE:

#FCBC2<LF>A	Retrieve the value of a read/write register REG:202.

NOTE: A SRQ will be sent if an error occurs.

Modify the Value of a Variable

Use the following syntax to modify the value of a variable while in the IEEE Host Mode.

#FCC<var type><var number><LF>B<var value><LF>

where:

<var type> = "A" for Integer variables (BV:)

= "B" for Long variables (LV:)

= "C" for Float variables (FV:)

= "D" for Port variables (PV:)

<var number> = number of VAR:xx selected for modification.

<var value> = value to be passed to the specified variable.

EXAMPLE:

#FCCC1000<LF>B34.8<LF>	Modify the variable FV:1000 to the value of 34.8.

NOTE: A SRQ will be sent if an error occurs.

Retrieve the Value of a Variable

Use the syntax below to retrieve the value of a variable while in the IEEE Host Mode.

#FCC<var type><var number><LF>A

where:

<var type> = "A" for Integer variables (BV:)

= "B" for Long variables (LV:)

= "C" for Float variables (FV:)

= "D" for Port variables (PV:)

<var number> = number of VAR:xx selected for retrieval.

EXAMPLE:

#FCCA33<LF>A	Retrieve the value of the variable BV:33.

NOTE: A SRQ will be sent if an error occurs.

Modify the Value of a String Storage Buffer

Use the following syntax to modify the value of a string buffer while in the IEEE Host Mode.

```
#FCD<string number><LF>B<string data><LF>
```

where:

<string number> = number of SV:xx selected for modification.

<string data> = any sequence of ASCII printable characters up to and including 20 per string buffer.

EXAMPLE:

#FCD5<LF>Bthis is string 5<LF>	Modify the string buffer SV:5 with the following string: "this is string 5"

NOTE: A SRQ will be sent if an error occurs.

Retrieve the Value of a String Storage Buffer

Use the following syntax to retrieve the value of a string buffer while in the IEEE Host Mode.

#FCD<string number>A<LF>A

where:

<string number> = number of SV:xx selected for modification.

<string data> = any sequence of ASCII printable characters up to and including 20 per string buffer.

EXAMPLE:

#FCD1<LF>A	Retrieve the character string that is stored in the string variable SV:1.

NOTE: A SRQ will be sent if an error occurs.

Move Out of a Limit

Use the following syntax to reset a limit condition while in the IEEE Host Mode. Upon executing this command, the axis attempts to move out of the limit (PRM:128 determines the distance to move).

```
#EA
```

EXAMPLE:

#EA	Reset the limit.

NOTE: A SRQ will be sent if an error occurs.

Task Control Functions

These functions allow the user to perform block executes, feedholds, and task quit operations. Use the following syntax to perform these functions.

#EB<function>

where:

<function>	= "A" for Task A Block Execute (for programs running in the Block Mode).
	= "B" for Task A Feedhold
	= "C" for Task A Quit
	= "F" for Task B Block Execute (for programs running in the Block Mode)
	= "G" for Task B Feedhold
	= "H" for Task B Quit

NOTE: The user must be prepared to handle multiple Service Requests (all Service Requests must be acknowledged).

NOTE: The user must send the F1 code to return to the IEEE Host Mode.

Task Control Functions Con't

If a program is running in the Block Mode, a Service Request with a line # will be sent (SRQ must be acknowledged).

EXAMPLE:

#EBH	Quit the Task B operation.

<0x11>	Send the F1 code to return to the Host Mode.

NOTE: The F1 code is a single character with a Hex value of 0x11 or a decimal value of 17.

This Page Intentionally Left Blank.

APPENDIX C: COMMAND SUMMARY

This appendix summaries the UNIDEX 100 Command Set. However, it is important to know that some commands are not applicable for MDI commands.

For additional information on these commands refer to the *Unidex 100 User's Manual.*

COMMAND	DESCRIPTION	SYNTAX
BEGIN	Beginning of a Program	BEGIN
DEF	Definition File where the argument is a constant ranging from 1 to 9999.	DEF<number>
END	Program End	END
ENDMAC	End of the Macro	ENDMAC
MAC	Macro file where the argument is a constant ranging from 1 to 100.	MAC<number>
MAC	Used inside the MAC file, this command defines the macro program.	MAC<macro name>
TITLE	Attach a title to a program	TITLE<description>

Table C-1: Compiler Directives

COMMAND	DESCRIPTION	SYNTAX
CLN	Clear the screen from the specified cursor to the specified column.	CLN(<constant or var>)
CLS	Clear screen	CLS

CUR	Move the cursor to the specified row, and specified column.	CUR(<constant>,<constant>)
		CUR(<BV:number>,<BV:<number>)

GC	Get Character from User Input.	<var>=GC
GM	Get the message from the display and put into variable.	<var>=GM(<constant or var>) SV:<number>=GM(<constant or var>)
PM	Print the message to the screen where the argument is a variable or string.	PM(<var or string>)
PM	Print a message containing two arguments where the first argument is a variable or string, and the second is a string only.	PM(<var or string>,<string>)

Table C-2: Communication Commands

COMMAND	DESCRIPTION	SYNTAX
A	Acceleration/Deceleration Ramp Time in User (units/sec^{2})	A(<var or constant>)
ABSL	Absolute Positioning Mode	ABSL
ADC	Analog to Digital Conversion	<var>=ADC
D	Distance in User Units to move/position	D(<var or constant>)
DAC	Digital to Analog Conversion	DAC(<var or constant>)
DD	Direct Drive	DD(<var or constant>)
DW	Dwell Time (sec); Resolution (1 msec)	DW(<var or constant>)
GO	Begin a Move	GO
HM	Hardware Home	HM
INCR	Incremental Positioning Mode	INCR
T	Ramp Time (sec)	T(<var or constant>)
V	Velocity (user units/sec)	V(<var or constant>)

Table C-3: General Process Motion Commands

COMMAND	DESCRIPTION	SYNTAX
DI	Disable Interrupt	DI
EI	Enable Interrupt	EI
ELSE	Else	ELSE
ELSEIF	Else If	ELSEIF(<var or constant>)op<var or constant>)
ENDIF	End If	ENDIF
ENDSUB	End Subroutine	ENDSUB
ENDWHL	End While Loop	ENDWHL
EXIT	Exit Program Operation	EXIT
GOSUB	Goto subroutine argument where the argument is a constant ranging from 1 to 100.	GOSUB:<number>
GOTO	Goto label argument where the argument is a constant that ranges from 1 to 100.	GOTO:<number>
IF	If/then	IF(<var or constant>op<var or constant>)
LB	Label where the argument is a constant ranging from 1 to 100.	LB:<number>
RI	Reset Interrupt Latch	RI
SRQ	Send Service Request Code for an argument where the argument is a variable or a constant ranging from 1 to 15.	SRQ(<var or constant>)
SUB	Subroutine - defines the beginning of a subroutine.	SUB:<var or constant>
SYNC	Synchronize	SYNC
WHL	While	WHL(<var or constant>op<var or constant>)

Table C-4: Program Control Commands

COMMAND	DESCRIPTION	SYNTAX
+	Addition Operator	<var or constant> + <var or constant>
-	Subtraction Operator	<var or constant> - <var or constant>
*	Multiplication Operator	<var or constant> * <var or constant>
/	Division Operator	<var or constant> / <var or constant>
=	Assignment Operator	<var> = <var or constant>
ABS	Absolute Value function	ABS(<var or constant>)
AND	Logical AND	(<var or constant>AND<var or constant>)
CBI	Convert BCD to integer	CBI(<var or constant>)
CIB	Convert integer or floating point to BCD	CIB(<var or constant>)
COS	Cosine function (in radians)	COS(<var or constant>)
DEC	Decrement	DEC(<var>)
INC	Increment	INC(<var>)
MDX	Modulo Index	MDX(<var>)
OR	Logical OR	(<var or constant>OR<var or constant>)
SIN	Sine function (in radians)	SIN(<var or constant>)
SQRT	Square root	SQRT(<var or constant>)
TAN	Tangent function (in radians)	TAN(<var or constant>)
XOR	Exclusive Or	(<var or constant>XOR<var or constant>)

Table C-5: Math and Logical Operations

This Page Intentionally Left Blank.

INDEX

INDEX