UNIDEX™ 11
MOTION CONTROLLER
OPTIONS MANUAL

PN: EDU104

AEROTECH®

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

Rel. 2.0, August 1989

DISCLAIMER:
The information contained in this manual is subject to change due to improvements in design. Though this document has been checked for inaccuracies, Aerotech does not assume responsibility for any errors contained herein.

TRADEMARKS:
Unidex is a registered trademark of Aerotech, Inc.

IBM PC/XT/AT are registered trademarks of the International Business Machines Corporation.

ORGANIZATION OF DOCUMENTATION FOR UNIDEX 11:
Up to four manuals may have been shipped with your Unidex 11 Controller, depending on the options ordered. Of the four manuals, two supply basic data regarding programming and hardware support information. These manuals are respectively:

- *Unidex 11 Motion Controller Programming Manual*
- *Unidex 11 Motion Controller Hardware Manual*

Depending on the options supplied with your Unidex 11, one or both of the following documents may have also been supplied:

- *Unidex 11 Motion Controller Options Manual* – which is this manual
- *Unidex 11 Interactive Control Software Manual (SSP1)*

Please review, in detail, *Unidex 11 Motion Controller Programming Manual*, before proceeding to this or any other documentation supplied with your Unidex 11.

This manual is divided into six parts, each covering an individual Unidex 11 option. They are:

Part I: IEEE-488 Parallel Interface
Part II: Computer Enabled Joystick
Part III: Joystick Digitizing
Part IV: Digital I/O Port (and Two Digit Thumbwheel option)
Part V: High Speed Binary Interface (and MBI option)
Part VI: Password

Information on Service and Repair, as well as Warranty information, is located at the back of the manual.

PART I:

IEEE-488
PARALLEL INTERFACE

Table of Contents

CHAPTER 1: IEEE-488 INSTALLATION

SECTION 1-1 INTRODUCTION

The IEEE-488 option for the Unidex 11 makes it possible to control Unidex 11 from a host computer through the IEEE-488 bus. A simple command sequence through the IEEE-488 interface gains control of the device. The host computer may, from this point, perform different tasks using Unidex 11. These tasks may include executing a command block in the immediate mode, downloading a program into the user memory, running a program from the memory in auto-run or block-run mode, reading the axis position values, reading the directory, reading a program, or reading the entire memory.

Unidex 11 may also be set up for interactive control. In this mode, Unidex 11 sets the service request after executing a program, a command block, or if there is an error condition. Subsequently, the host computer is required to serial poll Unidex 11 before proceeding.

Before continuing, it is recommended that the user review the *Unidex 11 Motion Controller Programming Manual.*

SECTION 1-2 REQUIRED HARDWARE

A. IEEE-488 INTERFACE

IEEE-488 has 8 data lines and 8 control lines. It can accommodate up to 14 devices and provides a service request line from all devices to the controller. All of these properties lead to a more rapid form of communication between Unidex 11 and the controller. You need not concern yourself with bus disciplines if your controller has IEEE-488 interface and device driver software that "hooks" into the Basic, Fortran, Pascal or other language you intend to use.

Different ways of connecting multiple devices to the IEEE-488 interface is shown in figure 1-2.

The devices connected to the bus have certain roles assigned to them. The roles represent the three basic functional elements necessary for effective communication. These three roles are:

1. Listener
2. Talker
3. Controller

1. Device As Listener

A "listener" is a device that has the capability of receiving data from the bus. It can be addressed by an interface message to listen. When addressed to listen, the listener will receive data placed on the bus.

2. Device As Talker

A "talker" is a device with the capability of sending data through the bus when addressed by an interface message to talk.

3. Device As Controller

A "controller" is a device with the capability of controlling and directing the activity on the bus. A controller can address other devices to listen or to talk. It can also send interface messages to command specific actions from the other devices connected to the bus. You will need a device to act as a controller when implementing the IEEE-488 interface.

Listener, talker and controller capabilities can occur individually or in combinations. For instance, devices such as the Unidex 11 or a terminal can be implemented to talk or to listen, but not to control. Many computers, however, are capable of talking, listening and controlling.

4. *Signal Lines of the IEEE-488 Bus*

The IEEE-488 transfers data and commands between devices on 16 signal wires.

Eight of the lines are for the transfer of data (DI01 to DI08).

Data and message transfers are asynchronous and are coordinated by the three handshake lines.

The remaining five lines, for example "ATN" (attention) and "SRQ" (service request), are for bus management. Each line, when asserted low (ground), represents a single line message sent on the bus. A description of these lines is given in the following subsection (*IEEE-488 Standard Interface Bus Signal Lines*).

5. *Cable Restrictions of the IEEE-488 Bus*

The devices in a system are connected together by a 24-wire cable using 24-pin connectors as specified in the IEEE-488 standard.

There are certain limitations on the length of the cables and the number of devices on the bus.

The maximum number of devices on the bus is limited to 14. The total length of the cable is limited to 20 meters or 2 meters multiplied by the number of devices (whichever is shorter in length). For a complete cable listing, refer to table 1-1.

6. *Parallel And Serial Polling*

Parallel polling is done to identify which device on the IEEE-488 bus is requesting service (SRQ). Serial polling is then done on the device requesting service in order to determine why.

a. Parallel Polling

When parallel polled by the controller, a device asserts a preassigned data line if it is requesting service.

The parallel poll bit assigned to each Unidex 11 may be selected through the front panel keyboard in the set-up mode, as described in section 2-1 of this part of the manual.

b. Serial Polling

In the serial poll, each of the devices requesting service is polled one at a time. You may serial poll any device at any time, regardless of the number of devices on the line.

A Unidex 11 will request service (set SRQ) at specific times, such as when a program is completely executed. At such a time, further operations will be suspended until Unidex 11 is serial polled by the controller. Upon being polled, the Unidex 11 will transmit its status.

B. IEEE-488 STANDARD INTERFACE BUS SIGNAL LINES

MNEMONIC	NAME	FUNCTION
		Bus Management Lines
IFC	Interface Clear	System controller alone can assert this line, to place all devices in the unaddressed state. Devices go into talker idle/listener idle state. If control has been passed to another device, system controller again becomes active by asserting IFC.
ATN	Attention	Asserted true by active controller to send bus interface messages on the bus. When ATN is asserted, signals on the data lines are interpreted as messages. ATN asserted with EOI to do a parallel poll. When ATN is false, data may be sent over the bus by a designated talker.
REN	Remote Enable	Asserted to program devices on the bus remotely. Any device addressed to listen while REN is true, is placed in remote mode of operation.
SRQ	Service Request	Asserted by a device to indicate its need for interaction with the controller.
EOI	End Or Identify	When asserted, indicates the termination of flow of data. Asserted when the last data byte is placed on the bus.

Data Handshake Lines

DAV	Data Valid	Asserted by the talker to indicate to all listeners that data on the bus is valid.
NRFD	Not Ready For Data	When true, indicates to talker that all listeners are not ready for data.
NDAC	Not Data Accepted	When true, indicates to the talker that all listeners have not accepted the data placed on the bus

Data Lines

DI01 - DI08	Data Lines	Used for sending data (ATN lines false) or bus interface messages (ATN line true).

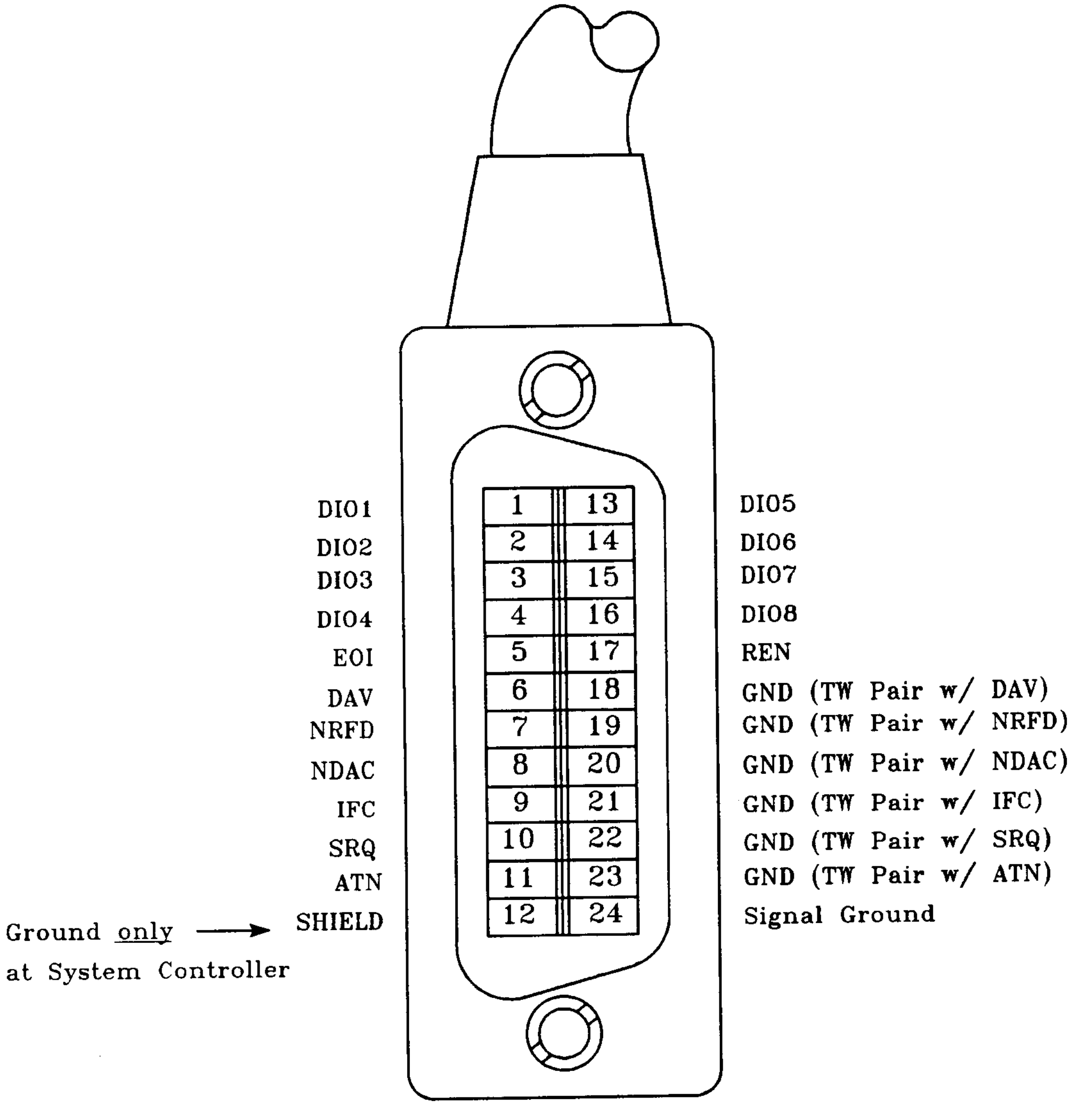


Figure 1-1: IEEE-488 Pin Connections

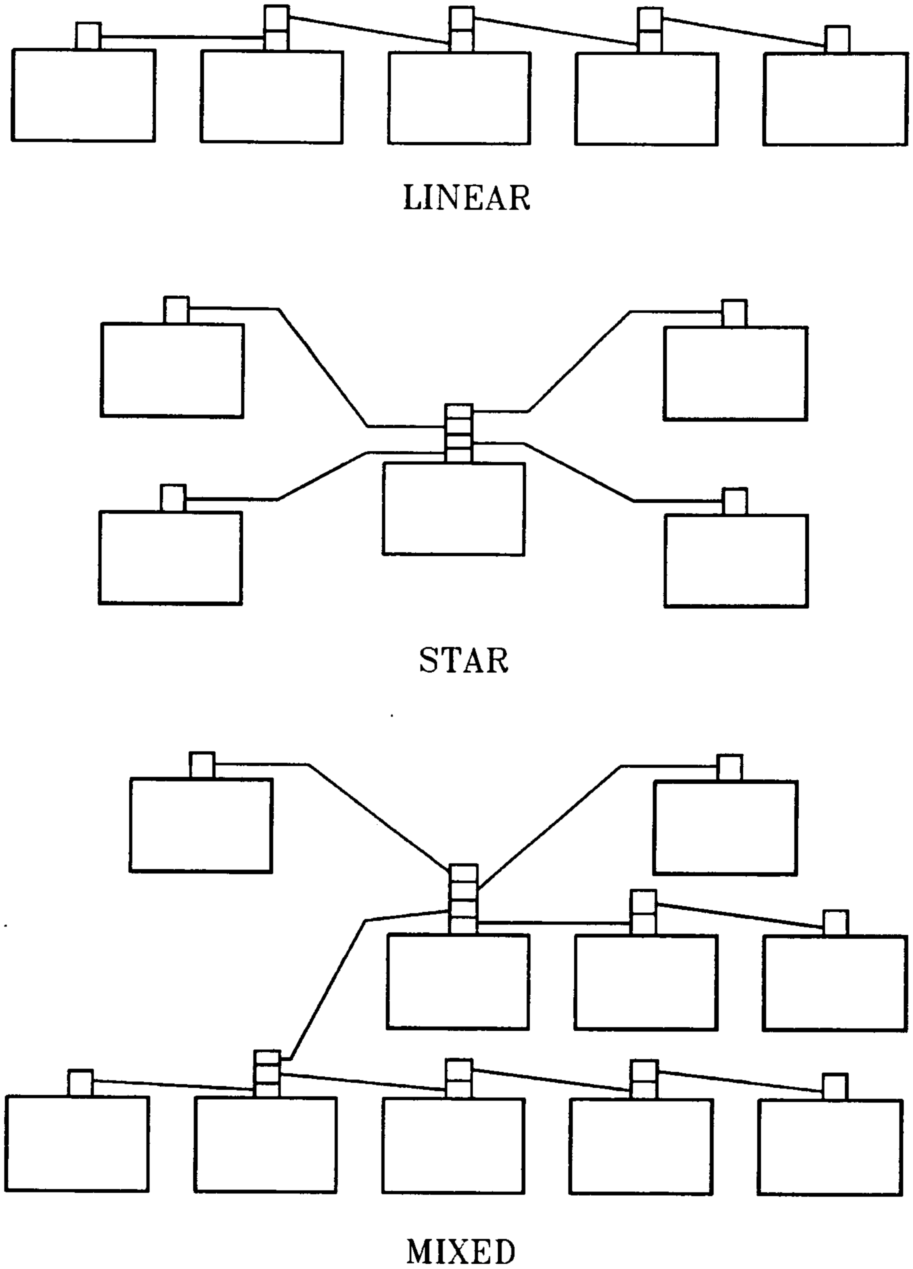


Figure 1-2: Cabling Configurations

HEWLETT-PACKARD
Palo Alto, California 94304

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

BELDEN CORPORATION
Richmond, Indiana 47374

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

Table 1-1: IEEE-488 Cable Manufacturers

CHAPTER 2: IEEE-488 OPERATION OF UNIDEX 11

SECTION 2-1 SETTING UP THE IEEE-488 PARAMETERS

It is required that the device address of Unidex 11 be set to a number that the controller uses to address it prior to any data transfer. Device addresses that can be set vary from "00" to "15". To change the device address of Unidex 11, put the system in the SET UP mode as described in the *Unidex 11 Motion Controller Programming Manual.*

Press SELECT until the set up screens are displayed. They will be displayed after the last axis' "Number of Digits after Point" screen. The display screen will show:

```
Insert Block Nos. when
Printing programs : NO
```

Press SELECT to see:

```
IEEE-488  INTERFACE
Device  Address   : 02
```

Now the +/- key may be pressed repeatedly until the address changes to the required value.

Press select again for the last set up screen related to the IEEE-488 interface.

IEEE-488 INTERFACE

Par. Poll Resp. : PPR1

This screen sets up the identity byte for the device. Each device on the IEEE-488 bus may be assigned one data line that the device asserts when an IDENTIFY command is sent by the controller (when the controller parallel polls the device). The identity bytes and the corresponding PPRx set ups are shown below:

IEEE-488 PARALLEL POLL RESPONSE

(SIGNAL ON THE DATA LINES)

	8	7	6	5	4	3	2	1
PPR1	0	0	0	0	0	0	0	1
PPR2	0	0	0	0	0	0	1	0
PPR3	0	0	0	0	0	1	0	0
PPR4	0	0	0	0	1	0	0	0
PPR5	0	0	0	1	0	0	0	0
PPR6	0	0	1	0	0	0	0	0
PPR7	0	1	0	0	0	0	0	0
PPR8	1	0	0	0	0	0	0	0

Figure 2-1: Parallel Poll Response

CHAPTER 3: IEEE-488 BUS PROGRAMMING

SECTION 3-1 COMMUNICATING WITH UNIDEX 11

Unidex 11 is ready for communication at power up if the device address has been correctly set up and the system has a battery back up or if the default values are the required format. Default values are:

DEVICE ADDRESS : 02
PARALLEL POLL RESPONSE : PPR1

If the keyboard is operated, the communication interface is disabled and Unidex 11 will not respond to commands on the IEEE- 488 bus until the interface is enabled from the keyboard. To enable IEEE-488 interface, bring up the fifth screen of the main menu:

```
1. DIG.  OUT      3. COMM ENAB
2. ACL/DCL        4. PRINT
```

Pressing key #3 will take you to the following screen:

```
1.  RS-232/IEEE-488
2.  PARALLEL PORT
```

Press key #1 and Unidex 11 gets ready to communicate through IEEE- 488.

```
IEEE-488 INTERFACE
COMMUNICATION ENABLED
```

The host device must now address Unidex 11 to listen. Most controllers provide the user with a high level language statement that can be executed in the controller to accomplish this. When addressed to listen, Unidex 11 will display:

```
IEEE-488 INTERFACE
COMMUNICATION ACTIVE
```

SECTION 3-2 TYPES OF COMMANDS

A command sent to Unidex 11 through the IEEE-488 interface may be classified into two types:

1. *System Commands:* These are commands needed to interact with Unidex 11 as a device and perform operations such as resetting Unidex 11, printing a program, printing position values, running a program, downloading a program, etc. Each system command establishes a mode of operation.

2. *Program (Motion) Commands:* These are the user program blocks in a motion program that Unidex 11 executes when running the program or in the immediate mode. Program commands are valid only in the immediate or edit mode.

SECTION 3-3 SERVICE REQUEST AND SERIAL POLL

Service request is an important concept in device control when there is a controller (host computer) as the master and a controlled device (such as a printer or Unidex 11) as the slave. The purpose of Service Request is for the slave device to catch the attention of the master controller.

Typically, a controller has more than one device being controlled by it, and it would be very inefficient for the master controller to continually read the statuses of all the devices to check for error states.

The controlled (slave) device therefore has the capacity to send a signal (service request) to the master controller whenever the attention of the master is required. The reason for requesting service may be an error condition or to signal the completion of a task.

Unidex 11 implements a Service Request by asserting the SRQ line on the IEEE-488 bus. The controller (master) may be set up to be interrupted by a Service Request and to take the necessary action.

The minimum necessary action that the controller must take once Unidex 11 has sent the Service Request signal is to serial poll Unidex 11.

Unidex 11 waits until it is serial polled by the controller, and will not respond to any other system command until this is done. The purpose of the serial poll is to read a status byte from Unidex 11.

The 8 bits of this status byte represent different internal states of Unidex 11. Responding to a serial poll is a hardware function and therefore the controller may serial poll Unidex 11 at any time.

The status byte may be analyzed by the controller to determine the cause of the Service Request. Each of the bits in the status byte is described below.

SERIAL POLL STATUS BYTE

	Zero	**One**
Bit 0	Incremental mode	Absolute mode
Bit 1	Not running a program	Running a program
Bit 2	Block run mode	Auto run mode
Bit 3	Non-corner rounding mode	Corner rounding mode
Bit 4	Communication disabled	Communication enabled
Bit 5	Not executing command	Executing a command
Bit 6	No Service Request signal sent	Service Request signal sent; waiting for serial poll
Bit 7	No errors detected	Error detected

Unidex 11 may be put into the Service Request mode by the system command: **J <CR> <LF>**

The Service Request mode may be cancelled by sending the system command **K <CR> <LF>**. In this mode Unidex 11 will not send a service request signal for any reason. This is the default mode. In this mode, to determine if an immediate command or a program has been completed, a serial poll may be done, and the status byte analyzed (bit 1 and bit 5). When these bits are clear (zero), Unidex 11 is ready to take the next command.

A. CONDITIONS FOR SERVICE REQUEST

When in the service request mode, Unidex 11 sets a service request (SRQ) under the following conditions:

1. When an Immediate command execution is complete.

2. When a program is completely executed in the Auto Run mode.

3. When a block is executed in the Block Run mode.

4. When a run time error condition is generated and the program is aborted.

5. When an axis limit is activated.

6. When a program or immediate command move is stopped by pressing the STOP key on the front panel keypad.

7. At the end of a program download operation, if an error was generated while downloading. (The SRQ is set by Unidex 11 after the "%" that ends the downloading of the program.)

8. When it is requested that a nonexistent program be printed. (If "Pnn <CR> <LF>" is sent to Unidex 11 and program #nn does not exist, Unidex 11 will set the SRQ line.)

NOTE: For more information on error bytes see chapter 4 of this part of the manual.

NOTE: For more information on IEEE-488, refer to the "ANSI/IEEE Std 488-1978" standard, published by:

The Institute of Electrical and Electronics
Engineers, Inc.
345 East 47th Street
New York, NY 10017

CHAPTER 4: ERROR CODES

2
13
8
104

An error condition may be detected by the host computer by checking the most significant bit (bit 7) of the serial poll status byte (section 3-3 of this part of the manual). If this bit is set to "1", an error has occurred.

To further determine the type of error, the system status may be read by the host by sending the command "PS <CR> <LF>". Unidex 11 will send back 13 bytes followed by <CR> <LF>. (Refer to chapter 5 of this part of the manual for details on printing status bytes.) These 13 bytes represent a complete status report of Unidex 11. They are described as follows:

BYTE 1

Same as Serial Poll status byte described in section 3-3 of this part of the manual.

BYTE 2 EDITOR ERROR STATUS

	Zero	One
Bit 0	No illegal character during download	Illegal character during download (illegal command code)
Bit 1	Memory not full during download	Memory full during download

Bit 2	No user memory checksum error	Checksum error during download of program
Bit 3	No illegal command	Illegal command during download (illegal command code)
Bit 4 || Bit 7	- Not used	

NOTE: If one of these errors is generated during download operation, Unidex 11 will assert SRQ (service request), if in the Service Request mode. It is recommended that the user then edit and correct that program.

BYTE 3 RUNTIME ERROR STATUS 1

	Zero	One
Bit 0	X axis not in limit	X axis in limit
Bit 1	Y axis not in limit	Y axis in limit
Bit 2	U axis not in limit	U axis in limit
Bit 3	V axis not in limit	V axis in limit
Bit 4	No illegal byte in memory	Illegal byte in memory
Bit 5	Program number valid	Invalid program called out for run
Bit 6	Memory not clear	No programs in memory (Memory clear)
Bit 7	No user memory checksum error	User memory checksum error

BYTE 4 RUNTIME ERROR STATUS 2

	Zero	One
Bit 0	Stop key not pressed	Stop key pressed to stop motion/program
Bit 1	No "Repeat Loop End Invalid" error	"Repeat Loop End Invalid" error
Bit 2	No "Repeat Loop Incomplete" error	"Repeat Loop Incomplete" error
Bit 3	Eight repeat loops not exceeded	Eight repeat loops exceeded
Bit 4	No "Return From Subroutine Invalid" error	"Return From Subroutine Invalid" error
Bit 5	No "Incomplete subroutine routine" error	"Incomplete subroutine" error
Bit 6	Eight subroutines not exceeded	Eight subroutines exceeded
Bit 7	No "Missing Label" error	"Missing Label" error

BYTE 5 COMMUNICATION STATUS 1

	Zero	One
Bit 0	No RS-232 hardware on OP4 board	RS-232 hardware on OP4 board

Bit 1	No IEEE-488 hardware on OP4 board	IEEE-488 hardware on OP4 board
Bit 2	RS-232 communication not active	RS-232 communication active
Bit 3	IEEE-488 communication not active	IEEE-488 communication active
Bit 4	" > " or "#" not received	" > " or "#" received
Bit 5	Not used	
Bit 6	Not in SRQ mode	In SRQ mode
Bit 7	Not in Hold mode	In Hold mode

BYTE 6 COMMUNICATION STATUS 2

	Zero	One
Bit 0	LCD Display 1 included in system (X and Y display present)	LCD Display 1 not included in system
Bit 1	LCD Display 2 included in system (U and V display present)	LCD Display 2 not included in system
Bit 2	In Hold mode, but no Trigger command received	In Hold mode, and Trigger command received

Bit 3	Unidex 11 "receive buffer" not full	Unidex 11 "receive buffer" full
Bit 4	Xon received during transmit	Xoff received during transmit
Bit 5	Not in program download mode	In program download mode
Bit 6	Status bytes printed in binary format	Status bytes printed in Hex-ASCII format
Bit 7	I/O parallel port input not enabled	I/O parallel port input enabled

BYTE 7 AXIS VALIDITY STATUS

	Zero	One
Bit 0	X axis not in system	X axis in system
Bit 1	Y axis not in system	Y axis in system
Bit 2	U axis not in system	U axis in system
Bit 3	V axis not in system	V axis in system
Bit 4 || Bit 7	Not used	

BYTE 8 RAMPER BOARD STATUS

	Zero	One
Bit 0	X axis does not have ramping	X axis has ramping (through 1MR)
Bit 1	Y axis does not have ramping	Y axis has ramping (through 1MR)
Bit 2	U axis does not have ramping	U axis has ramping (through 1MR)
Bit 3	V axis does not have ramping	V axis has ramping (through 1MR)
Bit 4 || Bit 7	Not used	

BYTE 9 AXIS MOTION STATUS

	Zero	One
Bit 0	X axis not moving	X axis moving
Bit 1	Y axis not moving	Y axis moving
Bit 2	U axis not moving	U axis moving
Bit 3	V axis not moving	V axis moving

Bit 4 || Bit 7	Not used

BYTE 10 FREE RUN MODE STATUS

	Zero	One
Bit 0	X axis not in free run mode	X axis in free run mode
Bit 1	Y axis not in free run mode	Y axis in free run mode
Bit 2	U axis not in free run mode	U axis in free run mode
Bit 3	V axis not in free run mode	V axis in free run mode
Bit 4 || Bit 7	Not used	

BYTE 11 I/O STATUS

	Zero	One
Bit 0	Input 1 is a 0 (I1)	Input 1 is a 1 (I1)
Bit 1	Input 2 is a 0 (I2)	Input 2 is a 1 (I2)
Bit 2	Input 3 is a 0 (I3)	Input 3 is a 1 (I3)

Bit 3	Input 4 is a 0 (I4)	Input 4 is a 1 (I4)
Bit 4	Output 1 is a 0 (O1)	Output 1 is a 1 (O1)
Bit 5	Output 2 is a 0 (O2)	Output 2 is a 1 (O2)
Bit 6	Output 3 is a 0 (O3)	Output 3 is a 1 (O3)
Bit 7	Output 4 is a 0 (O4)	Output 4 is a 1 (O4)

BYTE 12 AND BYTE 13 DIGITAL OUTPUT STATUS

Bit 0 || Bit 11	12 bit value of digital output
Bit 12 || Bit 15	Not used

CHAPTER 5: TYPES OF COMMANDS

SECTION 5-1 SYSTEM COMMANDS

System commands interact with Unidex 11 as a device and perform operations such as resetting it, printing a program, printing position values, running a program, downloading a program, transferring status byte information from Unidex 11, etc. Each system command establishes a mode of operation once it is received by Unidex 11. Each system command must be entered as a capital letter. Transmission to Unidex 11 may be terminated by a <LF>, <CR> <LF>, EOI or <CR> <LF> EOI.

A. GETTING UNIDEX 11'S ATTENTION

To activate the IEEE-488 interface in Unidex 11, it must be addressed to listen. The controller may do this by asserting the ATN and REN line and sending the appropriate Listen Address Code. Generally, the user need not be concerned with the low level bus operations. Most controllers provide means to communicate on the IEEE-488 bus from a high level language through familiar statements such as OUTPUT, PRINT, ENTER, READ, INPUT, etc.

B. AUTO MODE

Executing a program in the auto mode enables the program to run automatically, executing the motion commands of the program with no need of further user intervention.

To run a program in the auto mode, send "A", the program number ("nn") and <CR> <LF>. Example:

```
A 10 <CR> <LF>
```

If in the service request mode (see section 3-3 of this part of the manual), once the program has been executed Unidex 11 will set the service request and wait for a serial poll. After the serial poll, you may execute the same program again by sending another <CR> <LF>. To run a different program, send "A nn <CR> <LF>" again.

C. BLOCK MODE

A motion program can be run one block at a time, instead of automatically as discussed in the above subsection. To run a program in the block mode, send "B" for block, the program number ("nn") and a <CR> <LF>. Example:

```
B 10 <CR> <LF>
```

If in service request mode, Unidex 11 will set SRQ after each block has been executed. If this is the case, the host must serial poll Unidex 11 after the execution of each block.

After the execution of the first command block and the serial poll, send a <CR> <LF> to execute the next block. Bit 1 of the status byte (section 3-3) may be checked to detect completion of the program. This bit is cleared after the last block in the program.

D. REMOTE RESET

Sending the command "C" followed by <CR> <LF> resets Unidex 11 after the previous system command is executed. This will take it back to power up conditions. Example:

```
C <CR> <LF>
```

The IEEE-488 bus functions "Device Clear" (DCL) and "Selected Device Clear" (SDC) also effect the same response from Unidex 11.

E. DISABLING JOYSTICK MODE OR REMOTE MODE

(The system command "D" is available with the Unidex 11 Joystick Option (JP4C) only. See Part II, *Computer Enabled Joystick*, of this manual for details.) The command:

```
D <CR> <LF>
```

will do one of the following:

1. Disable the Computer Enabled Joystick Mode and return control to the host controller. (The position registers will be updated with the absolute position values before returning control.)

2. Disable the Remote Mode and return control to the host. (The position registers are updated with the absolute position values before returning control.)

When either the Computer Enabled Joystick Mode or Remote Mode is active, Unidex 11 will only recognize the "D" command and the Serial Poll command. All other system commands will be ignored.

F. DOWNLOADING A PROGRAM TO UNIDEX FROM HOST

The "E" command, followed by a program number ("nn") and an end-of-block character (* or /), will put Unidex 11 into the edit mode and set it up to enter the program commands into program "nn" in the user memory. If an existing program with the same number already resides in Unidex 11, it will be deleted automatically when the new program "nn" is downloaded. Example:

```
E10 * HXY * XF100D1000YF1000D-2000 * %
```

In the above example, the commands following "E10*" will be downloaded into the Unidex 11 user memory and stored in program #10. The motion commands that may be included in this program will be discussed in section 5-2, *Motion Program Commands*.

G. DELETING A PROGRAM

In order to delete a program from the Unidex 11 user memory, send the command "E", followed by the character "$", the program number "nn" and an end-of-block character, either "*" or "/". Example:

```
E $ 10 / (or *)
```

Program 10 will be erased.

H. DELETING ALL PROGRAMS (USER MEMORY)

In order to delete *all* programs from the Unidex 11 user memory, send the command "E", followed by the character "$", two zeros ("00") and an end-of-block character, either "*" or "/". Example:

```
E $ 00 /
```

All programs will be erased.

I. BLOCK NUMBERING

If you want programs to be printed with block numbers, send the "F" command and a <CR> <LF>. (Block numbering may make editing the program easier.) Example:

```
F <CR> <LF>
```

After this command is sent to Unidex 11, any programs will be printed with block numbers.

J. BLOCK NUMBERING CANCEL

In order to cancel block numbering when a program is printed, as established in the above subsection, send the command "G" along with <CR> <LF>. Example:

```
G <CR> <LF>
```

After this command has been sent to Unidex 11, programs will be printed without block numbering.

These system commands (F and G) do not change the system set up feature stored in the battery backed memory.

K. HOLD

The command to "hold" the execution of a command string or an entire program is established by the "H" command and a CR> <LF>. Example:

```
H <CR> <LF>
```

The above command will cause Unidex 11 to suspend execution of any Immediate, Auto or Block commands which may follow it. This is useful if you wish to synchronize axis motion with some other action. Unidex 11 will only execute the commands when it receives a "T" (for Trigger) command or a "Group Execute Trigger" (GET) bus message (discussed in a following subsection). Example:

```
H <CR> <LF>
A 20 <CR> <LF> Program #20 not executed (held)
T <CR> <LF>    Now Program #20 executes
or
(GET) Bus interface message
```

L. CANCEL HOLD

To cancel the hold command and let Unidex 11 execute Auto, Immediate and Block commands without the trigger command, send:

```
O<CR> <LF>
```

M. IMMEDIATE MODE

The "I" command, followed by motion program commands, an end-of- block character (* or /) and a <CR> <LF>, allows a block of motion commands to be executed immediately instead of being entered as a motion program. Each block of immediate commands must begin with an I. For example:

```
I X F10000 D20000 * <CR> <LF>
```

The above immediate command will send the X axis 20000 steps at a feedrate of 10000 steps per second (or whatever units might be set in the system). If in SRQ mode, Unidex 11 will send a service request and wait for a serial poll after the command is executed. After being polled, Unidex 11 is ready to execute another block of commands.

All motion commands that make up a Unidex 11 motion program are listed in section 5-2 of this chapter. Although all motion commands are valid in the edit mode, not all are valid in the immediate mode. Following is a list of motion commands that are valid in the immediate mode (for full explanation of each, see section 5-2 of this chapter):

AB*
AD*
BF*
BN*
CO*
DB*
DD*
DW*
HX*
IN*
IT*
LX*
NC*
OR*
OS*
OT*
RX*
SX*
X F D*
Y F D*
U F D*
V F D*

N. SERVICE REQUEST SET UP

In order to establish the service request mode, send the "J" command, followed by <CR> <LF>. After the SRQ mode has been established through the J command, Unidex 11 will assert the SRQ line under conditions described in section 3-3. It will then wait until it is serial polled by the controller before executing any further commands. (For more detailed information on SRQ, see section 3-3 of this part of the manual).

O. SERVICE REQUEST CANCEL

In order to cancel the service request (SRQ) mode established by the J command (above subsection), send a K command, followed by <CR> <LF>. Example:

```
K <CR> <LF>
```

SRQ cancelled is the default status.

P. STATUS BYTE IN BINARY FORMAT

To establish the format of the status bytes as binary upon transmission, sent command "M", followed by <CR> <LF>. Example:

```
M <CR> <LF>
```

Transmission of the status bytes in binary format is the default status.

Q. STATUS BYTE IN HEX-ASCII FORMAT

To establish the status bytes in the Hex-ASCII format upon transmission, send the command "N", followed by <CR> <LF>. Example:

```
N <CR> <LF>
```

R. PRINT AXIS POSITION

1. Print X Axis Position (PX)

In order to print the X axis position register, send:

PX <CR> <LF>

When Unidex 11 is addressed to talk, the axis position is sent in the following format:

<Space> or <Negative Sign> <10 Digits> <CR> <LF> EOI

2. Print Y Axis Position (PY)

In order to print the Y axis position register, send:

PY <CR> <LF>
(and address Unidex 11 to talk)

3. Print U Axis Position (PU)

In order to print the U axis position register, send:

PU <CR> <LF>
(and address Unidex 11 to talk)

4. Print V Axis Position (PV)

In order to print the V axis position register, send:

PV <CR><LF>
(and address Unidex 11 to talk)

S. PRINTING DIRECTORY LISTING

To get a listing of the programs in the Unidex 11 directory send:

PD <CR><LF>
(and address Unidex 11 to talk)

Bytes of memory remaining in Unidex 11 will be printed as well. Transmission of data will be terminated by <ETX> EOI.

T. PRINTING A PROGRAM

To have one program printed, send the command "P", the program number ("nn") and <CR><LF>. Example:

P10 <CR><LF>
(and address Unidex 11 to talk)

The above command will cause program #10 to be printed out. Transmission of data will be terminated by <ETX> EOI.

U. PRINTING ALL PROGRAMS

To have all programs printed, send the command "P", two zeros (00) and <CR><LF>. Example:

P00 <CR><LF>
(and address Unidex 11 to talk)

The above command will cause all programs in memory to be printed out. Transmission of data will be terminated by <ETX> EOI.

V. PRINTING STATUS BYTES

To have the status bytes listed in chapter 4 printed out, send:

```
PS <CR> <LF>
(and address Unidex 11 to talk)
```

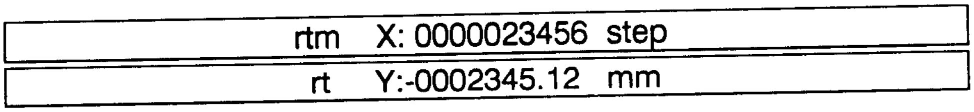
Depending on "M" or "N" command sent earlier, the status will be sent as 13 binary bytes followed by <CR> <LF> EOI or 13 sets of 3 bytes (2 hex-ASCII + space) ended by <CR> <LF> EOI.

W. REMOTE MODE

The system command:

```
R <CR> <LF>
```

will enable Unidex 11 with the JP4C option to be driven through the auxiliary controls. The display shows the remote mode tracking screen:



```
rtm   X: 0000023456  step
rt    Y:-0002345.12  mm
```

The host controller may now signal an external device to take control of Unidex 11.

Unidex 11 keeps track of the axes' positions during external control.

X. ENABLING JOYSTICK

(The system command "S" is available with the Unidex 11 Joystick (JP4C) Option only. See Part II, *Computer Enabled Joystick*, of this manual for details.)

The system command:

```
S <CR> <LF>
```

puts Unidex 11 into the joystick mode. The display changes to the joystick mode tracking display.

In a system with more than 2 axes, the initially active axes will be X and Y. The joystick may now be operated to move the axes.

Y. TRIGGER

To execute the program that is suspended with a Hold command (H), send:

```
T <CR> <LF>
```

(or execute the bus function "Group Execute Trigger" (GET))

SECTION 5-2 MOTION PROGRAM COMMANDS

The motion program commands make up the program that Unidex 11 executes when running in the auto or block mode. While all of the program commands are valid in the edit mode, only some are valid in the immediate mode. A list of the motion program commands that will operate in the immediate mode are listed in section 5-1 M (*Immediate mode*) of this chapter.

A. END OF BLOCK

An end-of-block terminates a block of a program. It may be one of two characters:

```
* or /
```

If two or more axes are to run simultaneously, keep those axes commands within one block, i.e., place an end-of-block character after the axes moves have been entered. For example:

```
X F10000 D150000
Y F10000 D150000
V F500 D10000 * (or /)
```

B. AXIS MOTION COMMANDS

The axis to move must, of course, be specified by an axis command (X, Y, U or V).

The speed with which it travels must be specified by a feedrate command (F).

The distance which it is to travel (or the position it is to attain if in the Absolute Mode) must be specified with a distance command (D).

An example of a program block utilizing the above commands is:

```
X F10000 D150000
```

The above command would send the X axis a distance of 150000 system units at a feedrate of 10000 system units/second.

NOTE: When programming through IEEE-488 communication, the feedrate need only be entered with the first index block. After that, the feedrate need only be entered if it is to be changed.

1. *Axis Free-Run*

The axes are commanded to free-run by the command R and a " + " or "-" sign to signify CW (+) or CCW (-). Example:

```
Y F10000 R+ *
```

The above command tells the Y axis to free-run in the CW direction at a speed of 10000 system units/second.

C. DWELL

A program dwell can be entered into your program with a DW command, followed by the duration of the dwell in tenths of seconds. Example:

```
DW 10 *   or   DW 1.0 *
```

The above command will cause a 1 second dwell within your program.

D. HOME

Send an axis Home with command H followed by the axis or axes to be sent home and an end-of-block character. Example:

```
H X *
or
H XYUV *
```

The first example will send X axis home. The second command will send all axes home.

E. OUTPUT STATE

To establish the output as a 1 (true), a 0 (false) or an X (don't care), send the command "OT" followed by a 1, 0 or X. Example:

```
OT 10XX *
```

In the above example, O1 is to be programmed to a 1, O2 to a 0, O3 is a "don't care" and O4 is also a "don't care". "Don't care" leaves the state of an output unchanged.

F. INPUT STATE

To set up the state you wish the inputs to attain before the program continues, program an "IT" command, followed by the desired statuses and an end-of-block character. Example:

```
IT X001 *
```

The above command states that the program should wait until I2 is a 0, I3 is a 0 and I4 is a 1. State of Input I1 does not matter since it is programmed as a "don't care".

G. OUT/STOP STATE

To put out values to the outputs when the program is stopped (through the STOP key, a feedhold or pressing key #3 on the front panel), enter the command "OS" followed by the desired values and an end-of-block character. Example:

```
OS 0011 *
```

When the program is stopped, a zero will be output on O1 and O2, a one will be output on O3 and O4.

H. OUT/RUN STATE

To output values when the program is allowed to run again by pressing RUN or releasing the feedhold, (after the STOP key, key #3, or feedhold has been pressed), program "OR" followed by the desired values and an end-of-block. Example:

```
OR XXXX *
```

In the above example, when the program run is allowed to continue, the outputs will remain unchanged (because "don't care" has been programmed for each output).

I. REPEAT LOOP START

The command to start a repeat loop in your program, and the number of times the loop is to be executed, is established with a "RS" command. It is followed by the number of times to repeat and an end-of-block. Example:

```
RS 8 *
```

The repeat loop beginning has been marked, and the loop is to be repeated 8 times.

J. REPEAT LOOP END

To mark the end of the repeat loop started as discussed in the above subsection, program command "RE" followed by an end-of- block. Example:

```
RE *
```

The repeat loop may be nested 8 levels deep.

K. CONDITIONAL REPEAT LOOP END

To end the repeat loop based on input conditions, program "RC" followed by the required input state and an end-of-block. Example:

```
RC 10XX *
```

The above example states that the repeat loop will end when the inputs are as follows: I1 is a 1, I2 is a 0. I3 and I4 have no control over the above program block.

L. STARTING FREE RUN AFTER A STOP FREE RUN

After a free run has been stopped (discussed in the next subsection), programming an "R" followed by the axis or axes to be restarted, will start the axis or axes again. Example:

```
R X *  (or)
R XYUV *
```

The first example restarts an X-axis free run. The second example restarts an all-axes free run.

M. STOP AXES FREE RUN

To stop an axis or axes' free run, program an "S" command, followed by those axes that are to be stopped, and an end-of-block character. Example:

```
S X *  (or)
S XY *  (or)
S XYUV *
```

The first example stops an X-axis free run. The second stops an X and Y-axes free run. The third stops an all-axes free run.

N. REPEAT PROGRAM

To repeat the entire program from the start, enter command "RP" and an end-of-block. Example:

```
RP *
```

Remember, any commands following this command within your program will not be executed.

O. LOAD POSITION REGISTERS

You may load any of the axes position registers with an "L" command, followed by a distance and an end-of-block. The value, which is in system units, may be a positive or negative number or may be a zero. Example:

```
LX0Y0 *
```

In the above example, the X and Y position registers are loaded with zeros. This command may be used to establish an absolute reference position. The axes may be moved to this reference position by programming, in the absolute mode, an Index block with distance values equal to the reference position.

P. INCREMENTAL MODE /ABSOLUTE MODE

In the incremental mode, a distance command tells Unidex 11 how much further to move the axes. For example:

```
X F10000 D1000 *
```

This command would move the X axis 1000 steps in the CW direction each time it was executed. Program "IN", followed by an end-of-block, to establish the incremental mode. Example:

```
IN *
```

In the absolute mode, on the other hand, a distance command is an absolute position. For example:

```
X F10000 D1000 *
```

When in the absolute mode, the above example tells Unidex 11 to send the X axis to the position 1000. Once there, the reexecution of the above command will not move the axis any further since it is already at the position commanded. To establish the absolute mode of programming, enter:

```
AB *
```

Q. BEEPER

To turn the beeper ON, program "BN" followed by end-of-block. To turn it OFF, program "BF" followed by end-of-block. Example:

```
BN *
DW .5 *
BF *
```

The above example turns on the beeper for 5/10 of a second and then turns it off.

R. LABEL

A label (0 to 99) labels a block of program as the place to which the program goes when a GOSUB (go to subroutine) or a GOTO command is encountered. It is programmed with a "LB" command, followed by the number and an end-of-block. Example:

```
LB55*
```

S. GOTO

This command directs program flow to a label. Enter "GT" and a label number, followed by an end-of-block. Example:

```
GT 20 *
```

The above command tells Unidex 11 to continue program execution at label 20.

T. GOSUB

The command that tells Unidex 11 to execute a subroutine at label #nn is the "GS" command, followed by the block label number and an end-of-block. Example:

```
GS 15 *
```

The subroutine to be executed is located at label #15.

U. SUBROUTINE RETURN

This command causes Unidex 11 to return from the subroutine execution to the program block immediately after the "GS nn" block that called the subroutine. Every subroutine should end with a Return from Subroutine.

```
SR *
```

Subroutines may be nested 8 levels deep.

V. PROGRAM STOP

Program stop marks the place in the program at which program execution ends. Enter:

```
PS *
```

Subroutines may be placed after this block.

W. CONDITIONAL GOTO

The command "CT", followed by a label number, an "I" and four input values, states that the program should go to a specific block if the input statuses match the specified values. For example:

```
CT 22 I 10X0 *
```

The above command tells Unidex 11 to go to the program block labeled "22" when I1 is 1, I2 is 0, and I4 is 0. If the inputs are not these values, continue to the next program block.

X. CONDITIONAL GOSUB

The command to send the program to a subroutine if the input statuses match specified values is "CS", followed by the label number, an "I" and the desired input values. For example:

```
CS 33 I 110X
```

The above command tells Unidex 11 to go to the subroutine labeled "33" when the value of I1 is 1, I2 is 1, I3 is 0. If these input conditions do not exist, continue with the next program block.

Conditional subroutines, in combination with regular subroutines, may be nested 8 levels deep.

Y. CORNER ROUNDING /NON-CORNER ROUNDING

The command to implement corner rounding is "CO" followed by an end-of-block. (It can only apply to *all* axes in the system.) Example:

```
CO *
```

NOTE: Corner rounding mode is not recommended for stepping motor control.

The CO command would enable corner rounding for all four axes.

The command to go back to non-corner rounding is "NC" followed by an end-of-block. Example:

```
NC *
```

The above command would take all four axes out of the corner rounding mode.

Z. ACCELERATION/DECELERATION RAMP TIME

The accel/decel ramp time can be programmed in milliseconds by the command "AD" followed by the desired time and an end-of-block character. Example:

```
AD 10 * or AD 5821 *
```

In the above example, the acceleration ramp time as well as the deceleration ramp time will be 10 milliseconds. The maximum value is 9999. This command will be executed only when the required option is included in the system.

AA. BINARY DIGITAL OUTPUT/BCD DIGITAL OUTPUT

To program the Digital Output to be a binary number, enter command "DD" followed by the number ("nnnn") to be output and an end-of-block.

Note: (nnnn < = 4095). The binary equivalent of the number you enter will be output. Example:

```
DD 22 *
```

In the above example the binary equivalent of the number 22 (000000010110) will be output on the 12 output lines.

To program the Digital Output to be a BCD number, enter command "DB", followed by number ("nnn") to be output and an end-of-block. Note: (nnn < = 999). Example:

```
DB 22 *
```

The number 22 will be output in BCD format (0000 0010 0010). A mSec strobe will be output on the strobe line after the 12 bit value settles.

BB. END EDIT

The character that is placed at the end of a downloadable file is the percent sign (%). Just place it at the end of your file without an end-of-block. Example:

```
%
```

When downloading a program, this character takes Unidex 11 out of the edit mode and back to the system command mode.

SECTION 5-3 SAMPLE IEEE-488 COMMANDS

A few brief samples will help demonstrate the commands discussed in the last section.

SAMPLE IMMEDIATE COMMANDS

```
Address Unidex 11 to listen             ; Interface active
I H XY * <CR> <LF>                      ; Send home X and Y
I X F10000 D10000 * <CR> <LF>           ; Move X axis
I Y F10000 D10000 * <CR> <LF>           ; Move Y axis
I BN * <CR> <LF>                        ; Beeper ON
```

```
I BF * <CR> <LF>                      ; Beeper OFF
I X F100 D1000 Y F100 D1000 *         ; X and Y axes move
<CR> <LF>
```

SAMPLE MOTION PROGRAM

```
Address Unidex 11 to listen           ; Interface active
E 01 *                                ; Edit program 1
H XY *                                ; Send X and Y axes home
X F10000 D10000 *                     ; Move X axis
Y F10000 D10000 *                     ; Move Y axis
BN *                                  ; Beeper ON
DW .2 *                               ; Dwell for 2/10 second
BF *                                  ; Beeper OFF
%                                     ; End edit mode
```

Send F <CR> <LF> to set Unidex 11 to the block number printing mode.

```
P01 <CR> <LF>
```

The above command will cause program #1 to be printed with block numbers when Unidex 11 is addressed to talk.

Cancel numbering with a G command.

Program #1 may be run block by block by sending:

```
B01 <CR> <LF>
```

and successive <CR> <LF> for successive blocks.

To run this same program in the Auto Mode, send:

```
A01 <CR> <LF>
```

It may be deleted by sending:

```
E $ 01 *
```

Check your directory with a PD command to verify that program #1 has been deleted.

The X position register may be read by sending:

PX <CR> <LF>
(and addressing Unidex 11 to talk)

The Y position register may be read by sending:

PY <CR> <LF>
(and addressing Unidex 11 to talk)

Send J <CR> <LF> to put Unidex 11 in the Service Request Mode. When Unidex 11 asserts SRQ, perform a serial poll before continuing. If running the program in the auto mode, the service request will come after the program execution. If running a program one block at a time, it will follow each block.

Send command K <CR> <LF> to cancel the service request mode.

Appendix 1
COMMAND SUMMARY FOR UNIDEX 11 : IEEE-488

NOTE: All system commands must be entered as upper case letters.

The following is a list of commands for the Unidex 11 when operating through the IEEE-488 communication bus in the system command mode.

SYSTEM COMMANDS

A nn <CR> <LF> : Run program # nn in AUTO mode (nn = 0 to 99).

B nn <CR> <LF> : Run program # nn in BLOCK mode (subsequent <CR> <LF> will execute successive program blocks).

C <CR> <LF> : Reset Unidex 11.

D : Cancel S or R command.

E nn * : Begin downloading program #nn. Existing program #nn will get deleted automatically.

E $ nn * : Delete program # nn.

E $ 00 * : Clear program memory (all programs).

F <CR> <LF> : Insert block numbers when printing programs.

G <CR> <LF> : Cancel block number printing (default state)

H <CR> <LF> : Put Unidex 11 in HOLD mode (Trigger required to execute programs). Hold mode canceled by O <CR> <LF>.

I (string) * <CR> <LF> : Execute program block (string) in the immediate mode.

J <CR> <LF> : Set up Unidex 11 to assert Service Request after execution.

K <CR> <LF> : Cancel set up to send SRQ (default state)

L <CR> <LF> : Not used

M <CR> <LF> : Set up to transmit status in binary format (default state)

N <CR> <LF> : Set up to transmit status in Hex-ASCII format

O <CR> <LF> : Cancel HOLD mode (default state)

PX <CR> <LF> : Print X axis position register value when addressed to talk

PY <CR> <LF> : Print Y axis position register value when addressed to talk

PU <CR><LF>	: Print U axis position register value when addressed to talk
PV <CR><LF>	: Print V axis position register value when addressed to talk
PD <CR><LF>	: Print Directory listing when addressed to talk
Pnn <CR><LF>	: Print program #nn when addressed to talk
P00 <CR><LF>	: Print all programs in memory when addressed to talk
PS <CR><LF>	: Print Status bytes when addressed to talk
Q	: Not used
R <CR><LF>	: Enable Remote Mode from host controller
S <CR><LF>	: Enable Joystick Mode from host controller
T <CR><LF>	: Trigger to start program execution

MOTION PROGRAM COMMANDS

* or /	: End of block (terminates block)
X F ffffff D dddddddddd	: X axis move at feedrate ffffff steps/sec a distance of dddddddddd steps, CW
Y F ffff R +	: Y axis feedrate ffff steps/sec, free run CW
U F fffff R -	: U axis feedrate fffff steps/sec, free run CCW
V F ff D - dddd	: V axis move at feedrate ff steps/sec, a distance of dddd steps CCW
	(* placed as required)
DW nnn.n *	: Dwell nnn.n seconds
H X *	: X axis home
H Y *	: Y axis home
H XYUV *	: All four axes home
OT 10XX *	: Output O1, O2, O3, O4
	1: True 0: False X: Don't care
ITX001*	: Wait until input state I1, I2, I3, I4 matches X001
OS 0011 *	: OUT/STOP state
OR XXXX *	: OUT/RUN state
RS nnnn *	: Repeat loop start nnnn times
RE *	: Repeat loop end
RC 10X0 *	: End Repeat loop on input condition 10X0
R X *	: Start free run axis X after Stop (S X)
R YUV *	: Start free run axes Y, U and V
S X *	: Stop free run axis X

S YUV *	: Stop free run axes Y, U and V
RP *	: Repeat Program
L X dddddddddd	: Load position register X with dddddddddd
L Y ddd	: Load Y axis position register with ddd
L U 0	: Load U axis register with 0 (zero).
L V -dddd	: Load V axis position register with -dddd

(* placed as required)

IN *	: Incremental mode
AB *	: Absolute mode
BF *	: Beeper OFF
BN *	: Beeper ON
LB nn *	: Label # nn
GT nn *	: GoTo label #nn
GS nn *	: GoSub label #nn
SR *	: Subroutine Return
PS *	: Program Stop (end of program execution)
CT nn I 10X0 *	: Conditionally GoTo label # nn if input state is 10X0, else continue
CS nn I 1XXX *	: Conditionally GoSub label #nn if input state is 1XXX, else continue
CO *	: Corner rounding mode
NC *	: Non-corner rounding mode
nnnn *	: Accel/Decel ramp time in milliseconds
DD nnnn *	: Binary digital output (nnnn < = 4095)
DB nnn *	: BCD digital output (nnn < = 999)
%	: End edit (downloading)

Appendix 2

IEEE-488 BUS INTERFACE FUNCTIONS

These are messages sent on the bus with the ATN line asserted. The bus functions listed below are valid only when Unidex 11 is in the Interface Active mode or the Communication Enabled mode. If Unidex 11 is in the Local mode, bus handshake protocol may not be completed.

BUS FUNCTION	UNIDEX 11 RESPONSE
DCL (Not addressed) SDC (Addressed)	Goes into Power-up state (Unidex 11 requires about 2 seconds to come out of reset).
GTL (Addressed)	Goes into Local mode with Communication Enabled. Position registers and previous command block are not cleared.
GET (Addressed)	Executes command block (and sets SRQ) if in Hold mode. No response if not in Hold mode.
SPE (Addressed) SPD (Addressed)	These two components of a Serial Poll cause Unidex 11 to put out the status byte and reset SRQ if set.
UNL UNT MLA MTA LAG TAG	These are components of Data Transfers. They cause appropriate data transfer sequences
IDY (Identify) (Parallel Poll)	Puts the identity on the bus as a response to a parallel poll; transparent to Unidex 11.
TCT (Addressed)	Bus hangs up
LLO	No response

BUS FUNCTION	UNIDEX 11 RESPONSE
UUCG (Undefined Universal Command Group)	No response
UACG (Undefined Addressed Command Group)	No response

Index

PART II:

COMPUTER ENABLED JOYSTICK

(JP4C)

Table of Contents

CHAPTER 1: INTRODUCTION

This option allows a host computer controlling Unidex 11 to enable it for Joystick operation or remote (external) control through the auxiliary inputs. Manual operation of the keyboard is no longer necessary to put Unidex 11 into the joystick mode of operation, thus providing very efficient interactive control capability with a host controller.

Enabling Joystick or Remote control may be done through either communication interface (RS-232C or IEEE-488).

SECTION 1-1 REQUIRED HARDWARE

The Unidex 11 joystick and a communication interface are required to implement this option. For the relevant communication, see the *Unidex 11 Motion Controller Programming Manual* for information on the RS-232 Interface, or Part I of this manual for information on the IEEE-488 Interface. For any hardware considerations, see the *Unidex 11 Motion Controller Hardware Manual.*

CHAPTER 2: OPERATION

SECTION 2-1 JOYSTICK CONTROL

The Unidex 11 Joystick can be enabled and disabled by simple system commands from the host controller through the communication interface.

A. ENABLING THE JOYSTICK

The system command:

```
S <CR> <LF>
```

puts Unidex 11 into joystick mode. The display will change to the joystick mode tracking display (Part II, section 2-2 H of the *Unidex 11 Motion Controller Programming Manual*).

```
jkm   X: 0000023456   step
jk    Y:-0002345.12   mm
```

In a system with more than 2 axes, the initially active axes will be X and Y. The joystick may now be operated to move the axes.

When an axis runs into a limit, Unidex 11 indicates the condition by flashing the "Axis in Limit" message on the display and beeping. Reversing the joystick direction of the axis will automatically remove the limit condition.

B. BUTTONS A AND B

The active axes pair can be changed by pressing Button A. Pressing Button B will decrease joystick speed by a factor of 64. Pressing it again will return to the original speed. (Refer to Part II of the *Unidex 11 Motion Controller Programming Manual*, section 2-2H, for a detailed description of Buttons A and B.)

C. BUTTON C

Pressing Button C will cause one of two different actions on the part of Unidex 11:

1. **WHEN UNIDEX 11 IS IN SERVICE REQUEST MODE:** (For information on Service Request and Serial Poll, refer to Part IV, section 3-3, of the *Unidex 11 Motion Controller Programming Manual* (for RS-232), or section 3-3 of Part I of this manual (for IEEE-488).

   When Unidex 11 is in the Service Request Mode and Button C is pressed, Unidex 11 asserts a service request as follows:

   With the RS-232C interface, this is implemented by sending a character from Unidex 11 and waiting for a serial poll command "Q <CR> <LF>".

   With the IEEE-488 interface, Service Request is a primary bus function and Unidex 11 asserts the SRQ line and waits for a serial poll by the controller.

   When Unidex 11 asserts a service request, the host controller is required to do a serial poll. After the serial poll, the host must disable the joystick before sending any system commands, except the Print Position commands.

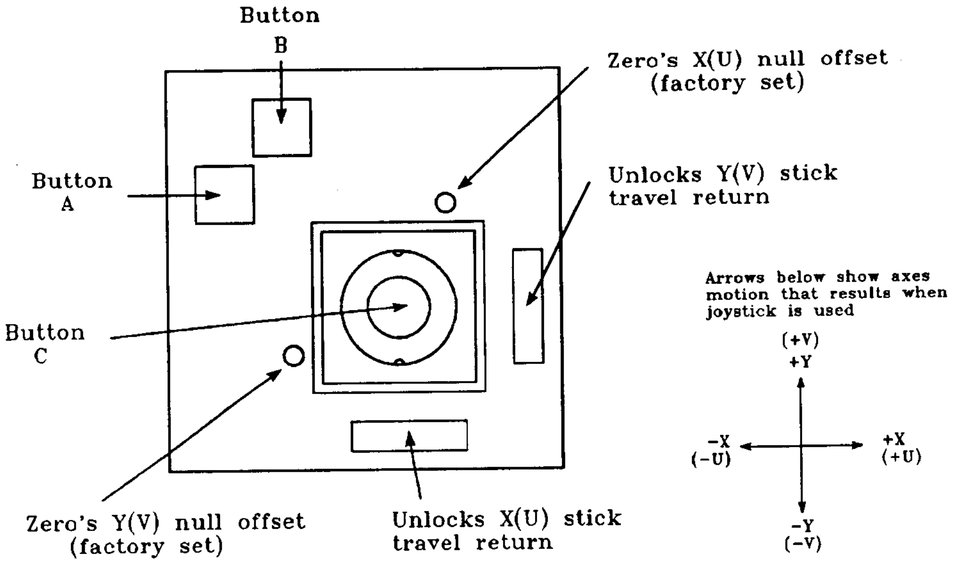


Figure 2-1: Unidex 11 Joystick

Positions may be read without disabling the joystick after a serial poll. (This leaves the joystick status unchanged.)

Responding to the service request through an interrupt routine in the host controller software will speed up overall system response time. The following sample program, written on an HP-85 desktop computer using the IEEE-488 interface, prints the axis position values when Unidex 11 asserts an SRQ. The device address of the Unidex 11 in the following example is 2.

SAMPLE PROGRAM

```
10 CLEAR                        ! Clear screen
20 REMOTE 702                   ! Interface active
30 SET TIMEOUT 7 ;1000          ! 1 second interface
                                ! timeout
40 OUTPUT 702; "J"              ! Unidex 11 in SRQ mode
50 ON INTR 7 GOSUB 200          ! Define interrupt routine
60 ENABLE INTR 7; 8             ! Enable SRQ interrupt
```

```
70  OUTPUT 702; "S"              ! Enable joystick
80  GOTO 80                      ! Wait for SRQ
                                 ! Interrupt routine
200 OFF INTR 7                   ! Turn off interrupt
210 STATUS 7,1;S1                ! Clear flag
220 S = SPOLL(702)               ! Serial poll
230 OUTPUT 702; "D"              ! Disable joystick (optional)
240 OUTPUT 702 USING "K";        ! Command to print
"PXYV" @ ENTER 702               ! position values
USING "K";X,Y,V @
DISP X;Y;V
250 OUTPUT 702; "S"              ! Enable joystick mode (option-
                                 al)
260 ON INTR 7 GOSUB 200          ! Return from interrupt
@Enable INTR 7; 8 @
Return
```

2. WHEN UNIDEX 11 IS IN THE SERVICE REQUEST CANCEL MODE

When Unidex 11 is in the Service Request Cancel Mode, pressing Button C will cause Unidex 11 to send the absolute axis position register values to the controller. Each position value is 11 characters (a space or a minus sign followed by 10 digits), terminated by a <CR> <LF>. The output from Unidex 11 for a three axis system is shown below:

```
 1234567890 <CR> <LF>          (X POSITION)
-0000123.56 <CR> <LF>          (Y POSITION)
 0000000012 <CR> <LF>          (V POSITION)
```

The axes' data is sent out in an X, Y, U, V order.

With the RS-232 interface, sending data to the host controller does not require a prompt from the controller. The host should, however, be ready to accept the data when it is sent.

With the IEEE-488 interface, the host is required to address Unidex 11 to talk before either of them can send data. Pressing Button C will cause Unidex 11 to wait until addressed to talk. The following program for a HP-85 desk-top computer illustrates this:

```
10  CLEAR                         ! Clear screen
20  SET TIMEOUT 7; 10000          ! Button C should be
                                  ! pressed every 10 seconds
                                  ! to prevent timeout.
30  REMOTE 702                    ! IEEE-488 interface active
40  OUTPUT 702 ;"S"               ! Enable joystick
50  ENTER 702 USING "K";          ! Read data
X,Y,V                             ! from Unidex 11
60  DISP X;Y;V                    ! Display data
70  GOTO 50                       ! Wait for next button C
                                  ! operation
```

D. DISABLING THE JOYSTICK

The system command:

```
D <CR> <LF>
```

disables the joystick and returns control to the host computer. The position registers are updated with the absolute position values before returning control.

When the joystick is active after an "S" command, Unidex 11 will only recognize the "D" command and the serial poll command. All other system commands will be ignored.

SECTION 2-2 REMOTE (AUXILIARY) CONTROL

The Remote mode enables optional external clock and direction signals to drive the axes. The connections are made at connectors on the back of the respective Unidex 11 unit. (Refer to the *Unidex 11 Motion Controller Programming Manual*, Part II, section 2-2, and *Unidex 11 Motion Controller Hardware Manual*, for more information.)

A. ENABLING REMOTE MODE

The system command:

```
R <CR> <LF>
```

enables Unidex 11 to be driven through the auxiliary controls. The display shows the remote mode tracking screen.

```
rtm  X:  0000023456   step
rt    Y: -0002345.12    mm
```

The host controller may now signal an external device to take control of Unidex 11.

Unidex 11 keeps track of the axis positions during external control.

B. DISABLING REMOTE MODE

The system command:

```
D <CR> <LF>
```

disables the remote mode and returns control to the host. The position registers are updated with the absolute position values before returning control.

When remote mode is active after an "R" command, Unidex 11 will only recognize the "D" command and the serial poll command. All other system commands will be ignored.

C. ENABLE JOYSTICK

The system command "U" < CR > < LF > will enable the joystick just as command "S" < CR > < LF >, but with the following differences:

1. *Unidex 11 is forced into the SRQ mode*

The three joystick buttons function the same as when the joystick is enabled by the "S" command.

2. *When an axis runs into a limit:*

a. The joystick is disabled. (Unidex 11 automatically executes a "D" command).
b. The clock pulses sent to the axes are shut off.
c. The position registers are updated.
d. Unidex 11 performs a service request.

The controller may now serial poll Unidex 11 to receive an error status and return Unidex 11 to the system command level. The axis in limit may be identified by decoding the detailed status bytes that are available using the print command "PS". The positionn registers may be read using the print position commands, "PX", "PY", "PU" and "PV".

Unidex 11 remains in the SRQ mode.

The limit flags in the detailed status bytes are cleared when:

a. Unidex 11 is put into the joystick mode again, or

b. When a motion command is executed.

The SRQ mode may be cancelled using the "K" command.

Index

PART III:

JOYSTICK DIGITIZING

(JP4D)

Table of Contents

CHAPTER 1: INTRODUCTION

The Joystick Digitizing Option for the Unidex 11 provides the ability to obtain and remember one or more sampled axis positions or moves. This option allows for up to 4 axes of positioning and will output the information in either the absolute or incremental mode. Positions may be stored in memory in the form of programs or may be transmitted externally through RS-232 (see *Unidex 11 Motion Controller Programming Manual*).

During the digitizing process, the axis selection (X/Y or U/V) and the speed range (hi/low) can be selected by two buttons on the joystick. The third button will cause the position sample to be taken and recorded when pressed (see section 2-2 H of Part II of the *Unidex 11 Motion Controller Programming Manual*). All axis positions are displayed on the liquid crystal displays (LCDs) while digitizing.

It is recommended that you review the *Unidex 11 Motion Controller Programming Manual* (Part IV), before utilizing RS-232.

SECTION 1-1 REQUIRED HARDWARE

The Unidex 11 joystick is required to operate the joystick digitizing option. (Refer to the *Unidex 11 Motion Controller Hardware Manual.)*

CHAPTER 2: OPERATION

SECTION 2-1 INITIAL SETUP

Initial set-up for digitizing is done before entering the digitizing mode. This set up will depend on whether the digitized points will be stored in Unidex 11 as a program or transmitted externally via RS-232. If the digitized points will be stored in Unidex 11, the available free memory space should be checked and additional space made free if more space is needed. When RS-232 will be used to transmit the digitized points, the set-up requirements will be found in *Unidex 11 Motion Controller Programming Manual.*

If digitizing is to begin at home or at another known position, the axis should be positioned at that point before entering the digitizing mode. After initial set up is complete, the user is ready to enter the digitizing mode.

SECTION 2-2 DIGITIZING SET-UP

A. DIGITIZE SELECT

Entry into the Digitizing mode is through the Joystick screen (refer to the *Unidex 11 Motion Controller Programming Manual*). After selecting the joystick mode (key #4), you will see the following screen:

```
1. X:Y   3. DIVISOR   5. DGTZE
2. U:V   4. SEL PGM
```

NOTE: This screen is present only in a Unidex 11 with JP4D option.

Press key #5 to select digitizing.

B. STORE/TRANSFER SELECT

After pressing key #5 in the Digitize Select Screen (section 2- 2A), the display will show the Store/Transfer Screen. If RS- 232 is to be used, the external device should be in the ready mode:

```
1. Store in User Memory
2. Transfer via RS-232
```

If key #1 is pressed, the digitizing points will be stored in the Unidex 11 user memory in standard program format.

If key #2 is pressed, the digitizing points will be sent to an external device through RS-232 in the standard RS-232 program format.

Pressing either 1 or 2 will cause the next digitizing set-up screen, the Program # Screen, to appear.

C. PROGRAM # SCREEN

The program number screen is used to enter the program number when the digitized points are to be stored or transmitted.

```
Input  Program
      00
```

To enter the program #, enter the number and press the ENTER key.

After the program number has been entered, it will be transmitted through RS-232 or stored in Unidex 11 memory, depending on the Store/Transfer set-up (section 2-2 B of this chapter). If the program is to be stored in the Unidex 11 memory and there is an existing program with the same number as the digitizing program number, the digitizing information will be inserted as index blocks at the end of the program.

D. INCREMENTAL/ ABSOLUTE MODE

After the program number has been entered, the incremental/ absolute screen will appear. This screen selects whether the digitized points will be in the form of absolute position or incremental moves. Press the +/- key to select and press ENTER to set the mode.

```
Inc/Abs (+/- & Enter)
     INCREMENTAL
```

```
Inc/Abs (+/- & Enter)
       ABSOLUTE
```

Absolute mode is the actual position being sampled. Incremental mode is the difference between the last sample and the present sample. No index blocks will be stored or transmitted for any axis if no change in position has taken place since the last sample.

E. FEEDRATE SETUP

After entering the incremental or absolute mode, the feedrate set-up screen will be displayed. This screen allows the operator to set the feedrate for each digitized movement. This feedrate does not effect the joystick speed while digitizing.

```
X  F 0010000   D 0000000000
Y  F 0010000   D 0000000000
```

The D (distance) portion of the screen is loaded during digitizing. This part of the screen(s) should be ignored. The (F) feedrate should be entered in the manner described in section 2-2 D of Part II of the *Unidex 11 Motion Controller Programming Manual.* During digitizing, the block (or screen) shown above will be duplicated automatically with the D (distance) parameter updated for each digitized sequence. The F (feedrate) for each digitized block will be duplicated with the initial value which was programmed above. This allows the user to return to the program allocated for digitizing and change the feedrates for given digitized blocks, if desired. After all feedrates have been entered, press ENTER to get into the digitizing mode.

F. DIGITIZING TRACKING DISPLAY SCREEN

The digitizing screen is always displayed during the digitizing process. This screen is the same as the Joystick Tracking Display screen, except instead of displaying "jk" (joystick), the digitizing screen will contain "jd" (joystick digitizing). For additional information, refer to *Unidex 11 Motion Controller Programming Manual,* section 2-2H of Part II.

```
jd X : 0000000000  step
jd Y : 0000000000  step
```

At this time, the joystick is active and digitizing is operative. See section 2-2G of this manual for joystick operation. During this time, the following keys on Unidex 11 are active:

INSERT	Displays again the Digitized Feedrate screen (section 2-2 E of this chapter) to allow feedrates to be changed for upcoming digitized blocks.
DELETE	*CLEAR ABSOLUTE POSITION.* (NOTE: Use of this key will reset the absolute position and display all zeroes. If Button "C" (digitizing button of the joystick) is pressed after pressing "Delete", this "change" in position will be recorded. In the absolute mode, zeroes will be recorded. In the incremental mode, zero minus the previous position (i.e., the change in position) will be recorded.
BACK	*QUIT.* See section 2-2 H of this chapter.

G. JOYSTICK OPERATION

The operation of the joystick in the digitizing mode is the same as in the joystick mode except for button "C". Button "C" is used to digitize the current position. When pressed, a beep will sound and the digitized point will be sampled, processed, and stored as an index block. Samples will only be taken for an axis that has changed its position since the last sample.

H. DIGITIZING TERMINATION

To terminate digitizing, press the Back key when in the Digitizing Tracking Display screen (section 2-2F). When the Back key is pressed, digitizing will be terminated and Unidex 11 will transmit the End of Program characters if RS-232 is active (section 3-2, RS-232C Format). The Unidex 11 will then go back to the joystick screen. Pressing Back again will send Unidex 11 to the second main menu screen (see the *Unidex 11 Motion Controller Programming Manual*).

CHAPTER 3: DIGITIZING FORMATS

SECTION 3-1 STORE FORMAT

A brief sample program is shown below (in Unidex 11 "Menu" format) to illustrate how sampled points can be stored, as well as some of the hidden capabilities of digitizing.

SAMPLE PROGRAM #1

```
0          PROGRAM #07
1          ABSOLUTE MODE
2          INDEX X F1000  D78
           Y F500   D-61
           U F500   D125
           V F  1   D62
3          INDEX X F1000  D123
4          INDEX Y F500   D-278
5          INDEX X F1000  D17
           Y F500   D-2
           U F500   D-24
           V F1     D55
6          INDEX Y F500   D11
7          END
```

NOTE: Program, absolute mode, end statement and feedrates have been determined previously (section 2-2 C - F of this chapter).

Sample program #1 was generated in the following manner.

LINE 0	Set for Store Data, Program #7
LINE 1	Absolute mode; set feedrate
LINE 2	Four axes moved, positions sampled and stored
LINE 3	Only X axis moved, position sampled and stored
LINE 4	Only Y axis moved, position sampled and stored
LINE 5	Four axes move, positions sampled and stored.
LINE 6	Only Y axis moved, position sampled and stored.
LINE 7	Exit digitizing (press BACK key in Digitizing Tracking Display).

SECTION 3-2 RS-232C FORMAT

Digitized points are transmitted in an index block format. The following sample program shows a typical transfer format example.

SAMPLE PROGRAM #2

```
0          E06 *                   PROGRAM #6

1          AB *                    ; ABSOLUTE MODE

2          X F1000 D21             ; INDEX X AXIS
           Y F1000 D-7             ; "  Y AXIS
           U F500 D64              ; "  U AXIS
           V F1000 D1 *            ; "  V AXIS

3          Y F1000 D-21 *          ; INDEX Y AXIS

4          X F1000 D35 *           ; INDEX X AXIS

5          % <CR> <LF>             ; END OF PROGRAM
```

The sample program (#2) was generated in the following manner:

Set up RS-232 interface and enable.

LINE 0	Select digitizing, transfer through RS-232, program #06
LINE 1	Absolute mode; set feedrate
LINE 2	All axis positions digitized and transmitted (absolute mode)
LINE 3	Y axis move, position digitized and transmitted
LINE 4	Only X axis moved. Position digitized and transmitted
LINE 5	Press BACK key (Digitizing Tracking Display) to exit digitizing and output %, CR LF (End of Program characters)

Additional information concerning RS-232 may be found in the *Unidex 11 Motion Controller Programming Manual.*

CHAPTER 4: TROUBLESHOOTING

This chapter is divided into two sections, Software Malfunctions and Hardware Malfunctions. Software malfunctions are problems displayed as error messages or codes on the Unidex 11 display. Hardware malfunctions are problems that do not provide error messages or codes on the Unidex 11 display.

SECTION 4-1 SOFTWARE MALFUNCTIONS

Software malfunctions are displayed on the Unidex 11 display in the form of error codes. In digitizing there is only one directly related error message (MEMORY FULL). All error messages including the Memory Full message are described in the *Unidex 11 Motion Controller Programming Manual*, Part II.

SECTION 4-2 HARDWARE MALFUNCTIONS

All hardware malfunctions concerning digitizing should be isolated to one of the following three areas. **NOTE:** Refer to section 4-1, above, if error messages are displayed.

- **A. Joystick Control**
- **B. RS-232 Control (if used)**
- **C. Unidex 11 Editor and Memory**

A. JOYSTICK CHECK

The joystick operation can be checked by going into joystick operation without digitizing. Refer to the *Unidex 11 Motion Controller Programming Manual* (Part II, section 2-2 H). If operation is satisfactory, the joystick is most likely not causing the problem and the next related area should be checked.

B. RS-232 CHECK

If RS-232 is being used, it should be checked independent of digitizing by going into Print mode and checking the print operation. See the *Unidex 11 Motion Controller Programming Manual*, Part IV. If the RS-232 is working correctly check the next related area.

C. EDITOR AND MEMORY CHECK

The third section to check is the editor and memory. To check memory, see Part II, section 2-2 O of the *Unidex 11 Motion Controller Programming Manual.* To check the Editor operation, key a small sample program into the Unidex 11 user memory. Review the program and verify that the program is present and all blocks are valid.

Index

PART IV:

DIGITAL
I/O PORT

Table of Contents

CHAPTER 1: INTRODUCTION

The Digital I/O Option can be used as an output port for up to 12 bits of information or as a input port for calling programs. Both input and output modes utilize a strobe line to assist in gating information into and out of this port.

The outputs can be configured through manual (immediate) commands or by programmed commands. Outputs can be set up as either binary (0 - 4095) or BCD (0 - 999). A strobe output is provided for both BCD and binary outputs for interfacing purposes.

When the parallel port is set up as an input, programs may be called and executed. Program numbers are entered in BCD (0 - 99). The strobe input is used to gate in the program number input. Two additional lines, which are outputs, serve as Ready and Error indicators.

Before proceeding, it is assumed that the user has reviewed the *Unidex 11 Motion Controller Programming Manual.*

SECTION 1-1 REQUIRED HARDWARE

The digital I/O port is a 15 pin female "D" connector located at the rear of the Unidex 11 controller. (See the *Unidex 11 Motion Controller Hardware Manual* for details.)

DIGITAL I/O CONNECTION

PIN #	BINARY OUT	BCD OUT	INPUT
1	1 (BIN)	1 (BCD)	1 (BCD)
2	2 (BIN)	2 (BCD)	2 (BCD)
3	4 (BIN)	4 (BCD)	4 (BCD)
4	8 (BIN)	8 (BCD)	8 (BCD)
5	16 (BIN)	10 (BCD)	10 (BCD)
6	32 (BIN)	20 (BCD)	20 (BCD)
7	64 (BIN)	40 (BCD)	40 (BCD)
8	128 (BIN)	80 (BCD)	80 (BCD)
9	256 (BIN)	100 (BCD)	$\overline{\text{Strobe In}}$
10	512 (BIN)	200 (BCD)	$\overline{\text{Ready Out}}$
11	1024 (BIN)	400 (BCD)	Error Out
12	2048 (BIN)	800 (BCD)	----------
13	$\overline{\text{Strobe Out}}$	$\overline{\text{Strobe Out}}$	----------
14	Com	Com	Com
15	+5V *	+5V *	+5V *

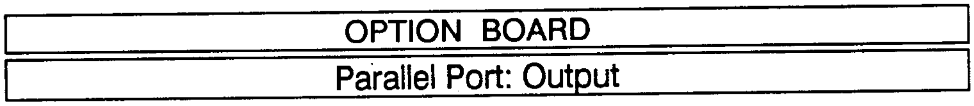
* Pin 15 was Common, prior to ECN 0680 and/or Rev.B of the Unidex 11 Option Board (Board number 690D1320).

CHAPTER 2: SET UP AND OPERATION

This chapter describes the set up and operation of the Digital I/O Option.

SECTION 2-1 INPUT/OUTPUT SET UP

The Input/Output set up screen allows the user to select either the Input or the Ouput mode. The Digital I/O set up screen can be selected by pressing the #3 key (Set Up) when in the third main menu (refer to Part II, section 2-2 K of the *Unidex 11 Motion Controller Programming Manual*). Step through the set up screens (press SELECT key) until the Digital I/O Set Up screen is displayed:



```
OPTION BOARD
Parallel Port: Output
```

This screen sets up the I/O port as either an input or output. Press the +/- key to change the mode (output or input). After the mode has been selected, press the SELECT key to exit this screen. Continue to depress the SELECT key for each of the following set up screens until you have returned to the main menu screen.

SECTION 2-2 OUTPUT MODE

The Digital I/O port will provide either a 12 bit (binary) or 3 digit (BCD) output with strobe. These outputs may be initiated either manually or through programs stored in the user memory. Binary/BCD mode are selected at the same time the numerical values are set up.

A. BINARY OUTPUTS

The binary output consists of 12 bits (see section 1-1 of this part of the manual) and a strobe output. In a typical output cycle the output bits are set high for 1 and low for 0. Approximately 300 microseconds later, a 1 millisecond negative pulse (strobe) is sent. This binary code is latched and will be present on the digital bus until a new command is given.

B. BCD OUTPUTS

The BCD output is a 12 line (3 digit) (see section 1-1 of this part of the manual) with strobe output. The output cycle sequence is the same as the Binary Output sequence listed above.

C. MANUAL MODE

The first step is to set this port for the output mode (section 2-1 of this chapter).

In the manual mode, outputs can be set up as binary or BCD type. This is done in the Digital Output screen. To access the Digital Output screen, the fifth main menu screen must be obtained:

```
1. DIG OUT      3. COMM ENAB
2. ACL/DCL      4. PRINT
```

Select DIG OUT (press key #1) to obtain the Digital Output screen.

```
BCD  Digital Output #
000
```

Press the +/- key to select either BCD or Binary. This also clears the number displayed. Enter the number that is to be sent to the digital output port and press ENTER when the information is complete. After pressing the ENTER key, the RUN/BACK screen will be displayed.

```
Press : RUN  to  execute
      : BACK to  quit
```

Press the RUN key to output digital code and strobe. The fifth main menu will then be displayed. If the BACK key is pressed, instead of RUN, the digital output and strobe will not occur, but the fifth main menu screen will be displayed.

D. PROGRAM MODE

To insert a digital output block into the program, go to the seventh Edit Screen (Part II, section 3-7 C in the *Unidex 11 Motion Controller Programming Manual*).

```
1. COR RND      3. ACL/DCL
2. PROG DEL     4. DIG OUT
```

Select DIG OUT (press key #4) and the Digital Output screen will appear.

```
BCD  Digital Output #
         000
```

The +/- key is used to select either BCD or Binary and clears the number if present. Enter the number and press ENTER to store. Additional editor information can be found in chapter 3 of Part II of the *Unidex 11 Motion Controller Programming Manual*.

NOTE: The set up screen for Parallel Port Output must be set for output before a program with a digital output block can be run (section 2-1 of this chapter).

SECTION 2-3 INPUT MODE

The Parallel Port Input mode provides a method by which programs (1 through 99) may be called and run from an external device.

A. ENABLE INPUT MODE

The parallel input port is enabled by the use of set up screens (section 2-1 of this chapter).

The first step is to set this port for the input mode. Refer to section 2-1 of this chapter for input mode set up. Next bring up the fifth main menu screen.

```
1. DIG OUT     3. COMM ENAB
2. ACL/DCL     4. PRINT
```

Press key #3 (COMM ENAB) and the following screen will be displayed:

```
1. RS-232/IEEE-488
2. PARALLEL PORT
```

Select the Parallel Port mode by pressing key #2. The parallel input port will now be enabled and the following screen will be displayed:

```
PARALLEL PORT ACTIVE
(To Exit Press Stop)
```

P0 - P7	- Data Inputs
P8	- Strobe Input
P9	- Ready Output
P10	- Error Output (Not Shown)

See section 2-3B for input procedure.

B. CALL-PROGRAM PROCEDURE

Before proceeding, make certain the Parallel Port is in the Input mode and has been enabled (section 2-1 of this chapter).

Initially, the inputs should be in the following states:

INITIAL SET UP

Pin #	Description	Logic Level
1	BCD Input Code 1	Don't Care
2	BCD Input Code 2	Don't Care
3	BCD Input Code 4	Don't Care
4	BCD Input Code 8	Don't Care
5	BCD Input Code 10	Don't Care
6	BCD Input Code 20	Don't Care
7	BCD Input Code 40	Don't Care
8	BCD Input Code 80	Don't Care
9	Strobe Input	HI (+5V)

INITIAL SET UP

Pin #	Description	Logic Level
10	Ready Output	LO (0V)
11	Error Output	LO (0V)
12	Not Used	------
13	Not Used	------
14	Common	Com
15	+5V *	+5V *

* Pin 15 was Common prior to ECN 0680 and/or Rev. B of the Unidex 11/4 Option Board (Board Number 690D1320).

NOTE: Pins 10 & 11 (Ready and Error) are Status Outputs.

PROCEDURE

1. Set BCD Code (Pins 1-8) HI for 1, LO for 0. (NOTE: Wait 100 microseconds after strobe is made HI before changing data, if program has just been strobed.)

2. Strobe must be HI (+5V) and $\overline{\text{READY}}$ output must be LO. Now strobe in BCD code by putting strobe input at 0V for at least 15 milliseconds. Do not change data until at least .1 milliseconds after making strobe HI (+5v).

3. The program will now be called and run by Unidex 11. If a RUN ERROR should occur, the Error Output will go HI and Unidex 11 will flash a error message (see *Unidex 11 Motion Controller Programming Manual*). The Ready Output will be HI (+5V) during the time the program is being run. When finished running the program or if an error occurs, the Ready output will go LO. The error output will be reset (LO, 0V) at the next data execution cycle.

C. POWER-UP INPUT PORT ENABLE

The power up Input Enable will be activated if the Parallel Port is set for the Input mode before power down (section 2-1 of this chapter). Battery back up will remember the mode during power off. To use this mode:

1. Apply power and put the Unidex 11 into the Parallel Port Input mode by sending a strobe pulse to it. The following screen will be displayed after the strobe pulse:

```
PARALLEL PORT ACTIVE
(To Exit Press Stop)
```

2. Unidex 11 is now in the Parallel Port Input mode and will accept commands using the procedure outlined in section 2-3B of this chapter.

D. EXITING THE INPUT MODE

To exit the Parallel Input Mode, give a command 00 BCD using the procedure in section 2-3B of this chapter. Upon receiving this command, the following screen will be displayed:

```
KEYBOARD & COMM ENABLED
Press  SELECT  for a Menu
```

The STOP key also causes Unidex 11 to exit the Parallel Input Mode. The "Ready" output will be high and the Error also will be high to indicate the STOP key mode of exit.

Press SELECT and the display will return to the main menu screen and deactivate READY and ERROR outputs.

CHAPTER 3: MASTER/SLAVE UNITS

By using the parallel and I/O ports of the Unidex 11, Master/Slave configurations can be constructed. Control between master and slave units is limited to selecting and running programs. Additional interaction can be obtained using the feedhold and I/O channels of the master and slaves. Programming the slaves involves setting the program number on the digital output of the master. All units contain their own programs and do not transfer programs or information between master and slave units.

NOTE: Master/slave configurations require additional I/O ports which will limit or eliminate the number of ports available to the user.

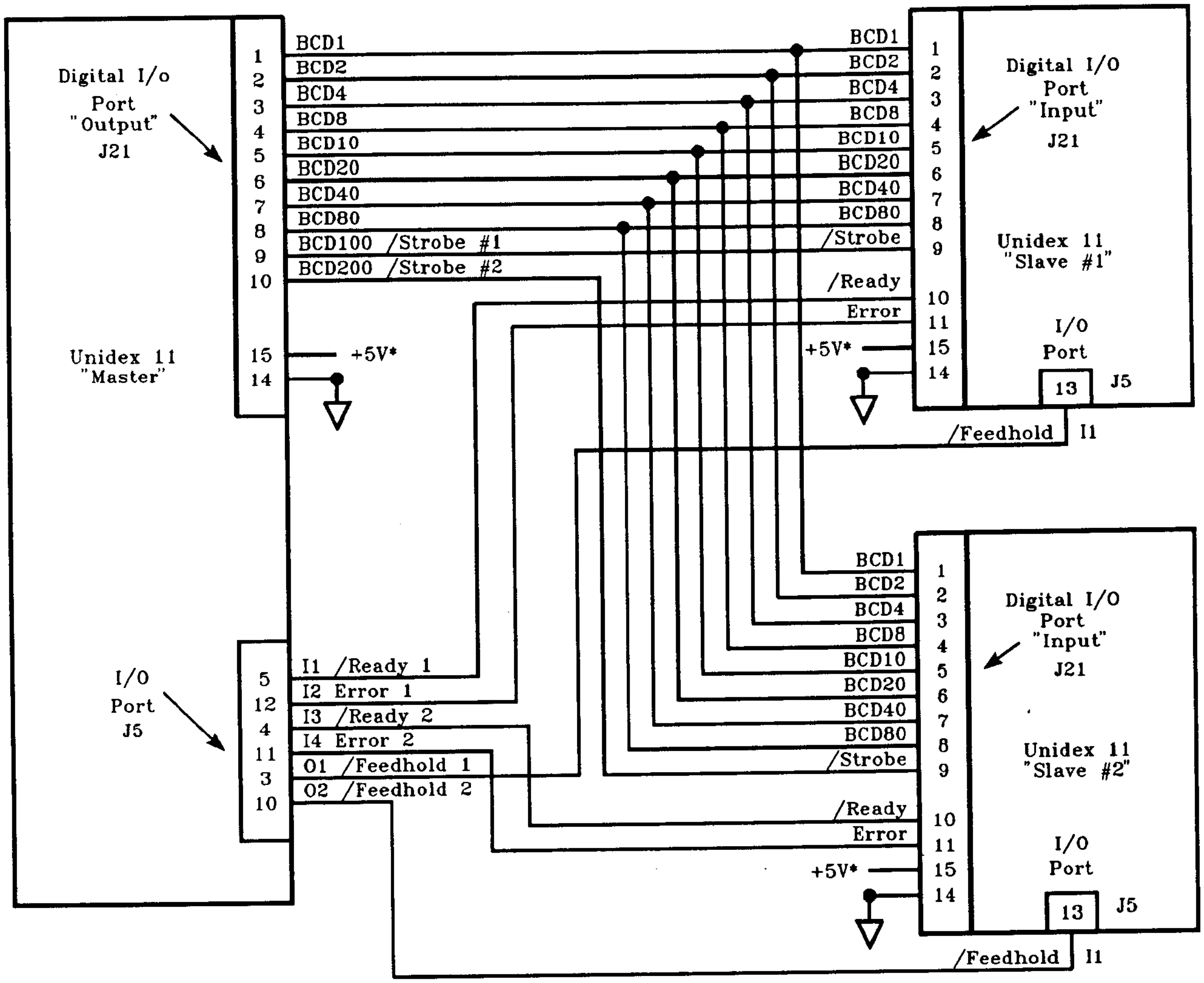
SECTION 3-1 MASTER/SLAVE EXAMPLE

Figure 3-1 shows an example of a master with two slave Unidex 11s. The parallel data bus (BCD 1 to BCD 80) is shared by both slave units. Individual pseudo strobes (BCD 100 and BCD 200) are used to select and initiate programs in either of the slaves. The Ready and Error outputs of the slaves are connected to the inputs of the I/O channel of the master to insure that the slave status may be read. The feedhold input of each slave is connected to two of the outputs of the master unit to provide a method of halting program execution.

Slave programs are called and initiated by a series of commands from the master unit. Following is an example:

Command	Description
DIG. OUT 201 BCD	Slave #2, Program #1
DWELL .1 Sec	Debounce time "strobe"
DIG. OUT 101 BCD	Slave #1, Program #1
INPUT 0XXX	Wait for #1 Ready, Program done
COND GO TO # for X1XX	Go to # if #1 Error present

The first 3 command examples are used to call and initiate a slave program. The last 2 commands will wait for program completion and test for error.



* Pin 15 of the Digital I/O Port was common, prior to ECN0680 and/or Rev. B of the Unidex 11 Option Board (Board Number 690D1320).

NOTE: Operation of the I/O Port is discussed in the *Unidex 11 Motion Controller Programming Manual.*

Figure 3-1: Example of Master/ 2 Slaves

CHAPTER 4: ELECTRICAL SPECIFICATIONS

The DIO option interfaces directly to a Rockwell #R6522AP VIA (versatile interface adapter).

Electrical specifications for input lines are listed below:

Maximum input voltage = 0 to +5.0 VDC
Maximum input low voltage = +.4 VDC
Maximum input low current (Vin = +.4 VDC) = -1.6mA
Minimum input high voltage = +2.4 VDC
Maximum input high current (Vin = +2.4 VDC) = +.1mA

These inputs are designed to accept TTL, +5 CMOS open collector drivers.

NOTE: These inputs are unbuffered. Dry contact inputs are *not* recommended. Interconnection wiring greater than 5 feet is not recommended.

Electrical Specifications for Output Lines are listed below:

Minimum output high voltage = 2.4 volts (Iload = +100μA)
Maximum output low voltage = .4 volts (Iload = 1.6mA)

NOTE: These outputs are unbuffered. Driving interface is limited to two TTL or 5 volt CMOS only. Interconnection wiring greater than 5 feed is not recommended.

CHAPTER 5: TROUBLESHOOTING

SECTION 5-1 SOFTWARE MALFUNCTIONS

Software malfunctions are problems which are discovered and displayed as error messages by Unidex 11. The error messages are described in the *Unidex 11 Motion Controller Programming Manual*, Part II.

SECTION 5-2 HARDWARE MALFUNCTIONS

PROBLEM	POSSIBLE CAUSE & SOLUTION
Digital I/O Bus Completely Dead	Ascertain that the Digital I/O Bus is wired for the option being used (see section 1-1). Check set up and operational procedure (chapter 2).

PROBLEM	POSSIBLE CAUSE & SOLUTION
Call-Program Does Not Work	Check set up and call procedure (section 2-3 B of this part of the manual). Verify that strobe is at least 15 milliseconds in duration and a negative pulse.

PROBLEM	POSSIBLE CAUSE & SOLUTION
Call-Program Does Not Work At Power Up	Parallel Port must be set for Input mode, the Battery Back-up option must be present, and the Strobe must be sent to enable the parallel port input mode before the call program will work.

Appendix 1
THE TWO DIGIT THUMBWHEEL

The Two Digit Thumbwheel Switch Assembly is a simple control device which runs programs already stored in Unidex 11 with the Digital I/O Option. (Set for the input mode – see figure AP1- 2.)

To operate the TDT, plug it into the Digital I/O connector located on the rear of Unidex 11 (see *Unidex 11 Motion Controller Hardware Manual*). Power up Unidex 11. Set the Thumbwheel Switches to read "00". Press "Strobe" once and release. The "Ready" LED will now be on. Unidex 11 is now ready to run programs called by the TDT.

Set the desired program number on the TDT, and press and release "Strobe".

The "Ready" LED will de-energize and Unidex 11 will execute the program if it is in memory. After executing the program, the "Ready" LED will re-energize.

If an error condition is generated because the program called is nonexistent, or it there is a run-time error in the program, Unidex 11 will energize the "Error" LED and the "Ready" LED simultaneously. A different program may be selected at this time in the above described manner.

To return control to the keyboard, set the program number to "00", then press and release "Strobe".

NOTE: For more details on running a program from an external device, refer to section 2-3, Input Mode, of this part of the manual.

An electrical diagram of the TDT Option is shown in figure AP1-1.

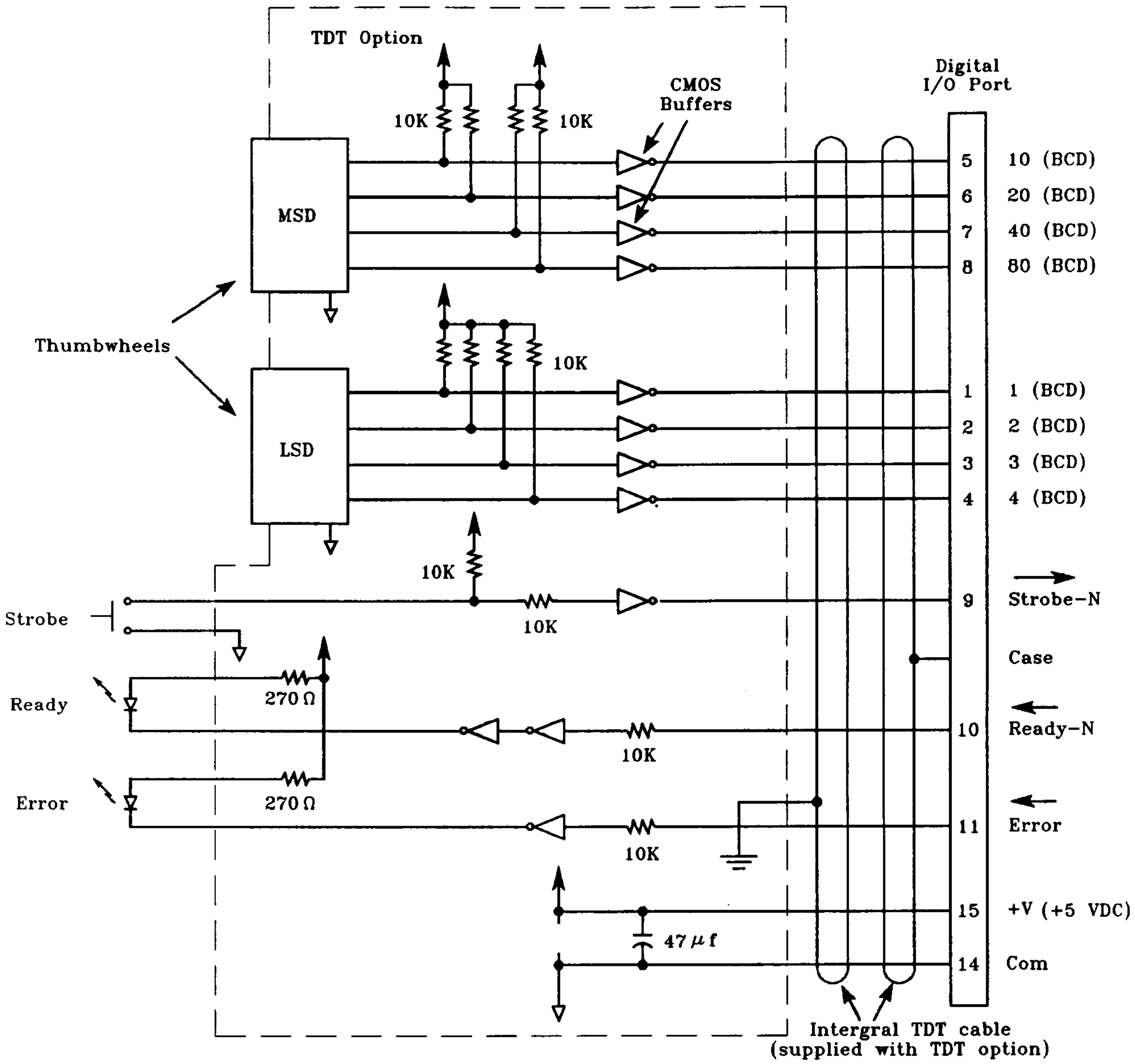


Figure AP1-1: Two Digit Thumbwheel Switch Schematic

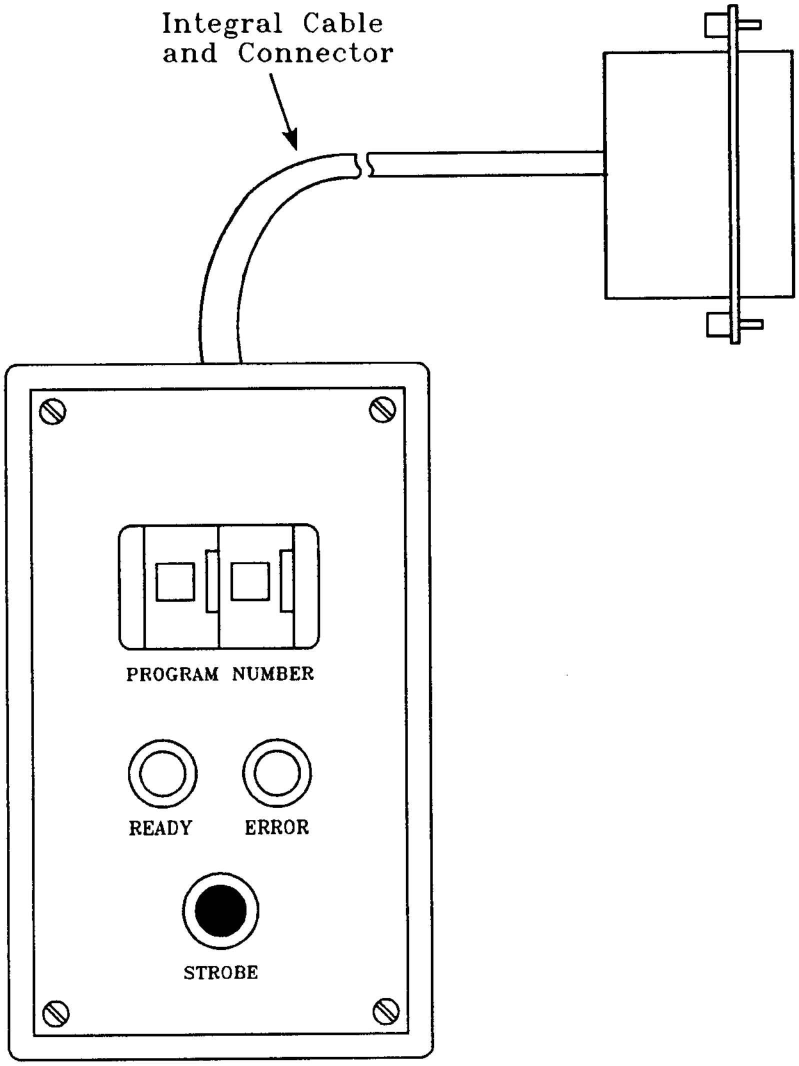


Figure AP1-2: Two Digit Thumbwheel Switch Assembly

Index

PART V:

HIGH SPEED BINARY INTERFACE

TABLE OF CONTENTS

CHAPTER 1: HIGH SPEED BINARY INTERFACE

SECTION 1-1 INTRODUCTION

The Unidex 11 High Speed Binary interface is an 8-bit parallel interface that permits communication speeds of up to 80K bytes/second, depending on the host computer. A PC-AT operating in the DMA mode with a Metrabyte PDMA-16 parallel interface board can achieve approximately 20K bytes / second. Unlike the RS-232 or the IEEE-488 interface, motion commands are given in a binary format that reduces the computational overhead time of UNIDEX 11 between moves.

CHAPTER 2: HARDWARE INTERFACE

The high speed binary interface is accessed through a 15 pin D-connector located on the rear of the Unidex 11 (see the *Unidex 11 Motion Controller Hardware Manual* for more information).

The pin configuration of this connector is as shown in figure 6-1 of this part of the manual, which also shows the required hardware to interface the Unidex 11 to a Metra-Byte PDMA-16 Digital I/O Board. None of the circuit shown is required for non-DMA type operation, although a shielded cable and pull-up resistors are highly recommended in all cases. All signal inputs and outputs are TTL compatible. The I/O structure of Unidex 11 hardware implementation is shown in figure 6-2.

A two line handshake is used for data transfers as shown in figure 6-3. When writing to Unidex 11, the host asserts the Host Ready (HOST RDY) line LOW after setting up data on the data lines. This falling edge initiates the auto-handshake mode of the 6522 VIA, latching the data. UNIDEX 11 then asserts Unidex 11 Ready (U11 RDY) line LOW after accepting the data. The host may now write the next byte in the same manner.

When reading from Unidex 11, the host negates the Host Ready (HOST RDY) line to a HIGH and waits for the U11 RDY line to be asserted LOW and, after accepting the data, asserts HOST RDY line LOW.

CHAPTER 3: PROGRAMMING

SECTION 3-1 PROGRAMMING THE BINARY INTERFACE

Upon power up, the Unidex 11 is in the "Local with Communication Enabled" state.

If the High Speed Binary Interface is activated by sending the character "?", the display will change to:

```
        PARALLEL INTERFACE
      COMMUNICATION ACTIVE
```

If a key on the keyboard is pressed, the communication modes will be locked out. Once a key has been pressed, the parallel interface must be enabled by selecting "COMM ENAB" from the Unidex 11's menu screens.

```
1.  DIG. OUT       3.  COMM ENAB
2.  ACL/DCL        4.  PRINT
```

Pressing key #3 will take you to the following screen:

```
1.  RS-232
2.  PARALLEL/BINARY
```

Press key #2, Parallel/Binary, and Unidex 11 is ready to communicate through the parallel port. You will see:

```
PARALLEL INTERFACE
COMMUNICATION ENABLED
```

The parallel interface has two modes, the High Speed Binary (HSB) mode, and the Programmable Logic Controller (PLC) mode. Unidex 11 may be set to these modes of operation by writing the following ASCII codes. Writing to Unidex 11 is done by placing the data on the data lines and asserting HOST RDY low.

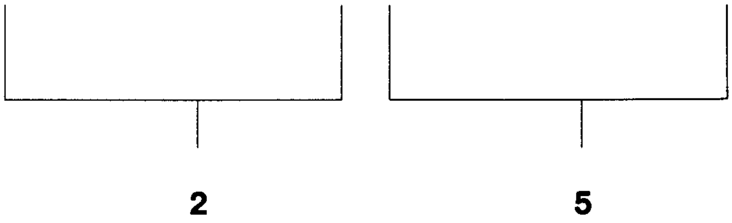
	CHARACTER	**ASCII**	**HEX**
HSB Mode	?	63	3F
PLC Mode	=	61	3D

In the HSB mode the Unidex 11 tracking display will not be updated in order to minimize overhead processing time, and the lo/hi current control will be set to the high current mode.

SECTION 3-2 PROGRAMMABLE LOGIC CONTROLLER

The Programmable Logic Controller (PLC) mode allows a program contained within the Unidex 11's memory to be executed by writing the BCD program number into the Unidex 11. The program number is placed on the data lines D7 - D0, and HOST RDY is asserted low. Example (Program #25):

D7	D6	D5	D4	D3	D2	D1	DO
0	0	1	0	0	1	0	1



Upon completion of the program the IN-POS line will go HIGH.

To return control to the keyboard, Unidex 11 must be set to the Local/Comm. enabled mode. This may be done only by setting Unidex 11 to the HSB mode first and then setting it to the Local/Comm. Enable state. The character sequence to be written is: "? L", or 63 76 (ASCII), or 3F 4C (Hex).

CHAPTER 4: HIGH SPEED BINARY MODE

This mode allows all of the RS-232 immediate mode commands to be executed (see chapter 7 of this part of the manual). It also allows the position and status to be read from the Unidex 11, the Unidex 11's LCD tracking display to be updated, the joystick to be enabled and disabled, and the remote clock and direction to be tracked (see chapter 5 of this part of the manual).

The commands are in a binary language which has a very restricted syntax. A motion command has a predefined number of bytes based upon the axis to be moved, whether the command is a distance only command, a feedrate only command (axis free run), or a feedrate and distance command. The first byte in each command block must specify the number of bytes in that command block that will follow. This is required, due to the fact that Unidex 11 is waiting for the specified number of bytes, and is not relying on microprocessor interrupt service routines, which would greatly increase the overall block processing time. A motion command has the following syntax:

First byte : Number of bytes that follow + 80 hex

Second byte : The second byte may be one of the following for a motion block:

25 (hex) : U axis feedrate only CW (3 bytes follow)
35 (hex) : U axis feedrate only CCW (3 bytes follow)
55 (hex) : Uaxis feedrate and distance (7 bytes follow)
75 (hex) : U axis distance only (4 bytes follow)

26 (hex) : V axis feedrate only CW (3 bytes follow)

36 (hex) : V axis feedrate only CCW (3 bytes follow)
56 (hex) : V axis feedrate and distance (7 bytes follow)
76 (hex) : V axis distance only (4 bytes follow)

28 (hex) : X axis feedrate only CW (3 bytes follow)
38 (hex) : X axis feedrate only CCW (3 bytes follow)
58 (hex) : Xaxis feedrate and distance (7 bytes follow)
78 (hex) : X axis distance only (4 bytes follow)

29 (hex) : Y axis feedrate only CW (3 bytes follow)
39 (hex) : Y axis feedrate only CCW (3 bytes follow)
59 (hex) : Yaxis feedrate and distance (7 bytes follow)
79 (hex) : Y axis distance only (4 bytes follow)

If the distance is to be specified, it must be a 4 byte word representing the move in steps; as a two's complement binary number. If the feedrate is to be specified it must be a 3 byte word representing the inverse of the clock rate in microseconds.

The following motion block:

X - 100,000 steps at 50,000 steps/sec (20 μsec)	8 bytes
Y - 5,000stepsat 1,000steps/sec(1000μsec)	8 bytes
U - Free run CCW at 10,000 steps/sec (100 μsec)	<u>4 bytes</u>
Total =	20 bytes

would be represented in HSB language as (shown in hex):

```
94  58 00 00 14   00 01 86 A0   59   00 03 E8   00 00 13 88   35   00 00 64   54
 #  X    FDRT.      DISTANCE     Y     FDRT.      DISTANCE     U     FDRT.     T
```

Unidex 11 will begin pre-move computations immediately after receiving the number of bytes specified. Upon completion of these computations, Unidex 11 will wait for the "T" or trigger character, shown above as 54 hex. If a feedrate of zero (00 00 00) is specified, it will stop that axis from free running.

The IN-POS line will be taken to the logical LOW state as the trigger character is being received. This means that the IN-POS line will go LOW as the trigger character is received but before the U11 RDY line is asserted, indicating that the byte has been accepted. At the end of the move the IN-POS line will then be asserted HIGH.

During this period, when the IN-POS line is LOW indicating that a move is in progress, no other motion blocks may be transferred to the Unidex 11. This can be clearly seen by the logic used for DMA transfers to the Unidex 11, as shown in Figure 6-1. The motion in progress may be interrupted and aborted by writing to Unidex 11 as shown below:

MOTION INTERRUPT

D7	D6	D5	D4	D3	D2	D1	D0	
0	1	1	1	1	1	1	1	System Reset
0	0	0	0	0	0	0	1	Stop X
0	0	0	0	0	0	1	0	Stop Y
0	0	0	0	0	1	0	0	Stop U
0	0	0	0	1	0	0	0	Stop V
0	0	0	0	1	1	1	1	Stop All Axes
0	0	1	0	1	0	1	0	Stop Y And V And Update Position Rgstrs

D7	D6	D5	D4	D3	D2	D1	D0	
0	1	0	0	0	1	0	0	Stop U And Reset All Drives
0	1	1	0	1	1	1	1	Stop All Axes, Reset Drives And Update Position Registers

Bit 5 :Will update the Position registers.

Bit 6 : Will reset all drives.

CHAPTER 5: COMMUNICATING WITH UNIDEX 11

In the High Speed Binary mode, the HOST may also read data from Unidex 11. A 3-byte status may be read by writing to Unidex 11 the ASCII character "S" (53 hex or 83 decimal) or ASCII character "s" (73 hex or 115 decimal). Axis positions may be read by writing to the Unidex 11 the ASCII character "P" (50 hex or 80 decimal) or "p" (70 hex or 112 decimal).

A. STATUS

To request status data from the Unidex 11 the HOST must write ASCII "S" (53 hex or 83 decimal), or ASCII "s" (73 hex or 115 decimal). Unidex 11 will return the status bytes in the same format as that in which it has been receiving information. The first byte will be 83 hex (131 decimal) indicating that three bytes will follow. These will be the three status bytes (byte 1, byte 2, byte 3). No "T" will follow. See Chapter 8 in this Part V of this manual for a description of these bytes. See below for Handshake sequence of "S" or "s" entries.

B. AXIS POSITION

To request axis positions from the Unidex 11, the HOST must write ASCII "P" (50 hex or 80 decimal) or ASCII "p" (70 hex or 112 decimal) and negate the HOSTRDY. The Unidex 11 will then output the bytes one at a time, asserting U11RDY each time.

The format of data is as follows :

First byte : Number of bytes + 128 decimal (80 hex)
Second byte : The second byte may be one of the following:

55 (hex) :(U axis) followed by 4 bytes of position
56 (hex) :(V axis) followed by 4 bytes of position
58 (hex) :(X axis) followed by 4 bytes of position
59 (hex) :(Y axis) followed by 4 bytes of position

Reading the position after the move executed in the previous chapter would result in the following data from Unidex 11:

```
8F 58 00 01 86 A0 59 00 00 13 88 55 00 00 80 00
#   X POSITION     Y POSITION     U POSITION
```

Unidex 11 sends position values for all valid axes. Free-run axes position values are updated only every 32768 steps. Status byte #3 will correctly indicate U-Axis in motion.

See below for Handshake sequence of "P" or "p" entries.

HANDSHAKING SEQUENCE FOR HOST ENTRY "S" OR "P"
(See Figure 6-3)

	HOST		UNIDEX 11
1.	Places"S"or"P" on the data bus.		
2.	Asserts HOSTRDY low.	→	Reads "S"or"P" and asserts U11RDY low.
3.	Changes Data Port to INPUT.		Waits for HOSTRDY high.
4.	Negates HOSTRDY high.	→	Changes Data Port to OUTPUT
5.	Waits for IN-POS to go low.		Negates U11RDY high.
		←	Asserts IN-POS low.
6.	Waits for U11RDY to go low to read data		Places first byte of data on the bus.
		←	Asserts U11RDY low
7.	Reads Data Byte.		Waits for HOSTRDY low transition.
8.	Asserts HOSTRDY low.	→	Negates U11RDY high.
9.	Negates HOSTRDY high.		

Steps 6 through 9 are repeated for each byte, including the last byte (refer to Figure 6-3 in Chapter 6).

		←	Asserts U11RDY low. Negates IN-POS high.
11.	Changes Data Port to OUTPUT.		

Host can now send data to Unidex 11.

HAND SHAKING SEQUENCE FOR HOST ENTRY "s" OR "p" (See Figure 6-4)

	HOST		UNIDEX 11
1.	Places"s"or"p" on the Data Port		
2.	(HOSTRDY low when data valid)	→	Reads "s"or"p" and asserts U11RDY low.
3.	Configures DMA interface for reading data. Sets AUX3 on MBI to enable DMA transfer		Changes Data Port to OUTPUT, Negates U11RDY high
4.	Waits for U11RDY to go low	→	Places data byte on Port ad asserts U11RDY low
5.	Negates HOSTRDY high.		
6.	Reads data byte		Waits for HOSTRDY low
7.	(HOSTRDY low when data read)	→	Negates U11RDY high.

Steps 4 through 7 are repeated for each byte of data including the last byte (refer to Figure 6-4 in Chapter 6).

8.	Configures DMA interface for writing data		Changes Data Port to INPUT.

C. TRACKING DISPLAY

To update Unidex 11's LCD tracking display to the current absolute position, the host must write the character "D" (44 hex or 68 decimal) to Unidex 11. Unidex 11 will then assent U11RDY true, and update its display.

Marker status for each axis will also be displayed on the Tracking Display. When an axis is at the marker (once per revolution for a rotary encoder; one home marker for a linear encoder) and the marker signal is active, the character "m" is displayed to the left of the axis name.

D. ENABLING/DISABLING JOYSTICK

The Joystick may be enabled and disabled in the High Speed Binary mode.

1. Enabling Joystick: command "J"

The character "J" (decimal is 74 and hex is 4A) puts Unidex 11 in the Joystick mode. The LCD displays change to the joystick tracking mode. The joystick is now active and the X and Y axes may be moved. If Unidex 11 cannot enter the joystick mode, the ERROR LINE along with the IN-POS LINE is asserted high.

Button A on the joystick may be pressed to toggle between X/Y and U/V axis pairs. Button B on the joystick may be pressed to toggle between normal and slow joystick clock rate. The ratio between the two rates is 64.

In the joystick mode, Button C may be pressed to call the attention of the computer. When Button C is pressed, the joystick is disabled and absolute position registers updated. Unidex 11 then asserts the IN-POS LINE high, to indicate to the computer that button C has been pressed. The joystick must now be disabled before any command such as "P" or "D" can be executed.

If an axis runs into a limit while using the joystick, Unidex 11 flashes and beeps to indicate the state. The second byte of the status (Runtime Error Status) is internally modified to reflect this limit. The status bit is cleared only when a new command is executed. Reading status immediately after the joystick mode, the limit bits will be set if the axis ran into a limit at any time during the joystick mode.

The only command that Unidex 11 will accept and respond to in the joystick mode is the character "K" to disable the joystick.

2. *Disable Joystick: command "K"*

The character "K" (decimal is 75 and hex is 4B) may be written to cancel the joystick mode. Unidex 11 disables the joystick, updates the absolute position registers, displays the screen:

```
PARALLEL INTERFACE
COMMUNICATION ACTIVE
```

and is ready to receive further commands.

E. ENABLE/DISABLE REMOTE CLOCK AND DIRECTION

Character "R" (decimal is 82 and hex is 52) sets up Unidex 11 to track remote clock and direction signals that may be applied to inputs provided on the rear panel connectors (see chapter 2 of the *Unidex 11 Motion Controller Hardware Manual*).

The LCD display changes to the remote tracking mode. Axis limits are disabled from interrupting in this mode. Any limit condition generated while in remote mode will not be detected by reading the status. However, the limits are still functional and will prevent overtravel. The limit signals are available for user interface on the rear panel of the Unidex 11 (see the *Unidex 11 Motion Controller Hardware Manual*).

Command "K" (decimal 75 and hex 4B) is used to exit this mode. Unidex 11 disables the remote clock and direction inputs and updates the position registers before returning to receive further commands.

The only command that Unidex 11 will accept and respond to in the remote mode is the character "K", in order to disable the mode.

CHAPTER 6: ERROR AND IN-POSITION LINES

The error line is set high to indicate any run-time error in the PLC mode. In the High Speed Binary mode, the error is set if any axis should run into a limit.

The IN-POS line indicates command complete status and the Unidex 11 is ready for the next index or program.

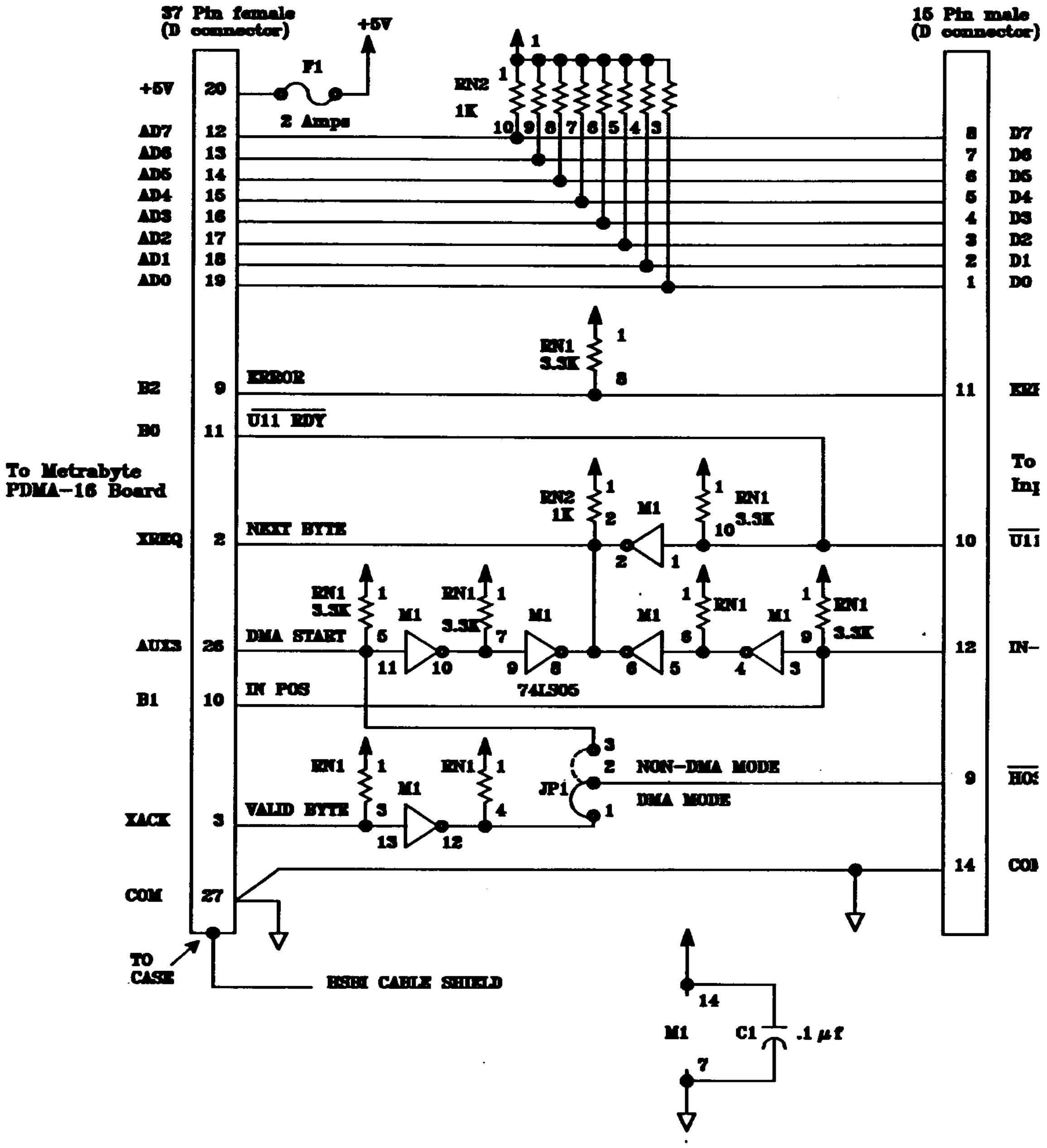
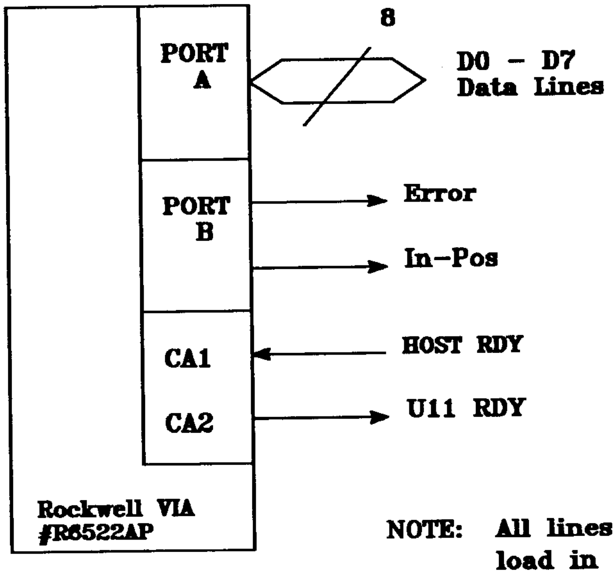


Figure 6-1: MBI (Metra-Byte Interface)Adapter Schematic

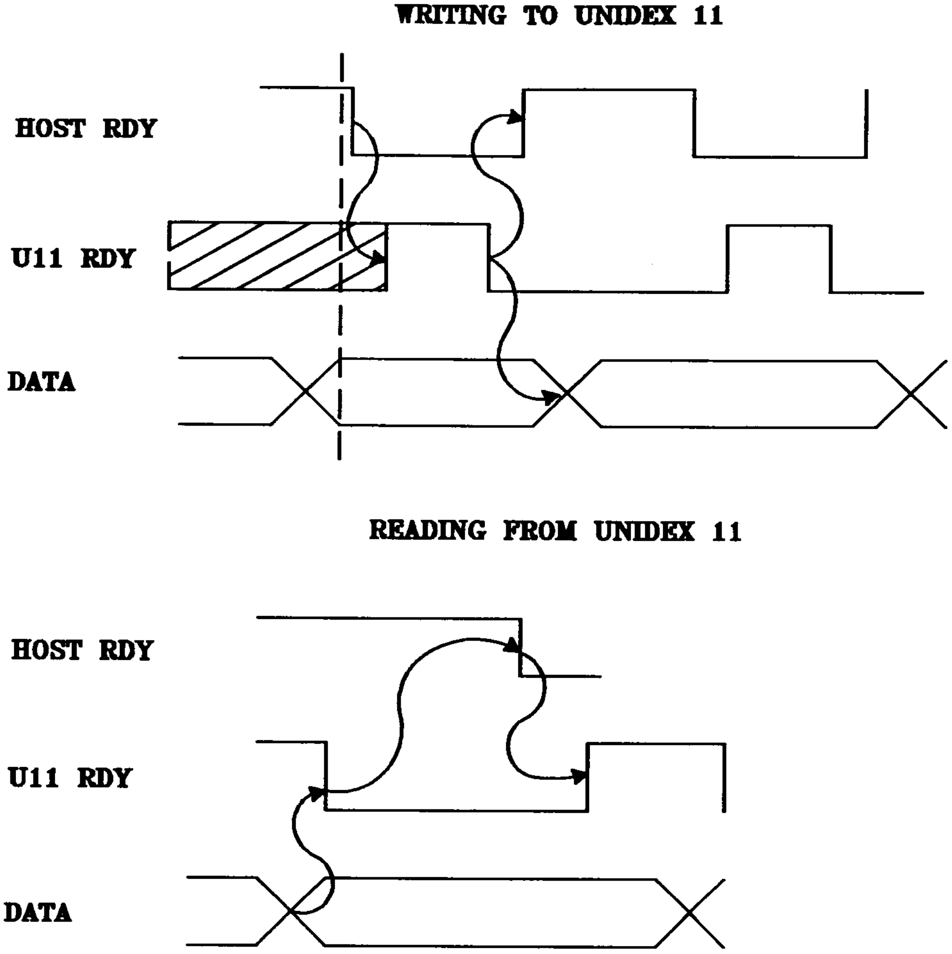


NOTE: All lines represent 1 TTL load in the input state, and will drive 1 TTL load in the output state.

ALL DATA IS TRANSFERRED IN THE AUTO HANDSHAKE MODE

In the DMA (high speed transfer) mode, approximately 5 feet of cable may be used between the host computer and Unidex 11 while maintaining reliable operation.

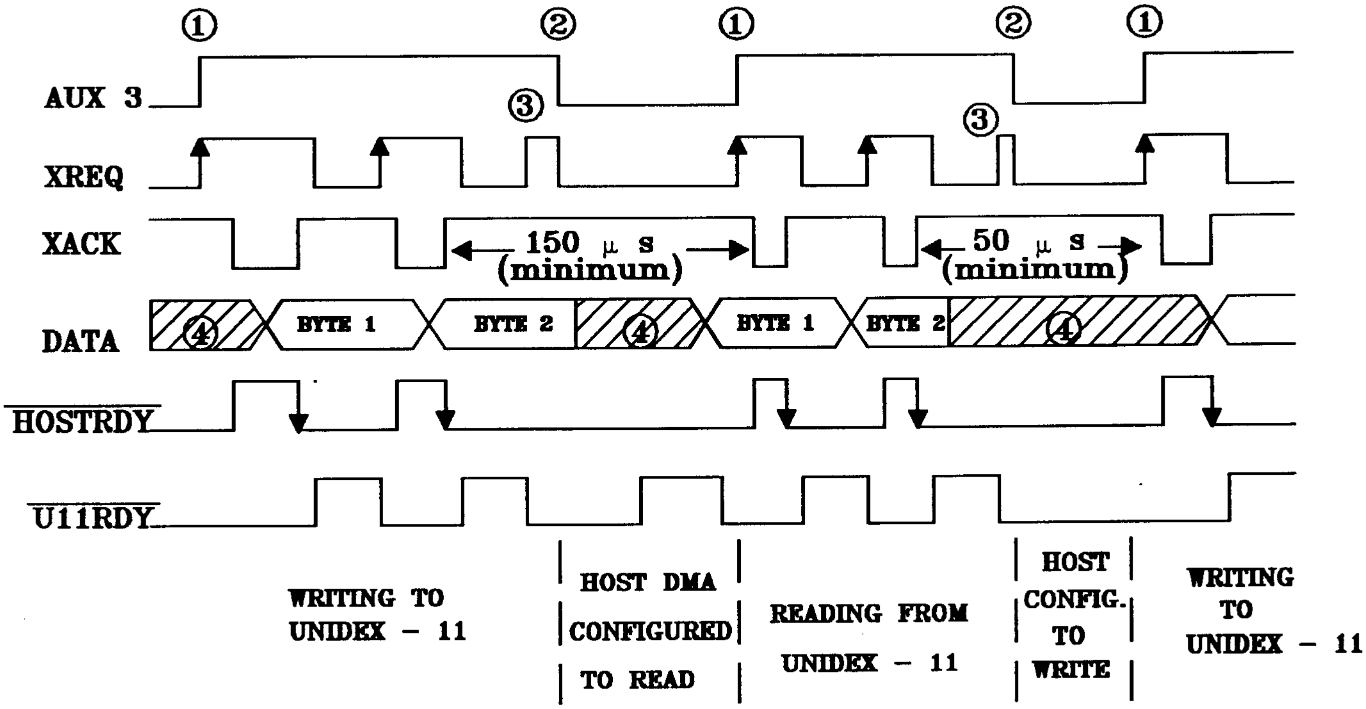
Figure 6-2: HSB I/O Structure



WRITING TO UNIDEX 11

READING FROM UNIDEX 11

Figure 6-1: MBI (Metra-Byte Interface)Adapter Schematic



① AUX3 SET HIGH TO START TRANSFER

② AUX3 SET LOW TO RE-CONFIGURE DMA

③ EXTRA "XREQ" DUE TO THE TIME DELAY BETWEEN $\overline{\text{U11RDY}}$ & AUX3

④ DON'T CARE DATA

Figure 6-4: HSB-DMA Handshaking

CHAPTER 7: HSB COMMAND FORMAT

Each of the commands shown in this chapter will show the number of bytes, and the "T" command as part of the High Speed Binary Command Format.

1. Dwell command

Byte 1 = 85 hex (133 decimal)
Byte 2 = 02 hex (2 decimal)
Byte 3 - 6, is the dwell time in microseconds. Where byte 3 is the most significant byte and byte 6 is the least significant byte.
Byte 7 = 54 hex (84 decimal) "T" trigger byte

For example, the following command would result in a dwell of 1 second:

85 02 00 0F 42 40 54

2. Home

Byte 1 = 82 hex (130 decimal)
Byte 2 = 03 hex (3 decimal)
Byte 3 = Where:

D7	D6	D5	D4	D3	D2	D1	D0
V	U	Y	X	0	0	0	0

Byte 4 = 54 hex (84 decimal) so H XY, would be 82 03 30 54 (hex) 130 03 48 84 (decimal)

3. Output

Byte 1 = 82 hex (130 decimal)
Byte 2 = 04 hex (4 decimal)
Byte 3 where:

D7	D6	D5	D4	D3	D2	D1	D0
4	3	2	1	4	3	2	1

identifies the bits that correspond to each output. The least significant nibble (D0 - D3) indicates the bit to set or reset. The most significant nibble (D4 - D7) indicates the logic value to which the bit should be set.

Byte 4 = 54 hex (84 decimal)

4. Input

Byte 1 = 82 hex (130 decimal)
Byte 2 = 05 hex (5 decimal)
Byte 3 where:

D7	D6	D5	D4	D3	D2	D1	D0
4	3	2	1	4	3	2	1

identifies the bits corresponding to each input. The least significant nibble (D0 - D3) indicates the inputs which are active. The most significant nibble (D4 - D7) indicates if inputs are active high or active low.

Byte 4 = 54 hex (84 decimal)

5. Start axis

Byte 1 = 82 hex (130 decimal)
Byte 2 = 0B hex (11 decimal)
Byte 3 = where:

D7	D6	D5	D4	D3	D2	D1	D0
V	U	Y	X	0	0	0	0

The most significant nibble indicates the axis that is to start free - runnig. The corresponding bits are set to 1.

Byte 4 = 54 hex (84 decimal)

6. Stop axis

Byte 1 = 82 hex (130 decimal)
Byte 2 = 0C hex (12 decimal)
Byte 3 = where:

D7	D6	D5	D4	D3	D2	D1	D0
V	U	Y	X	0	0	0	0

The most significant nibble indicates the axis to stop free - running. The corresponding bits are set to 1.

Byte 4 = 54 hex (84 decimal)

7. Load position registers

Byte 1 = 92 HEX (146 decimal)
Byte 2 = 0E hex (14 decimal)
Byte 3 = bits 0-3 are "0", bits 4-7 are as shown:

bit 4 = 1 if X axis is to be loaded
bit 5 = 1 if Y axis is to be loaded
bit 6 = 1 if U axis is to be loaded
bit 7 = 1 if V axis is to be loaded

The respective bit is 0 if the axis is not to be loaded.

Bytes 4 through 19: 4 bytes per axis represent the absolute position values to be loaded. A "0" in byte 3 will ignore the respective axis.

Byte 20 = 54 hex (84 decimal)

For example, to load X axis with 100,000 and V axis with 5000, the following byte sequence must be sent to Unidex 11.

```
92 0E 90 00 01 86 A0 XX XX XX XX XX XX XX XX 00 00 13 88 54
 X axis                  Y & U  don't care          V axis
```

8. Incremental mode

IN - 81 10 54 (hex)
129 16 84 (decimal)

9. Absolute mode

AB - 81 11 54 (hex)
129 17 84 (decimal)

10. Beeper on

BN - 81 13 54 (hex)
129 19 84 (decimal)

11. Beeper off

BF - 81 12 54 (hex)
129 18 84 (decimal)

12. Corner - rounding mode

CO - 81 1B 54 (hex)
129 27 84 (decimal)

13. Non corner - rounding mode

NC - 81 1C 54 (hex)
129 28 84 (decimal)

14. Accel/decel control

Byte 1 = 83 hex (131 decimal)
Byte 2 = 1D hex (29 decimal)
Byte 3 = MSB of ACL/DCL time (X4)
Byte 4 = LSB of ACL/DCL time (X4)
Byte 5 = 54 hex (84 decimal) - "T" trigger

NOTE: The ACL/DCL time in milliseconds is multiplied by 4 and then converted to binary.

CHAPTER 8: STATUS BYTES

BYTE 1: STATUS BYTE

	Zero	One
BIT 0	Incremental mode	Absolute mode
BIT 1	Not running a program	Running a program
BIT 2	Block run mode	Auto run mode
BIT 3	Non-corner rounding mode	Corner rounding mode
BIT 4	Communication disable	Communication enable
BIT 5	Inactive - Not executing a command in Immed. mode	Active - Executing a command in Immed. mode
BIT 6	No service request signal sent	Service request signal sent - waiting for "Q"
BIT 7	No errors detected	Error detected

BYTE 2: RUNTIME ERROR STATUS

	Zero	One
BIT 0	X axis not in limit	X axis in limit
BIT 1	Y axis not in limit	Y axis in limit
BIT 2	U axis not in limit	U axis in limit
BIT 3	V axis not in limit	V axis in limit
BIT 4	No illegal byte in memory	Illegal byte in memory
BIT 5	Program number valid	Invalid program called out for run

BIT 6	Memory not clear	No programs in memory (memory clear)
BIT 7	No user memory check sum error	User memory check sum error

BYTE 3: MOTION STATUS

	Zero	One
BIT 0	X axis not in motion	X axis in motion
BIT 1	Y axis not in motion	Y axis in motion
BIT 2	U axis not in motion	U axis in motion
BIT 3	V axis not in motion	V axis in motion
BIT 4	X axis not at marker	X axis at marker
BIT 5	Y axis not at marker	Y axis at marker
BIT 6	U axis not at marker	U axis at marker
BIT 7	V axis not at marker	V axis at marker

CHAPTER 9: MBI (METRA-BYTE INTERFACE)

This section describes the hardware (MBI option) required to interface a Metrabyte PDMA-16 digital I/O board to the Unidex 11 through the High Speed Binary Interface (see figure 6-1).

For proper shielding, the MBI mounting screws must be securely fastened to the computer's interface board.

Unidex 11 is capable of data transfer rates of up to 80K bytes per second. High data transfer rates can most easily be achieved by Direct Memory Access (DMA) data transfers. This technique bypasses the host's processor and associated overhead time by use of a hardware controller, which is inherently fast. The DMA controller will transfer commands from the host's RAM to the Unidex controller in the background, while the host is occupied with another job. This achieves higher data transfer rates with a minimal loss of speed to the foreground's program execution.

The DMA controller circuit requires a rising edge to initiate a DMA transfer. However, the DMA controller must be initialized before receiving the transfer request (XREQ). This is not possible with the host and Unidex being powered up at the same time. Therefore, a provision has been made in the MBI to allow the generation of a rising edge to the XREQ to initiate a DMA transfer.

To initiate a DMA transfer, the DMA controller must first be initialized and the Unidex must be in position (IN POS) and ready (U11-RDY) for commands. The DMA start line (figure 6-1 of this part of the manual) must then change from a logic low to a logic high state, creating the XREQ rising edge, and enabling the DMA transfer. The

DMA controller may be monitored for completion, and the process may be repeated.

The MBI option may also be used in the non-DMA mode of operation (which does not require a hardware interface) by changing JP1 to position 2-3 (see figure 6-1). In this mode, the DMA start output becomes the Host Rdy input to the Unidex 11. This line is then negated after a valid byte has been placed on the data bus (timing diagrams in figure 6-3.) The U11 RDY line is then asserted true after receiving the data byte. This mode will not normally be used because of its low throughput.

Located below is a programming example written in Microsoft Quickbasic. The program will load a Basic Data File (Decimal) into memory and initiate a DMA transfer. It will then display its progress until completion. The first command in the data file must be decimal 63 to enable the HSB mode.

```
'
' HSB DMA TEST - AEROTECH INC., 1987
' The first command in the data file must be decimal 63 to enable the
HSB mode!
COLOR 14,1
CLS
BASEADD = &H380                 ' PDMA-16 I/O address
' NEXT LINE IS TO RESET KICK START BEFORE KICKING!
OUT BASEADD+3,0                 ' Set aux3 low here

DEF SEG = &H5000                ' This line will vary, based on
                                your memory
                                ' configuration
COUNT = 0                       ' Reset byte index
LOCATE 3,25:PRINT"DMA TEST"  LOCATE 7,3:INPUT"Enter file name
with decimal HSB commands to execute";FILE$
OPEN FILE$ FOR INPUT AS #1
```

```
LOOP:    IF EOF(1) THEN GOTO
         FILEND
        INPUT #1,BYTE                        ' Get byte
        POKE (COUNT),BYTE                    ' Stick in memory

        COUNT = COUNT + 1                    ' Increment index
        GOTO LOOP
FILEND: NT = COUNT                           ' Set # of transfers + 1

        --MERGE METRABYTES DMA2.BAS PROGRAM HERE--
        DELETE LINES 210 AND 280 AND SET VARIABLES AS YOUR SYS-
        TEM REQUIRES

'
' KICK START DMA TRANSFER HERE - SET AUX 3 HIGH HERE
OUT BASEADD+3,8                              ; SET AUX3 HIGH HERE
PRINT:PRINT:PRINT"DMA transfer running "
PRINT
PRINT NT;"Bytes to transfer"
GOTO DMATST
DMATST: XL = INP(3)                          ' GET DMA LOW BYTE COUNT
        XH = INP(3)                          ' GET DMA HIGH BYTE COUNT
        B = 256 * XH + XL                    ' FIND # OF BYTES TO GO
        IF B = 65535 THEN B=0
        LOCATE 15,3
        PRINT B;"Bytes left to go "
        IF B=0 THEN END ELSE                 ' DONE?
        GOTO DMATST
```

Index

PART VI:

PASSWORD

Table of Contents

CHAPTER 1: PASSWORD OPTION

This option allows the user to install a password in the system so that the Unidex 11 is protected from unauthorized editing of system user parameters and user programs.

Once protected, any function (keyboard or communication) that alters user parameters or user programs requires a password input to access that function. The password is a number up to 6 digits in length.

SECTION 1-1 PASSWORD INSTALLATION

A new Unidex 11, in which a password has not yet been installed, can be operated as a normal system, i.e., without a password. The user may choose to install a password in the Setup Mode. The procedure is as follows:

Select the Setup mode by pressing key #3 in the third main screen. Unidex 11 will now display:

```
CHANGE PASSWORD ?
1. Yes     2. No
```

Select "Yes" by pressing key #1. Now the display changes to:

```
ENTER NEW PASSWORD
(up to 6 dig) ......
```

The first dot on the dotted line will be flashing. Any numeral key may be pressed to enter the first digit of the password. When the first digit is typed in, the flashing cursor moves to the next position. For security purposes, the digit itself is not displayed.

A non-numeral key (+/-, SELECT, INSERT, DELETE, BACK, ENTER) will terminate the entry process. When the desired password is typed in, press any non-numeral key to enter it. A 6 digit password will automatically be entered on entry of the 6th digit. The Unidex 11 will now require you to enter this same password once more, for confirmation. The previous message is repeated and the password must be entered in the same manner.

The password entered the second time is compared to the first entry and if they are not identical, Unidex 11 will required you to enter it again. The entry of a new password (or installation of one) is not complete until the same password is entered twice in succession.

CHAPTER 2: UNIDEX 11 PASSWORD OPERATION

When a keyboard function such as RUN SPEED is selected from the menu on the first main screen, Unidex 11 displays the instruction:

```
ENTER PASSWORD
(up to 6 dig) : ......
```

Now type in the password digits and enter them. If the correct password was entered, Unidex 11 will display the screen for RUN SPEED entry. Values may be entered as desired.

If the password entered does not compare with the stored password, Unidex 11 indicates so by displaying:

```
** PASSWORD INCORRECT **
1. Retry    2. Quit
```

Pressing key #1 will give you another chance to enter the password. Pressing key #2 will return you to the first main screen (or the screen which initiated the password entry function).

SECTION 2-1 CHANGING THE PASSWORD

To change the password (to program a new password), select the Setup mode from the third main screen. The current password is required to enter the set up mode. Once in the set up mode, Unidex 11 displays the message:

CHANGE PASSWORD ?
1. Yes 1. No

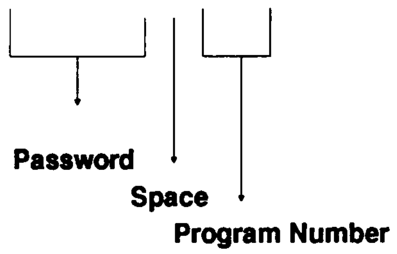
Press key #1 and proceed as described in the section on installation of a new password (chapter 1).

SECTION 2-2 COMPUTER CONTROL OF UNIDEX 11

With the password option, the password is required in the Edit mode to download a program or to delete a program or to clear the entire user memory. To download a new program, the password must follow the command character "E" as described below.

The string:

"Epppppp 24*"



will download the subsequent data strings into program number 24. The six digit password in this case is terminated by a Space. The space is always required as a delimiter.

To delete a program : "Epppp $24*"
To clear all memory : "Eppppp $00*"

SECTION 2-3 PASSWORD CORRUPTION

In the unlikely event of an alteration of the user memory, Unidex 11, upon power-up, will display:

```
USER MEMORY CHECKSUM ERR
Press SELECT for a MENU
```

It is recommended that the user performs a LOAD DEFAULT operation to set the system parameters to their default values stored in the firmware. The LOAD DEFAULT function is available in the DIRECTORY mode from the fourth main screen.

The password may or may not have been corrupted. To verify, invoke the password entry function by selecting the Setup mode. If the password has been corrupted, Unidex 11 will display:

```
** PASSWORD CORRUPTED **
Default PSWD : ......
```

The default password stored in system firmware is the number "012345". Enter this password correctly and Unidex 11 will display:

```
ENTER NEW PASSWORD
(up to 6 dig) : ......
```

The new password may be installed as described in the previous chapter.

If the default password is entered incorrectly, the display will show:

```
** PASSWORD INCORRECT **
1. Retry      2. Quit
```

Press key #1 to return to the screen requesting default password. Key #2 will return you to the third main screen.

To reduce the probability of a password corruption going undetected, Unidex 11 stores the logic complement of the password as well as the password itself. Whenever the password entry function is invoked, Unidex 11 verifies the password with its complement before requesting the user to enter it.

Index

SERVICE AND REPAIR

General repair of equipment consists entirely of solutions listed in the chapter on Troubleshooting in the *Unidex 11 Motion Controller Programming Manual* (chapter 4 of Part II), or the removal and replacement of the Translator or Control Board(s), should the need arise.

If under warranty, repairs of defective electrical components of the Control and Translator boards should not be attempted, since to do so would void the entire warranty.

If necessary, any on-site service should be performed by an experienced electronic technician, preferably trained by Aerotech, Inc.

SHIPMENT

The procedure for shipping equipment back to Aerotech, which is described below, pertains to warranty as well as non-warranty repairs.

1. Before shipping any equipment back to Aerotech, the person making the return must call ahead for a "Return Authorization Number".

2. The equipment being returned must be encased in a proper cushioning material and enclosed in a cardboard box.

Call for a "Return Authorization Number" if it is necessary to ship any part to the factory.

Warning: Damage due to improper packaging voids warranty!

Aerotech Sales and Service offices are listed on the following pages. For service and information, contact the office servicing your area.

AEROTECH, INC. SALES OFFICES

World Headquarters
AEROTECH, INC.

101 Zeta Drive
Pittsburgh, PA 15238

Phone (412) 963-7470
FAX (412) 963-7459
TWX (710) 795-3125

AEROTECH, CENTRAL-EAST
856 Cottonwood Drive
Monroeville, PA 15146

Phone (412) 373-4160
FAX (412) 373-4163
WV, western PA, western NY, eastern OH

AEROTECH WEST

Suite 217
7002 Moody Street
La Palma, CA 90623
Phone (213) 860-7470
FAX (213) 860-4639
AZ, southern CA

AEROTECH NORTHEAST
Executive Suite 120
270 Farmington Avenue
Farmington, CT 06032
Phone (203) 673-3330
or (203) 673-2503
FAX (203) 674-1536

MA, CT, VT, ME, RI, NH, eastern NY

AEROTECH SOUTHWEST
6001 Village Glen Drive
#3101
Dallas, TX 75206
Phone (214) 987-4556
FAX (214) 987-4706
TX, OK, LA, AR, CO, UT,MT, WY, ID, NM

AEROTECH MID-ATLANTIC
521 Kingwood Road
King of Prussia, PA 19406

Phone (215) 265-6446
FAX (215) 265-3566
MD, DC, DE, NJ, northern VA, eastern PA

AEROTECH CENTRAL-WEST

26791 Lake Vue Drive #8
Perrysburg, OH 43551
Phone (419) 874-3990
FAX (419) 874-4280
MI, IN, KY, western OH

AEROTECH NORTHWEST

444 Castro Street
Suite 400
Mountain View, CA 94041
Phone (415) 967-4996
FAX (415) 967-4998
northern CA, OR, WA, NV

AEROTECH SOUTH ATLANTIC
8804 Lomas Court
Raleigh, NC 27615
Phone (919) 848-1965
FAX (919) 848-3393
NC, TN, southern VA, AL, FL, GA, SC, MS

AEROTECH MIDWEST
PO Box 625
Dundee, IL 60118
Phone (312) 428-5440
FAX (312) 428-5471
IL, MO, KS, WI, MN, ND, SD, IA, NE

INTERNATIONAL SALES OFFICES

AEROTECH LTD.
3 Jupiter House, Calleva Park
Aldermaston
Berkshire RG7 4QW England
Phone (07356) 77274
TLX 847228
FAX (07356) 5022

AEROTECH GMBH
Neumeyerstrasse 90
8500 Nuernberg 10
West Germany
Phone (0911) 521031
TLX 622474
FAX (0911) 521235

AEROTECH AUSTRALASIA
224 Carr Street
Suite 7
Leederville 6007
Western Australia
Phone (619) 328-2540
FAX (619) 227-6670

INTERNATIONAL REPRESENTATIVES

BRASITEC
Rue Americo Brasiliense, 2069
Chacara Santo Antonio
Cep 04715 - Sao Paulo - SP
Brazil
Phone (5511) 523 4044
TLX 1130691 BRTC

Y. BEN MOSHE
PO Box 18125
Tel Aviv 61181
Israel
Phone (9723) 7515007
or (9723) 7513268
TLX 342436 BMS IL
FAX (9723) 727319

SIMCO
208, First Floor
Ajmeri Gate
New Delhi 110 006 India
Phone 652986
TLX 031-62176 HARS IN
FAX (9111) 510697

OPTIKON CORPORATION LTD.

410 Conestogo Road
Waterloo, Ontario
Canada, N2L 4E2
Phone 519-885-2551
FAX 519-885-4712

DONG DO TRADING CO. LTD.
Rm 903, Kwang Sung Bldg.
831-47 Youksamdong
Kangnam-Ku, Seoul, Korea
Phone (822) 556-2292
FAX (822) 556-2902
TLX 29734 DONG DO

TOKYO INSTRUMENTS INC.

Asahi-Seimei Bldg.
6-8-10 Nishikasai
Edogawa-Ku
Tokyo 134 Japan
Phone (813) 686-4711
FAX (813) 686 0831

HISCO (MALAYSIA) SDN.BHD.
1 Lorong SS13/6A
Subang Jaya Indust. Estate
47500 Petaling Jaya
Selanger, Malaysia
Phone (603) 733-4236
FAX (603) 733-6281
TLX 36226 HISCO MA

AEROTECH®

Warranty and Field Service Policy

Aerotech, Inc. warrants its products to be free from defects caused by faulty materials or poor workmanship for a minimum period of one year from date of shipment from Aerotech. Aerotech's liability is limited to replacing, repairing or issuing credit, at its option, for any products which are returned by the original purchaser during the warranty period. Aerotech makes no warranty that its products are fit for the use or purpose to which they may be put by the buyer, whether or not such use or purpose has been disclosed to Aerotech in specifications or drawings previously or subsequently provided, or whether or not Aerotech's products are specifically designed and/or manufactured for buyer's use or purpose. Aerotech's liability on any claim for loss or damage arising out of the sale, resale or use of any of its products shall in no event exceed the selling price of the unit.

Laser Product Warranty

Aerotech, Inc. warrants its laser products to the original purchaser for a minimum period of one year from date of shipment. This warranty covers defects in workmanship and material and is voided for all laser power supplies, plasma tubes and laser systems subject to electrical or physical abuse, tampering (such as opening the housing or removal of the serial tag) or improper operation as determined by Aerotech. This warranty is also voided for failure to comply with Aerotech's return procedures.

Return Products Procedure

Claims for shipment damage (evident or concealed) must be filed with the carrier by the buyer. Aerotech must be notified within (30) days of shipment of incorrect materials. No product may be returned, whether in warranty or out of warranty, without first obtaining approval from Aerotech. No credit will be given nor repairs made for products returned without such approval. Any returned product(s) must be accompanied by a return authorization number. The return authorization number may be obtained by calling an Aerotech service center. Products must be returned, prepaid, to an Aerotech service center (no C.O.D. or Collect Freight accepted). The status of any product returned later than (30) days after the issuance of a return authorization number will be subject to review.

Returned Product Warranty Determination

After Aerotech's examination, warranty or out-of-warranty status will be determined. If upon Aerotech's examination a warrantied defect exists, then the product(s) will be repaired at no charge and shipped, prepaid, back to the buyer. If the buyer desires an air freight return, the product(s) will be shipped collect. Warranty repairs do not extend the original warranty period.

Returned Product Non-Warranty Determination

After Aerotech's examination, the buyer shall be notified of the repair cost. At such time the buyer must issue a valid purchase order to cover the cost of the repair and freight, or authorize the product(s) to be shipped back as is, at the buyer's expense. Failure to obtain a purchase order number or approval within (30) days of notification will result in the product(s) being returned as is, at the buyer's expense. Repair work is warranted for (90) days from date of shipment. Replacement components are warranted for one year from date of shipment.

Rush Service

At times, the buyer may desire to expedite a repair. Regardless of warranty or out-of-warranty status, the buyer must issue a valid purchase order to cover the added rush service cost. Rush service is subject to Aerotech's approval.

On-Site Warranty Repair

If an Aerotech product cannot be made functional by telephone assistance or by sending and having the customer install replacement parts, and cannot be returned to the Aerotech service center for repair, and if Aerotech determines the problem could be warranty- related, then the following policy applies.

Aerotech will provide an on-site field service representative in a reasonable amount of time, provided that the customer issues a valid purchase order to Aerotech covering all transportation and subsistence costs. For warranty field repairs, the customer will not be charged for the cost of labor and material. If service is rendered at times other than normal work periods, then special service rates apply.

If during the on-site repair it is determined the problem is not warranty related, then the terms and conditions stated in the following "On-Site Non-Warranty Repair" section apply.

On-Site Non-Warranty Repair

If an Aerotech product cannot be made functional by telephone assistance or purchased replacement parts, and cannot be returned to the Aerotech service center for repair, then the following field service policy applies.

Aerotech will provide an on-site field service representative in a reasonable amount of time, provided that the customer issues a valid purchase order to Aerotech covering all transportation and subsistence costs and the prevailing labor cost, including travel time, necessary to complete the repair.