
UNIDEX™ 12
MOTION CONTROLLER
OPTIONS MANUAL
PN: EDU 112



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

Aerotech, Inc.
Rel. 1.0, March, 1990

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 12:

Up to four manuals may have been shipped with your Unidex 12 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 12 Motion Controller Programming Manual*
- *Unidex 12 Motion Controller Hardware Manual*

Depending on the options supplied with your Unidex 12, one or both of the following documents may have also been supplied:

- *Unidex 12 Motion Controller Options Manual* (this manual)
- *Unidex 12 Interactive Control Software Manual (SSP3)*

Please review, in detail, *Unidex 12 Motion Controller Programming Manual*, before proceeding to this or any other documentation supplied with your Unidex 12.

ORGANIZATION OF THIS MANUAL

This manual is divided into six parts, each covering an individual Unidex 12 option. They are:

Part I: IEEE-488 Parallel Interface

Part II: Joystick

Part III: Axis Calibration

Part IV: Two Digit Thumbwheel

Part V: High Speed Binary Interface (and MBI option)

Part VI: Password

TABLE OF CONTENTS

PART I: IEEE-488 PARALLEL INTERFACE

CHAPTER 1: IEEE-488 INTRODUCTION	1-1
SECTION 1-1: REQUIRED HARDWARE	1-1
1-1-1 IEEE-488 Interface	1-1
1-1-2 Signal Lines of the IEEE-488 Bus	1-3
1-1-3 Cable Restrictions of the IEEE-488 Bus	1-3
1-1-4 Parallel and Serial Polling	1-3
1-1-4-1 Parallel Polling	1-4
1-1-4-2 Serial Polling	1-4
1-1-5 Standard Interface Bus Signal Lines	1-5
CHAPTER 2: IEEE-488 OPERATION OF UNIDEX 12	1-9
SECTION 2-1: SETTING UP THE IEEE-488 PARAMETERS	1-9
CHAPTER 3: IEEE-488 BUS PROGRAMMING	1-13
SECTION 3-1: COMMUNICATING WITH UNIDEX 12	1-13
SECTION 3-2: TYPES OF COMMANDS	1-14
SECTION 3-3: SERVICE REQUEST AND SERIAL POLL	1-15
3-3-1 Conditions for Service Request	1-17
CHAPTER 4: ERROR CODES	1-19
CHAPTER 5: TYPES OF COMMANDS	1-27
SECTION 5-1: SYSTEM COMMANDS	1-27
5-1-1 Getting Unidex 12's Attention	1-27
5-1-2 Auto Mode	1-27
5-1-3 Block Mode	1-28
5-1-4 Remote Reset	1-28
5-1-5 Disabling Joystick Mode or Remote Mode	1-29
5-1-6 Downloading Program To Unidex From Host	1-29
5-1-7 Deleting a Program	1-30
5-1-8 Deleting all Programs	1-30
5-1-9 Block Numbering	1-30
5-1-10 Block Numbering Cancel	1-31
5-1-11 Hold	1-31
5-1-12 Cancel Hold	1-31
5-1-13 Immediate Mode	1-32

5-1-14 Service Request Set Up	1-33
5-1-15 Service Request Cancel	1-33
5-1-16 Status Byte in Binary Format	1-33
5-1-17 Status Byte in Hex-ASCII Format	1-33
5-1-18 Print Axis Position	1-34
5-1-19 Printing Directory Listing	1-34
5-1-20 Printing a Program	1-35
5-1-21 Printing All Programs	1-35
5-1-22 Printing Status Bytes	1-35
5-1-23 Remote Mode	1-36
5-1-24 Enabling Joystick "S"	1-36
5-1-25 Trigger	1-36
5-1-26 Enabling Joystick "U"	1-37
5-1-27 Resume Program Execution After Programmed SRQ	1-37
5-1-28 Resume and Complete Move After a Encoder Verification Error	1-37
5-1-29 Axis Calibration Table Load Command	1-38
5-1-30 Resetting Unidex 12	1-38
SECTION 5-2: MOTION PROGRAM COMMANDS	1-39
5-2-1 End of Block	1-39
5-2-2 Axis Motion Commands	1-39
5-2-3 Incremental Mode/Absolute Mode	1-40
5-2-4 Acceleration/Deceleration Ramp Time	1-41
5-2-5 Enable Velocity Profile	1-41
5-2-6 Enable Counterclockwise (Minus) Soft Limit	1-41
5-2-7 Circular Moves	1-42
5-2-8 Corner Rounding/Non-Corner Rounding	1-42
5-2-9 Disable Counterclockwise (Minus) Soft Limit	1-43
5-2-10 Disable Tracking Display	1-43
5-2-11 Dwell	1-43
5-2-12 Enable Tracking Display	1-44
5-2-13 High Run Current	1-44
5-2-14 Encoder Verification	1-44
5-2-15 Feedhold on Interrupt	1-45
5-2-16 Home	1-45

5-2-17 Interpolation Feedrate	1-46
5-2-18 Linear Move	1-47
5-2-19 Call Joystick	1-46
5-2-20 Enable Clockwise Limit	1-47
5-2-21 Disable Clockwise (Plus) Soft Limit	1-47
5-2-22 Load Position Registers	1-47
5-2-23 Counterclockwise (Minus) Soft Limit	1-48
5-2-24 Out/Run State	1-48
5-2-25 Out/Stop State	1-48
5-2-26 Print and Display Message	1-49
5-2-27 Display Message	1-49
5-2-28 Print Message	1-50
5-2-29 Clockwise (Plus) Soft Limit	1-50
5-2-30 Receive Message	1-50
5-2-31 Reset All Axis	1-51
5-2-32 Starting Free Run After a Stop or Restarting an Interrupted Move	1-51
5-2-33 Send Service Request Number	1-51
5-2-34 Stop Axes Free Run	1-52
5-2-35 Disable Velocity Profile	1-52
SECTION 5-3: I/O AND PROGRAM FLOW COMMANDS	1-53
5-3-1 Abort Move and GoSub on Interrupt	1-53
5-3-2 Beeper	1-53
5-3-3 GoTo Label on Interrupt	1-53
5-3-4 Abort Move and GoTo on Interrupt	1-54
5-3-5 Call Program	1-54
5-3-6 Conditional GoSub on Inputs	1-54
5-3-7 Conditional GoSub on DIO	1-55
5-3-8 Conditional GoSub on Marker	1-55
5-3-9 Conditional GoSub on DIO.....	1-55
5-3-10 Conditional GoTo	1-56
5-3-11 Conditional GoTo on DIO	1-56
5-3-12 Conditional GoTo on Marker	1-56
5-3-13 Conditional GoTo On DIO (Binary)	1-57
5-3-14 Binary Digital Output/BCD Digital Output	1-57

5-3-15 GoTo DIO Label	1-57
5-3-16 DIO Output (1 of 12).....	1-58
5-3-17 Disable Interrupt.....	1-58
5-3-18 GoSub	1-58
5-3-19 GoTo	1-58
5-3-20 Input State	1-59
5-3-21 Label	1-59
5-3-22 Enable Interrupt	1-59
5-3-23 Output State	1-59
5-3-24 Program Stop	1-60
5-3-25 Conditional Repeat Loop End	1-60
5-3-26 Repeat Loop End	1-60
5-3-27 Repeat Program	1-61
5-3-28 Repeat Loop Start	1-61
5-3-29 GoSub Label on Interrupt	1-61
5-3-30 Subroutine Return	1-61
SECTION 5-4: SAMPLE IEEE-488 COMMANDS	1-62
APPENDIX 1: IEEE-488 COMMAND SUMMARY FOR UNNIDEX 12	

PART II: JOYSTICK

CHAPTER 1: INTRODUCTION	2-1
SECTION 1-1: REQUIRED HARDWARE	2-1
CHAPTER 2: OPERATION	2-3
SECTION 2-1: JOYSTICK CONTROL	2-3
2-1-1 Enabling the Joystick	2-3
2-1-2 Buttons A and B	2-3
2-1-3 Button C	2-4
2-1-3-1 Unidex 12 Is In Service Request Mode	2-4
2-1-3-2 Unidex 12 in the Service Request Cancel Mode	2-6
2-1-4 Disabling the Joystick	2-7
SECTION 2-2: REMOTE (AUXILLARY) CONTROL	2-8
2-2-1 Enabling Remote Mode	2-8
2-2-2 Disabling Remote Mode	2-8
2-2-3 Enable Joystick	2-9

CHAPTER 3: DIGITIZING	2-11
SECTION 3-1: REQUIRED HARDWARE	2-11
SECTION 3-2: OPERATION	2-12
3-2-1 Initial Setup	2-12
SECTION 3-3: DIGITIZING SET UP	2-12
3-3-1 Digitize Select	2-12
3-3-2 Store/Transfer Select	2-13
3-3-3 Program # Screen	2-13
3-3-4 Incremental/Absolute Mode	2-14
3-3-5 Feedrate Setup	2-14
3-3-6 Digitizing Tracking Display Screen	2-15
3-3-7 Joystick Operation	2-16
3-3-8 Digitizing Termination	2-16
SECTION 3-3: DIGITIZING FORMATS	2-17
3-3-1 Store Format	2-17
3-3-2 RS-232C Format	2-18
CHAPTER 4: TROUBLESHOOTING	2-21
SECTION 4-1: SOFTWARE MALFUNCTIONS	2-21
SECTION 4-2 HARDWARE MALFUNCTIONS	2-21
4-2-1 Joystick Check	2-21
4-2-2 RS-232 Check	2-22
4-2-3 Editor and Memory Check	2-22
 PART III: AXIS CALIBRATION	
CHAPTER 1: AXIS CALIBRATION; INTRODUCTION	3-1
CHAPTER 2: INTERNAL COMPUTATIONS	3-3
SECTION 2-1: INTERNAL COMPUTATIONS	3-3
SECTION 2-2: SET UP	3-3
SECTION 2-3: OPERATION	3-4
SECTION 2-4: LOADING CALIBRATION TABLES INTO UNIDEX 12	3-4
SECTION 2-5: CHECKSUM	3-5
SECTION 2-6: FACTORS INFLUENCING AXIS CALIBRATION	3-5
2-6-1 Temperature	3-5

2-6-2 Fixturing	3-5
2-6-3 Line of Measurement	3-6
CHAPTER 3: EXAMPLE: AXIS CALIBRATION TABLE	3-7

PART IV: TWO DIGIT THUMBWHEEL

CHAPTER 1: THE TWO DIGIT THUMBWHEEL	4-1
SECTION 1-1: INTRODUCTION	4-1
SECTION 1-2: OPERATION	4-1

PART V: HIGH SPEED BINARY

CHAPTER 1-1: INTRODUCTION	5-1
SECTION 1-1: HARDWARE INTERFACE	5-1
CHAPTER 2: PROGRAMMING	5-3
SECTION 2-1: PROGRAMMING THE BINARY INTERFACE	5-3
SECTION 2-2: PROGRAMMABLE LOGIC CONTROLLER	5-5
CHAPTER 3: HIGH SPEED BINARY MODE	5-7
CHAPTER 4: COMMUNICATING WITH UNIDEX 12	5-11
SECTION 4-1: STATUS	5-11
SECTION 4-2: AXIS POSITION	5-11
SECTION 4-3: TRACKING DISPLAY	5-13
SECTION 4-4: ENABLING/DISABLING JOYSTICK	5-14
4-4-1 Enabling Joystick: Command "J"	5-15
4-4-2 Disable Joystick : Command "K"	5-15
SECTION 4-5: ENABLE/DISABLE REMOTE CLOCK AND DIRECTION	5-15
CHAPTER 5: ERROR AND IN-POSITION LINES	5-17
CHAPTER 6: HSB COMMAND FORMAT	5-23
6-1 Dwell	5-23
6-2 Home	5-23
6-3 Output	5-24
6-4 Input	5-24
6-5 Start Axis	5-25
6-6 Stop Axis	5-25
6-7 Load Position Registers	5-25
6-8 Incremental Mode	5-26
6-9 Absolute Mode	5-26

6-10 Beeper On	5-26
6-11 Beeper Off	5-26
6-12 Corner - Rounding Mode	5-27
6-13 Non Corner - Rounding Mode	5-27
6-14 Accel/Decel Control	5-27
CHAPTER 7: STATUS BYTES	5-29
CHAPTER 8: MBI (METRA-BYTE INTERFACE)	5-31

PART VI: PASSWORD

CHAPTER 1: PASSWORD INSTALLATION	6-1
SECTION 1-1: INTRODUCTION	6-1
SECTION 1-2: INSTALLATION	6-1
CHAPTER 2: UNIDEX 12 PASSWORD OPERATION	6-3
SECTION 2-1: CHANGING THE PASSWORD	6-3
SECTION 2-2: COMPUTER CONTROL OF UNIDEX 12	6-4
SECTION 2-3: PASSWORD CORRUPTION	6-4

SERVICE AND REPAIR
INDEX

PART I
IEEE-488 INTERFACE

CHAPTER 1: IEEE-488 INTRODUCTION

The IEEE-488 option for the Unidex 12 provides control of the Unidex 12 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 the Unidex 12. 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.

The Unidex 12 may also be set up for interactive control. In this mode, Unidex 12 sets the Service Request after executing a program, a Command Block, or if there has been an Error condition. Subsequently, the host computer is required to Serial Poll Unidex 12 before proceeding.

Before continuing, it is recommended that the user review the *Unidex 12 Motion Controller Programming Manual*.

SECTION 1-1 REQUIRED HARDWARE

1-1-1 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 Bus Controller. These properties lead to a more rapid form of communication between Unidex 12 and the controller. Bus disciplines are not necessary if your controller has IEEE-488 interface and device driver software that interfaces with Basic, Fortran, Pascal or any other language you intend to use.

Connection configurations of multiple devices to the IEEE-488 interface are illustrated 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:

- Listener
- Talker
- Controller

Device As Listener

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

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.

Device As Bus Controller

A "Bus Controller" is a device with the capability of controlling and directing activity on the bus. A Bus 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. A device will be needed to act as a Bus 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 12 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.

1-1-2 SIGNAL LINES OF THE IEEE-488 BUS

The IEEE-488 transfers data and commands between devices through 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 used 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 Section 1-1-5.

1-1-3 CABLE RESTRICTIONS OF THE IEEE-488 BUS

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

Certain limitations exist concerning the length of the cables and the number of devices allowable on the bus. The maximum number of devices on the bus is 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.

1-1-4 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 to determine the reason for the request.

1-1-4-1 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 12 may be selected through the front panel keyboard in the Set-Up mode, as described in Part I; Chapter 2 of this manual.

1-1-4-2 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 12 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 12 is Serial Polled by the controller. Upon being polled, the Unidex 12 will transmit its status.

1-1-5 IEEE-488 STANDARD INTERFACE BUS SIGNAL LINES

MNEMONIC	NAME	FUNCTION
Bus Management Lines		
IFC	Interface Clear	Only the System Controller can assert this line, placing all devices in the unaddressed state. Devices go into talker idle/listener idle state. If control has been passed to another device, the 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 does a Parallel Poll. When ATN is false, data may be sent over the bus by a designated talker.
REN	Remote Enable	Remotely asserted to program devices on the bus. 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.

MNEMONIC	NAME	FUNCTION
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).

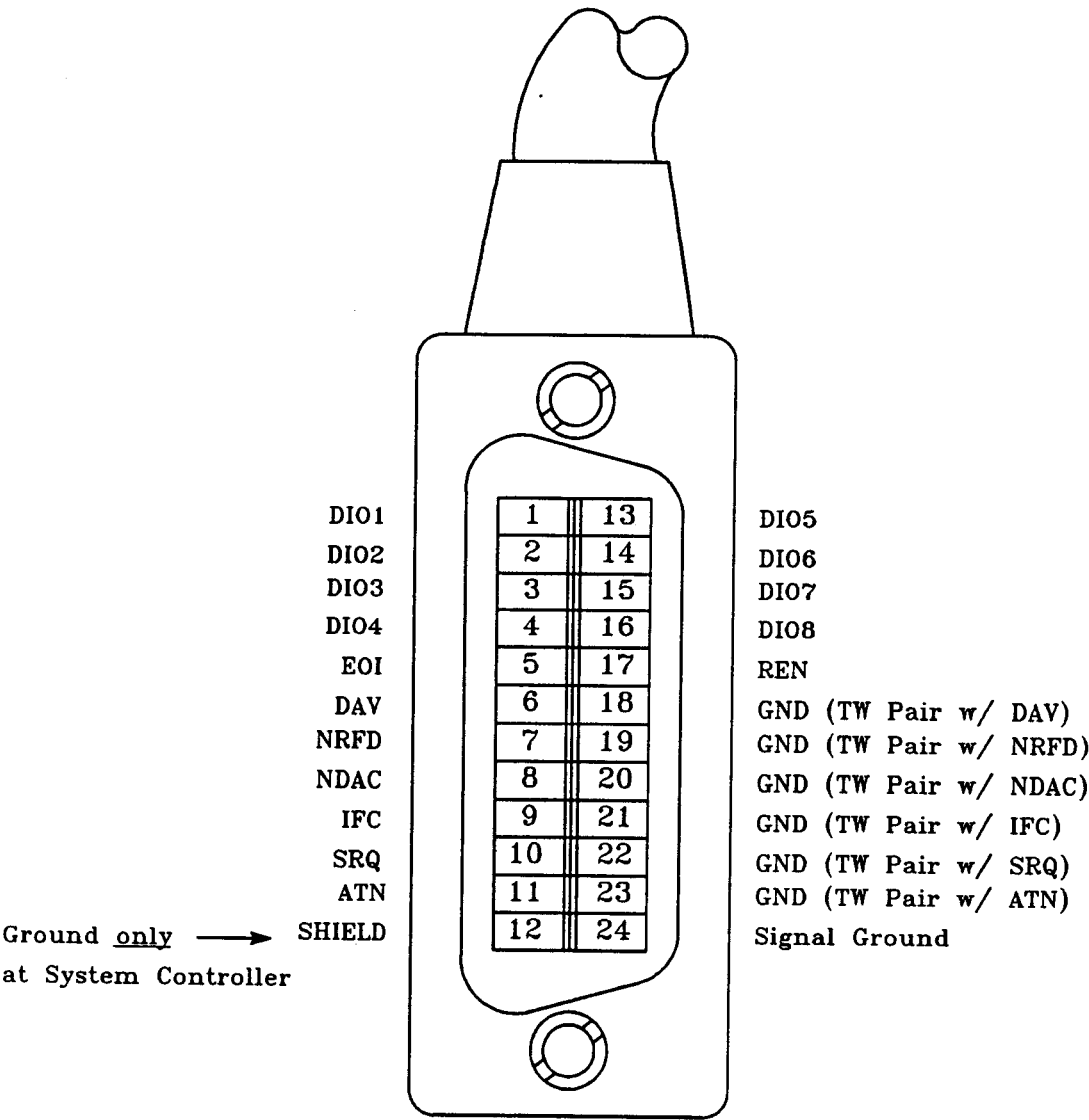
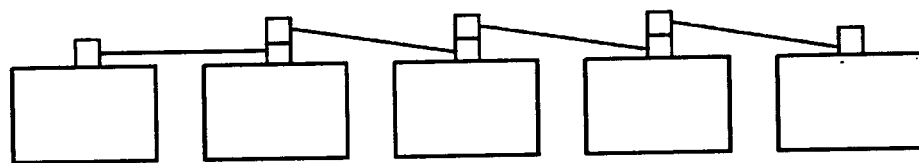
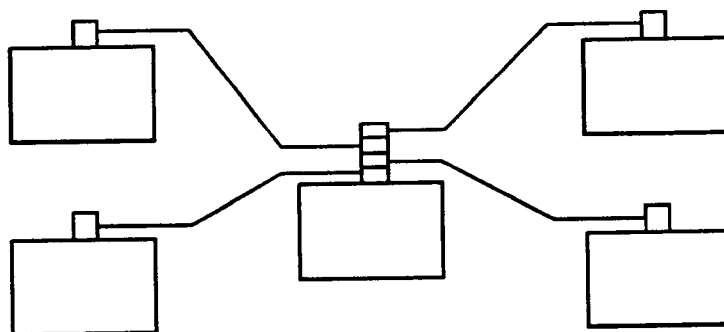


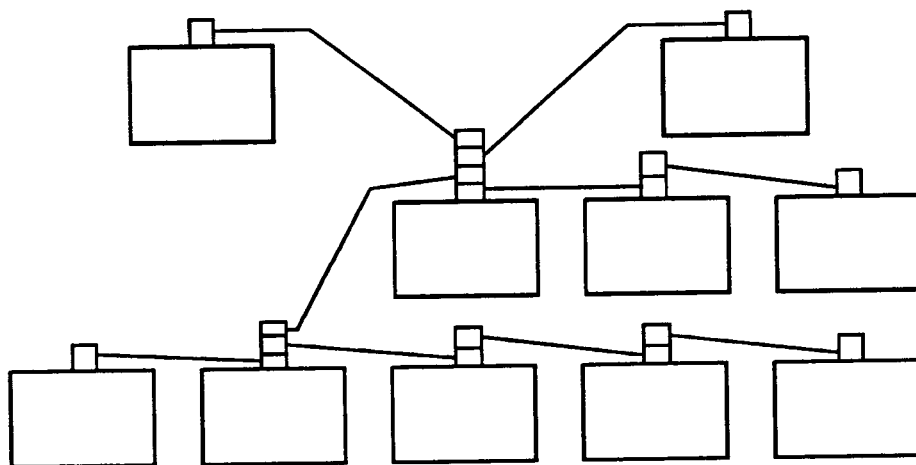
Figure 1-1: IEEE-488 Pin Connections



LINEAR



STAR



MIXED

Figure 1-2: Cabling Configurations

CHAPTER 2: IEEE-488 OPERATION OF UNIDEX 12

SECTION 2-1 SETTING UP THE IEEE-488 PARAMETERS

The device address of Unidex 12 must be set to the same number that the Bus 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 12, proceed as follows:

Press SELECT until the Main Menu's Third screen is displayed.

1. COMM. ENAB 2. PRINT 3. SET-UP
--

SELECT/BACK : NEW MENU Press 1-8 for a choice
--

Press key #3 to see:

1. Modes 2. Units 3. Comm (1) 4. Comm (2)
--

5. DIO Port 6. Drivers 7. Enc Param. 8. Other
--

BACK for Previous Menu 1-8 to make a choice
--

Press key #4 to see:

Print Block Nos. : NO Par. Poll Resp : PPR1
--

Digital Port : I/O Device Addr. (0-30) : 00
--

Press the SELECT key, the Print Block Nos. may now be enabled (Yes) or disabled (No) with the use of the +/- key.

Press the SELECT key a second time to proceed to the next parameter, Parallel Poll Response. The Parallel Poll Response number may be changed by using the +/- key. This screen sets up the Identity byte for the device. Each device on the IEEE-488 bus must be assigned one data line. The device will assert this data line when an IDENTITY 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

Press the SELECT key to proceed to the next parameter, Digital Port. Use the +/- key to select the desired configuration.

Press the SELECT key to proceed to the final parameter, Device Address (0-30). This requires the communication identification number for a Unidex 12 when more than one Unidex 12 is used.

Press BACK to return to the Main Menu's Third screen.

1. COMM ENAB 2. PRINT
3. SET-UP

SELECT/BACK : New Menu
Press 1-8 for choice

Insert Block Nos. when
Printing programs : NO

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 3: IEEE-488 BUS PROGRAMMING

SECTION 3-1 COMMUNICATING WITH UNIDEX 12

Unidex 12 is ready for communication at power up if the following conditions are met:

- Device Address has been correctly set.
- Default values are in the desired format.
- Unidex 12 is equipped with a battery back-up.

Default values are:

DEVICE ADDRESS : 00 PARALLEL POLL RESPONSE : PPR1
--

If the keyboard is operated, the communication interface becomes disabled and Unidex 12 will not respond to commands on the IEEE-488 bus until the interface is enabled from the keyboard.

To enable IEEE-488 Interface, from the Third screen of the Main Menu, press key #1 to see:

1. RS-232/IEEE-488 2. TDT/HSB

Press key #1, RS-232/IEEE-488 to see:

IEEE-488 COMMUNICATION ENABLED

The Host device must now address Unix 12 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, Unix 12 will display:

<p>IEEE-488 INTERFACE COMMUNICATION ACTIVE</p>
--

SECTION 3-2 TYPES OF COMMANDS

A command sent to Unix 12 through the IEEE-488 interface may be classified into two types:

System Commands:

These commands interact with Unix 12 as a device and perform operations such as resetting Unix 12, printing a program, printing position values, running a program, downloading a program, etc. Each system command establishes a mode of operation once it is received by Unix 12.

Program (Motion) Commands:

These are the user program blocks in a motion program that Unix 12 executes when running the program in the Auto or in the Block mode. Program commands are valid only if entered 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 12) 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 it requires the attention of the master. The reason for requesting service may be an error condition or to signal the completion of a task.

Unidex 12 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 12 has sent the Service Request signal is to Serial Poll Unidex 12. Unidex 12 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 12.

The 8 bits of this status byte represent different internal states of Unidex 12. Responding to a Serial Poll is a hardware function and therefore the controller may Serial Poll Unidex 12 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	Absolute mode	Incremental 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 12 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 12 will not send a Service Request signal for any reason. This is the Default mode. In the Default 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 12 is ready to take the next command.

3-3-1 CONDITIONS FOR SERVICE REQUEST

When in the Service Request mode, Unidex 12 sets a Service Request (SRQ) under the following conditions:

- When an Immediate command execution is complete.
- When a program is completely executed in the Auto Run mode.
- When a block is executed in the Block Run mode.
- When a run time error condition is generated and the program is aborted.
- When an axis limit is activated.
- When a program or immediate command move is stopped by pressing the STOP key on the front panel keypad.
- At the end of a program download operation, if an error was generated while downloading. (The SRQ is set by Unidex 12 after the "%" that ends the downloading of the program.)
- When it is requested that a nonexistent program be printed. (If "Pnn < CR > < LF > " is sent to Unidex 12 and program #nn does not exist, Unidex 12 will set the SRQ line.)

NOTE: For more information on Error Bytes see Part I: Chapter 4 of this manual.

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

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

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 4: ERROR CODES

The Host computer may detect an error condition by checking the most significant bit (bit 7) of the Serial Poll status byte (Part I: Chapter 3; Section 3-3 of this 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 12 will send back 13 bytes followed by <CR> <LF>. (Refer to Chapter 5 of Part I of this manual for details on printing status bytes.) These 13 bytes represent a complete status report of Unidex 12, and are described as follows:

BYTE 1 Same as Serial Poll status byte in Section 3-3 of Part I: Chapter 3 of this 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

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 number called for run
Bit 6 *	Memory not clear	No programs in memory (Memory clear)
Bit 7 *	No user memory checksum error	User memory checksum error

*** NOTE:** If one of these errors is generated during download operation, Unidex 12 will send an SRQ (Service Request) character if you are in the Service Request mode. It is recommended that the user then edit and correct that program.

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" error	"Incomplete subroutine" error
Bit 6 *	Eight subroutines not exceeded	Eight subroutines exceeded
Bit 7 *	No "Missing Label" error	"Missing Label" error

*** NOTE:** If one of these errors is generated during download operation, Unidex 12 will send an SRQ (Service Request) character if you are in the Service Request mode. It is recommended that the user then edit and correct that program.

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 present system	LCD Display 1 not included in system
Bit 1	Not Used	
Bit 2	In Hold mode, but no trigger command received	In Hold mode, and trigger command received
Bit 3	Unidex 12 "receive buffer" not full	Unidex 12 "receive buffer" full
Bit 4	X on received during transmit	X off 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	Not in TDT mode	In TDT mode

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	X axis no Encoder Verification Error	X axis Encoder Verification Error
Bit 5	Y axis no Encoder Verification Error	Y axis Encoder Verification Error
Bit 6	U axis no Encoder Verification Error	U axis Encoder Verification Error
Bit 7	V axis no Encoder Verification Error	V axis Encoder Verification Error

BYTE 8**RAMPER BOARD STATUS**

	Zero	One
Bit 0	X axis does not have in-position signal	X axis has in-position signal
Bit 1	Y axis does not have in-position signal	Y axis has in-position signal
Bit 2	U axis does not have in-position signal	U axis has in-position signal
Bit 3	V axis does not have in-position signal	V axis has in-position signal
Bit 4	X axis no Soft Limit Error	X axis Soft Limit Error
Bit 5	Y axis no Soft Limit Error	Y axis Soft Limit Error
Bit 6	U axis no Soft Limit Error	U axis Soft Limit Error
Bit 7	V axis no Soft Limit Error	V axis Soft Limit Error

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		
	Not used	
Bit 7		

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		
	Not used	
Bit 7		

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
|| - 12 bit value of digital output

Bit 11

Bit 12
|| - Not used

Bit 15

CHAPTER 5: TYPES OF COMMANDS

SECTION 5-1 SYSTEM COMMANDS

System commands interact with Unidex 12 as a device and perform operations such as resetting Unidex 12, printing a program, printing position values, running a program, downloading a program, transferring status byte information from Unidex 12, etc. Each system command establishes a mode of operation once it is received by Unidex 12. Each system command must be entered as a capital letter. Transmission to Unidex 12 may be terminated by a <LF>, <CR> <LF>, EOI or <CR> <LF> EOI.

5-1-1 GETTING UNIDEX 12'S ATTENTION

To activate the IEEE-488 interface in the Unidex 12, 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.

5-1-2 AUTO MODE

Executing a program in the Auto mode enables the program to run automatically, and execute 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 the Unidex 12 has been put into the Service Request mode prior to executing a program in the Auto Mode (see Section 3-3 of Part I: Chapter 3 of this manual), at the completion of program execution the Unidex 12 will send the Service Request character (SRQ) and wait for a Serial Poll. After the Serial Poll, you may execute

5-1-3 BLOCK MODE

The Block mode is used to run a program, 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 the Unidex 12 has been put into the Service Request mode, prior to executing a program block, the Unidex 12 will send the SRQ character after each block has been executed. The Host must then Serial Poll (see Section 3-3 of Part I: Chapter 3 of this manual) the Unidex 12, following the execution of each block.

After the execution of the first Command Block and Serial Poll, send a <CR> <LF> to execute the next block. Bit 1 of the Status Byte (see Section 3-3 of Part I: Chapter 3 of this manual) may be checked to detect completion of the program. This bit is cleared following the last block in the program.

5-1-4 REMOTE RESET

Sending the command "C" followed by <CR> <LF> resets Unidex 12 after a system command is executed. This returns it to power up conditions. Example:

C <CR> <LF>

The IEEE-488 bus functions "Device Clear" (DCL) and "Selected Device Clear"(SDC) effects the same Reset response from the Unidex 12.

5-1-5 DISABLING JOYSTICK MODE OR REMOTE MODE

This command is available with the Unidex 12 Joystick (JP4E) option only. For more information on this option, see Part II, of this manual.

The system 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 in the Computer Enabled Joystick Mode, Unidex 12 will respond to the Serial Poll command, the "D" command and the Print Position and Status commands. When in the Remote Mode, response will only be to the "D" commands.

5-1-6 DOWNLOADING PROGRAM TO UNIDEX FROM HOST

The "E" command, followed by a program number ("nn") and an end-of-block character (* or /), will put Unidex 12 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 12, 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 12 user memory and stored in program #10. The motion commands that may be included in this program will be discussed in Part I: Chapter 5; Section 5-2, of this manual.

5-1-7 DELETING A PROGRAM

To delete a program from the Unidex 12 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.

5-1-8 DELETING ALL PROGRAMS (FROM USER MEMORY)

To delete all programs from the Unidex 12 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.

5-1-9 BLOCK NUMBERING

If programs are 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 12, all programs will be printed with block numbers.

5-1-10 BLOCK NUMBERING CANCEL

To cancel block numbering during program printing, send the command "G" followed by <CR> <LF>. Example:

G <CR> <LF>

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

5-1-11 HOLD

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

H <CR> <LF>

The above command causes Unidex 12 to suspend execution of any Immediate, Auto or Block commands which may follow it. This is useful if synchronization of axis motion with some other action is necessary. Unidex 12 will only execute the commands when it receives a "T" (for Trigger) command (see Item 5-1-27). Example:

H <CR> <LF>

A 20 <CR> <LF> Program #20 not executed (held)

T <CR> <LF> Program #20 executes

OR

(GET) Bus Interface Message

5-1-12 CANCEL HOLD

"O" cancels the Hold ("H") command and allows Unidex 12 to execute the Auto, Immediate or Block command:

O <CR> <LF>

5-1-13 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 commands 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 Service Request mode, Unidex 12 will send a SRQ character and wait for a Serial Poll after the command is executed. After being polled, Unidex 12 is ready to execute another block of commands.

All motion commands that make up a Unidex 12 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 a full explanation of each, see Section 5-2 of this chapter):

AB*	HX*	OT*	CCA1vddddd
AD*	IN*	RX*	A2vddddd
BF*	IT*	SX*	I vddddd
BN*	LX*	X F D*	J vddddd
CO*	NC*	Y F D*	CWA1vddddd
DB*	OR*	U F D*	A2vddddd
DD*	OS*	V F D*	I vddddd
DW*	EA	FA	J vddddd
IFfff	JS	PD ^	IXvddddd
PM ^	PR ^	RD ^	Yvddddd
RR	DLdd	DHdd	Uvddddd
			Vvddddd

NOTE: All of the above commands must be preceeded by the character "I".

5-1-14 SERVICE REQUEST SET UP

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 12 will send the SRQ signal under conditions described in Section 3-3 of Part I: Chapter 3 of this manual. It then waits until it receives a Serial Poll from the Host device before executing any further commands.

5-1-15 SERVICE REQUEST CANCEL

To cancel the Service Request (SRQ) mode established by the J command, send a K command, followed by <CR> <LF>. Example:

K <CR> <LF>

(SRQ canceled is the default status.)

5-1-16 STATUS BYTE IN BINARY FORMAT

To establish the format of the Status Bytes as binary upon transmission, send command "M", followed by <CR> <LF>. Example:

M <CR> <LF>

Status Bytes will be sent as 13 binary bytes followed by <CR> <LF>. Binary format is the default status.

5-1-17 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>

Status bytes will be sent as 13 sets of 3 bytes (2Hex - ASCII + space) ended by <CR> <LF>.

5-1-18 PRINT AXIS POSITION

1. Print X Axis Position (PX)

To print the X axis position register, send:

PX <CR> <LF> (Address Unidex 12 to talk)

The axis position is sent in the following format:

<space> or <negative sign> <10 digits> <CR> <LF>
(Address Unidex 12 to talk)

2. Print Y Axis Position (PY)

To print the Y axis position register, send:

PY <CR> <LF> (Address Unidex 12 to talk)

3. Print U Axis Position (PU)

To print the U axis position register, send:

PU <CR> <LF> (Address Unidex 12 to talk)

4. Print V Axis Position (PV)

To print the V axis position register, send:

PV <CR> <LF> (Address Unidex 12 to talk)

5-1-19 PRINTING DIRECTORY LISTING

To print a listing of the programs in the Unidex 12 directory, send:

PD <CR> <LF> (Address Unidex 12 to talk)

Bytes of memory remaining in Unidex 12 will be printed as well. Transmission of data will be terminated by character <ETX>.

5-1-20 PRINTING A PROGRAM

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

P10 <CR> <LF> (Address Unidex 12 to talk)

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

5-1-21 PRINTING ALL PROGRAMS

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

P00 <CR> <LF> (Address Unidex 12 to talk)

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

5-1-22 PRINTING STATUS BYTES

To print the Status Bytes listed in Part 1: Chapter 4 of this manual, send:

PS <CR> <LF> (Address Unidex 12 to talk)

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

5-1-23 REMOTE MODE

The system command

R <CR> <LF>

enables Unidex 12 with the "JP4E" 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 12. Unidex 12 keeps track of the axes' positions during external control.

5-1-24 ENABLING JOYSTICK "S"

This command is available with the Unidex 12 Joystick (JP4E) option only. For more information refer to Part II of this manual.

The system command:

S <CR> <LF>

puts Unidex 12 into the "computer-enabled" Joystick mode. The display changes to the Joystick mode Tracking display (Part II: Chapter 2 of this manual).

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.

5-1-25 TRIGGER

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

T <CR> <LF>

or

GET (Group Execute Trigger)

5-1-26 ENABLING JOYSTICK "U"

This command :

U <CR> <LF>

enables the Joystick in much the same manner as Item 5-1-26 , however, upon an axis encountering a Limit, the Unidex 12 sends the Service Request character to the Host computer.

For more information refer to Part II of this manual.

5-1-27 RESUME PROGRAM EXECUTION AFTER PROGRAMMED SRQ

To resume program execution after a programmed Service Request is generated by the Unidex 12, send:

V <CR> <LF>

(See Item 5-2-33 for further information concerning a programmed Service Request.)

5-1-28 RESUME AND COMPLETE MOVE AFTER A ENCODER VERIFICATION ERROR

When one or more Axes in motion has an Encoder Verification Error at the end of an indexed move, the Unidex 12 will display an "OUT OF POSITION " error. The Program is halted and control is returned to the Host computer. If the Unidex 12 is in the SRQ mode, a Service Request is set. The Host must now do a Serial Poll("Q") and the status received will show an Error condition. Byte 7 of the detailed status (See PartIV: Chapter 4) provides information regarding the Axis containing the Encoder Verification Error. The Print Position commands may be used at this time if needed.

To resume Program execution the Host must send:

W <CR> <LF>

To terminate the Program Command the Host must send :

X <CR> <LF>

5-1-29 AXIS CALIBRATION TABLE LOAD COMMAND

The command "Z" followed by an Axis designation load an Axis Calibration Table into Unidex 12.

Refer to Part III of this manual for more information

5-1-30 RESETTNG UNIDEX 12

To reset Unidex 12, send the hexadecimal number 7F or FF. Either is the ASCII code for .

SECTION 5-2 MOTION PROGRAM COMMANDS

The motion program commands make up the program that Unidex 12 executes while 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 is given in Section 5-1-13 (Immediate Mode) of this Chapter.

5-2-1 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 /)
```

5-2-2 AXIS MOTION COMMANDS

The axis to move must 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) is specified with a distance command (D).

The following is an example of a program block utilizing the above commands:

```
X F10000 D150000 *
```

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

NOTE: When programming through RS-232 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.

5-2-3 INCREMENTAL MODE/ABSOLUTE MODE

The Incremental mode uses a distance command to tell Unidex 12 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 *

The Absolute mode uses distance command as an absolute position. For example:

X F10000 D1000 *

When in the Absolute mode, the above example instructs the Unidex 12 to send the X axis to the position 1000. Once there, the re-execution 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 *

5-2-4 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.

5-2-5 ENABLE VELOCITY PROFILE

Velocity Profile may be enabled by sending an "A" followed by the axis designation to be profiled. Example:

AX *
or
AV *

The first command enables the X axis for Velocity Profiling, the second example will enable the V axis.

5-2-6 ENABLE COUNTERCLOCKWISE (CCW) (MINUS) SOFT LIMIT

CCW Limit may be enabled by sending a "B" followed by the axis designation to be enabled. Example:

BY *
or
BXYUV *

The first example enables the Y axis CCW Limit. The second example enables the CCW Limit for all axes.

5-2-7 CIRCULAR MOVES

A Circular move may be sent by entering the character "C" and then following it with a "W" for a clockwise circle, or a "C" for a counterclockwise circle. This is then followed with the first axis character (X,Y,U, or V) and it's incremental end point distance. Following this should be the "I" and "J" characters with their incremental distances. Example:

CW X 20000 Y 15000 I 25000 J-2000 *

or

CC V 33000 U-25000 I 30000 J 3000 *

The first example is a clockwise circle with the first axis X and the second axis Y. Following the axes are the "I" and "J" values. The second example is a counterclockwise circle using V as the first axis and U as the second.

5-2-8 CORNER ROUNDING/NON-CORNER ROUNDING

The command to implement corner rounding is "CO" followed by an end-of-block. Example:

CO *

The above example would enable corner rounding for all four axes. In the corner rounding mode, a new mode is started without waiting for the Servo Lag from the previous move to be completed (In-Position signal). This feature serves to smooth the motion between blocks.

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.

5-2-9 DISABLE COUNTERCLOCKWISE (CCW) (MINUS) SOFT LIMIT

The CCW Limit may be disabled by the command "C" followed by the axis or axes to be disabled. Example:

CX
or
CXYUV

The first example will disable the X axis CCW Limit only. The second example disables the CCW Limit for all axes.

5-2-10 DISABLE TRACKING DISPLAY

The Tracking display may be disabled by the command "DT" followed by an end-of-block. Example:

DT*

Disabling the Tracking Display eliminates the time necessary to compute position every 200 ms. Overhead time between blocks becomes more consistent.

5-2-11 DWELL

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

DW 10 *
or
DW 1.0 *

The examples above would enter a Dwell (programmed wait) of 10 seconds and 1 second respectively within the program.

5-2-12 ENABLE TRACKING DISPLAY

Tracking Display is enabled by sending the command "ET". Example:

ET*

This command causes the display to be updated and enables the display if it is disabled.

5-2-13 HIGH RUN CURRENT

This command applies only to Stepper Motors that are run in the Open Loop mode. It sets the Run current for all axes. Enter a E, followed by the axes which are to run in high current. Example:

EY*
or
EXYUV *

The first example will cause the Y axis to run in high current and the remaining axes to run in low current. The second example causes all of the axes to run in high current.

5-2-14 ENCODER VERIFICATION

This command selects which axes are to be Encoder verified. Enter a F character followed by the axis or axes to be verified. Example:

FX *
or
FXYUV *

The first example will enable only the X axis for Encoder verification. The second example will enable all axes for Encoder verification.

Those axes having Encoder verification, compare the commanded position to the Encoder position at the end of each move block. An Error message will be displayed if these positions differ by more than the programmed margin (Refer to the Unidex 12 Programming Manual, Part II for more information) and the program will be halted.

When in the "Local" mode of operation the RUN key may be depressed to continue motion and complete the move and resume program execution. Pressing the BACK key will terminate the program.

When using a Host computer, the Unidex 12 will halt the program and return to system level. If in the SRQ mode, a Service Request is set. Following a Serial Poll, the Host computer may be used to send :

W <CR> <LF> : Resumes Program
X <CR> <LF> : Terminates Program

5-2-15 FEEDHOLD ON INTERRUPT

The command Feedhold on Interrupt is entered by "FI" followed by the Input line number (1-4). Example:

FI2 *

This example will provide a Feedhold if an Interrupt occurs on Line # 2.

5-2-16 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 the X axis home. The second command will send all axes home.

5-2-17 INTERPOLATION FEEDRATE

The Interpolation Feedrate characters, "IF", followed by the Feedrate will load the Interpolation Feedrate. Example:

IF 10000 *

This Feedrate value is used in the Linear and Circular Interpolation calculations. (Refer to the Unidex 12 Programming Manual, Part II for further information)

5-2-18 LINEAR MOVE

A Linear Move may be sent by first sending the character "I" followed by the axis code and move value for that axis. Repeat the axis code and move value for each axis to be moved. Example:

I X 10000 U 25000 *

or

I X 5000 Y 8000 U 11000 V 975 *

The first example moves the X and U axes together in a Linear Move. The second example moves all axes in a Linear Move.

Refer to the Unidex 12 Programming Manual, Part II for further information.

5-2-19 CALL JOYSTICK

The command "JS" may be used to initiate Joystick control. Example:

JS *

The Joystick will return control to the program when Button C is pressed.

5-2-20 ENABLE CLOCKWISE (CW) LIMIT

The CW Limit is enabled by entering a "J" followed by the axis or axes to be enabled. Example:

JU *
or
JXYUV *

The first example enables the U axis CW Limit, the second example enables the CW Limit for all of the axes.

5-2-21 DISABLE CLOCKWISE (CW) (PLUS) SOFT LIMIT

To disable the CW Limit, send a "K" followed by the axis or axes to be disabled. Example:

KV *
or
KXYUV *

The first example will disable the V axis CW Limit. The second example will disable the CW Limit for all of the axes.

5-2-22 LOAD POSITION REGISTERS

Any of the axes Position Registers may be loaded 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.

5-2-23 COUNTERCLOCKWISE (CCW)(MINUS) SOFT LIMIT

One or more of the axes may be loaded with a positive or negative CCW Limit position by sending the CCW Limit code "M" followed by the axes limits. Example:

MX 20000 Y -30000 V 15000 *

In the above example, the X, Y, and V CCW Limit positions are loaded. A limit will now occur if a CCW commanded move takes the axis beyond one of the loaded limit positions.

5-2-24 OUT/RUN STATE

Permits output values to be sent to the Outputs when the program is instructed 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 (all have been programmed as "don't cares").

5-2-25 OUT/STOP STATE

Permits output values to be sent 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.

5-2-26 PRINT AND DISPLAY MESSAGE

The Print and Display message command "PD" may be followed by a maximum of 48 characters of message. The character "^" must be entered following the last message character. Example:

PD UNIDEX 12 ^ * (Address Unidex 12 to talk)

The previous example will print and display via the RS 232 (when in the Local Mode) the message "UNIDEX 12". Characters or codes which are used for control purposes should not be used within Print or Display messages or problems with the Unidex 12 could result. Some such characters are:

>, <, ## and 7F (Hex) (Address Unidex 12 to talk)

5-2-27 DISPLAY MESSAGE

The Display Message command "PM" may be followed by a maximum of 48 characters of message. The character "^" must be entered following the last message character. Example:

PM UNIDEX 12 ^ * (Address Unidex 12 to talk)

The above example will display the message "UNIDEX 12". Characters or codes which are used for control purposes should not be used within Display messages or problems with the Unidex 12 could result. Some such characters are:

>, <, ## and 7F (Hex) (Address Unidex 12 to talk)

5-2-28 PRINT MESSAGE

The Print Message command "PR" may be followed by a maximum of 48 characters of message. The character "^" must be entered following the last message character. Example:

PR UNIDEX 12 ^ * (Address Unidex 12 to talk)

The above example will print the message "UNIDEX 12" when in the Local mode. Characters or codes which are used for control purposes should not be used within Display messages or problems with the Unidex 12 could result. Some such characters are:

>, <, ## and 7F(Hex) (Address Unidex 12 to talk)

5-2-29 CLOCKWISE (CW) (PLUS) SOFT LIMIT

One or more of the axes may be loaded with a CW Limit position. This Limit position may be positive or negative. The command is entered by typing the character "P" followed by the designated axis then the limit. Example:

P X 1723 Y 2650 U -3400 *

In the above example, the X, Y, and U CW Limit positions are loaded. A limit will now occur if a CW commanded move reaches one of the loaded limit positions.

Refer to the Unidex 12 Programming Manual, Part II: Appendix 2.

5-2-30 RECEIVE MESSAGE

To receive a message enter "RD" followed by an end-of-block. Example:

RD *

This command will display a message on the screen as it is received. Message reception is terminated with <CR> <LF> or when 48 characters are received.

5-2-31 RESET ALL AXIS

All axes may be reset by entering "RR" followed by an end-of- block. Example:

RR *

This example will cause a reset to be sent to all axes drives.

5-2-32 STARTING FREE RUN AFTER A STOP FREE RUN OR RE-STARTING AN INTERRUPTED MOVE

After a Free Run has been stopped (see Item 5-2-34), 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 the X axis. The second example restarts all axes.

5-2-33 SEND SERVICE REQUEST NUMBER

Unidex 12 may be programmed to serve a Service Request command by entering "SS" followed by it's identification number (0-9). Example:

SS5 *

The above example sends a Service Request number (5). When Unidex 12 executes this command during a user program, it sends a SRQ signal to the Host Controller. The Host Controller must now perform a Serial Poll. The status sent back by Unidex 12 under this condition is unique.

Bit 7	-0	
Bit 6	-1	(SRQ Sent)
Bit 5	-1	(Immediate Mode Active)
Bit 4	-0	
Bit 3		(The ID Number of the SRQ)
Bit 2		" " " "
Bit 1		" " " "
Bit 0		" " " "

The Host may now resume program execution by sending the command <CR> <LF>. The ID Number will identify the Service Request if there are multiple program SRQs.

5-2-34 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.

5-2-35 DISABLE VELOCITY PROFILE

Velocity Profiling may be disabled by sending "ZS". Example:

ZS *

The above example disables velocity profiling.

SECTION 5-3 : I/O AND PROGRAM FLOW COMMANDS

5-3-1 ABORT MOVE AND GOSUB ON INTERRUPT

To initiate the Abort Move and GoSub on Interrupt command, send the characters "AI" followed by the GoSub Label number. Then send "I" followed by the Interrupt line number (1-4). Example:

```
AI10I2 *
```

The above example will Abort the move and GoSub Label # 10 for Interrupt on Input line #2.

5-3-2 BEEPER (ON/OFF)

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 1/2 of a second and then turns it off.

5-3-3 GOTO LABEL ON INTERRUPT

This command directs the program to a specified Label if a Interrupt should occur. Send the GoTo command "BI" followed by the Label number. Then send the Input Line code "I" followed by the Line number (1-4). Example:

```
BI 15 I3 *
```

The above example will cause the program to GoTo Label number 15 if an interrupt on Input Line 3 occurs.

5-3-4 ABORT MOVE AND GOTO ON INTERRUPT

The Abort a Move and GoTo command is initiated by sending the characters "CI" followed by the GoTo Label number. Then send an "I" followed by the Interrupt line number (1-4). Example:

CI 27 I3 *

5-3-5 CALL PROGRAM

For a Call Program command, send the characters "CP" followed by the desired program number. Example:

CP35 *

The above example calls program #35 and will execute it as if it was a subroutine and then return.

NOTE: A Program used with the Call Program command as a subroutine, does not require a Subroutine Return "SR" command at its end, control will return to the calling program.

5-3-6 CONDITIONAL GOSUB ON INPUTS

The command to send the program to a subroutine if the input status matches specified values is "CS". It must be followed by the label number, an "I" (for input) and the desired input values. For example:

CS 33 I 110X

The above command tells Unidex 12 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 may be nested in combination with regular subroutines to 8 levels deep.

5-3-7 CONDITIONAL GOSUB ON DIO (1 of 12)

A program may be sent to a subroutine if the DIO 1 of 12 line matches a specified value. Enter the characters "CS", followed by the Label number, the Logic level (L or H) and the DIO Line number. Example:

CS 24 L7 *
or
CS 59 H11 *

The first example is a conditional GoSub to Label #24 if the DIO line #7 is Low. The second example is a conditional GoSub to Label #59 if the DIO line #11 is High.

5-3-8 CONDITIONAL GOSUB ON MARKER

The command which will send the program to a Subroutine if the Markers for a specified axis is present, is entered by typing "CS", followed by the Label number, an "M" (for Marker) and the desired axes. Example:

CS 24 M Y *
or
CS 37 M XYUV *

The first example will cause the subroutine number 24 to be called if axis Y is on the Marker. The second example will cause the subroutine number 37 to be called if all of the axes are on their respective markers.

5-3-9 CONDITIONAL GOSUB ON DIO (BINARY)

For a Conditional GoSub on DIO (Binary) send the character "CS", followed by the Label number, a "B" (for Binary) and the DIO Input conditions (0,1,X). Example:

CS 41 B 10XXXXXXXXXXXXX *

The above example will call subroutine Label number 41 if the binary Input conditions MSB equals 1 and the next bit is equal to 0. The remaining bits are don't care (X).

5-3-10 CONDITIONAL GOTO

The command "CT", followed by a label number, an "I" and four input values, gives instruction 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 12 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 with the next program block.

5-3-11 CONDITIONAL GOTO ON DIO (1 of 12)

The Conditional GoTo on DIO permits branching to a Label if the DIO (1 of 12) line matches the specified value. Enter "CT" followed by the Label number, the Logic level (L or H) and the DIO Line number. Example:

CT 366 L4 *

or

CT 75 H 9 *

The first example is Conditional GoTo to Label #36 if the DIO Line #4 is Low. The second example is a Conditional GoTo Label #75 if the DIO Line #9 High.

5-3-12 CONDITIONAL GOTO ON MARKER

Conditional GoTo on Marker is enabled by typing "CT". It must be followed by the Label number, "M" (for Marker) and the desired axis. Example:

CT 17 MV *

or

CT 26 MXYUV

The first example will branch to Label #17 if axis V is on Marker. The second example will branch to Label #26 if all of the axes are on Marker.

5-3-13 CONDITIONAL GOTO ON DIO (BINARY)

Conditional GoTo on DIO (Binary) is enabled by typing "CT", followed by the Label number, a "B" (for Binary) and the DIO Input conditions (0,1,X). Example:

CT 34 BXXXXXXXXXX10 *

The above example will GoTo Label #34 if the LSB of the Binary Input conditions is equal to 0 and the next bit is equal to 1. The remaining bits are don't care (X).

5-3-14 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 the output. Example:

DD 22 *

In the above example the binary equivalent of the number 22 will be output on the 12 output lines as: 000000010110

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 as 0000 0010 0010. A mSec strobe will be output on the strobe line after the 12 bit value settles.

5-3-15 GOTO DIO LABEL

The command to execute GoTo DIO Label is "DG" followed by an end-of-block. Example:

DG *

This command will read the DIO port and retrieve the Label number, to which it will branch. The Label Number is formed by taking the DIO:8 - through DIO:1 (8 Bits) as a two digit Binary Coded number.

5-3-16 DIO OUTPUT (1 of 12)

To program the DIO 1 of 12 Output, enter a "D" followed by the Logic Level (L OR H) and the Output Line number # (1-12). Example:

DL 10 *
or
DH 7 *

The first example sets the DIO Output line # 10, Low. The second example sets the DIO Output line # 7 High.

5-3-17 DISABLE INTERRUPT

Disable Interrupt is initiated by entering "DI" followed by a Y to disable or a N to retain as enabled, for each Input (I1-I4). Example:

DI YNNN *

This example disables Input #1 as a interrupt. The remaining Interrupts will remain enabled if previously enabled.

5-3-18 GOSUB

This command tells Unidex 12 to execute a subroutine at label #nn. It is initiated by typing "GS", followed by the block label number and an end-of-block. Example:

GS 15 *

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

5-3-19 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 12 to continue program execution at label 20.

5-3-20 INPUT STATE

To set up the conditions that the inputs should attain prior to the program continuing, program an "IT" command, followed by the desired conditions 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".

5-3-21 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 *

5-3-22 ENABLE INTERRUPT

Enable Interrupt is initiated by typing "LI" followed by the interrupt conditions (0,1,X) for I1 to I4. Example:

LI 01XX *

The above example Interrupt I1 is a 0, I2 is a 1 and I3 and I4 are don't care.

5-3-23 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.

5-3-24 PROGRAM STOP

Program Stop marks the place in the program at which program execution ends. To initiate a Program Stop, type in "PS". Example:

PS *

Subroutines should be placed after this block.

5-3-25 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.

The Repeat Loop ends when the programmed number of iterations are completed, if the input conditions have not been satisfied until that time.

5-3-26 REPEAT LOOP END

To mark the end of the Repeat Loop (discussed in the above subsection), enter the program command "RE" followed by an end-of-block. Example:

RE *

Repeat loops may be nested to eight levels deep.

5-3-27 REPEAT PROGRAM

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

RP *

NOTE: Remember, any commands following this command within the program will not be executed.

5-3-28 REPEAT LOOP START

The command to start a repeat loop in your program and the number of times the loop executes is established with a "RS" command, 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. (The section of the program that is repeated 8 times, is delineated by the first RE* command that follows this command)

5-3-29 GOSUB LABEL ON INTERRUPT

The GoSub Label on Interrupt will call a specified subroutine if an Interrupt should occur. Enter the characters "SI", followed by the Label number, the Input Line code "I" then the Input line number (1-4). Example:

SI 27 I Y *

The above example will call the Subroutine with label #27 if the Interrupt on Input line 4 occurs.

5-3-30 SUBROUTINE RETURN

This command causes Unidex 12 to return from the Subroutine to the Program Block that immediately follows the "GS nn" block. Every subroutine should end with a Subroutine Return.

SR *

Subroutines may be nested to 8 levels deep.

SECTION 5-4 SAMPLE IEEE-488 COMMANDS

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

SAMPLE IMMEDIATE COMMANDS

Address Unidex 12 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 * <CR> <LF>	; X and Y axes move

SAMPLE MOTION PROGRAM

Address Unidex 12 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 12 to the block number printing mode.

P01 <CR> <LF> (Address Unidex 12 to Talk)

The above command will cause program #1 to be printed with block numbers when Unidex 12 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> (Address Unidex 12 to Talk)

The Y position register may be read by sending:

PY <CR> <LF> (Address Unidex 12 to Talk)

Send J <CR> <LF> to put Unidex 12 in the Service Request Mode. (You may enter a character of your own choosing or use the default Service Request character, %.) When this character is sent by Unidex 12, acknowledge it with a Q <CR> <LF> (query) before continuing. If running the program in the Auto mode, the Service Request will come after the program execution. If running a program block by block, it will follow each block.

Send command K <CR> <LF> to cancel the Service Request mode.

APPENDIX 1: IEEE-488 COMMAND SUMMARY FOR UNIDEX 12

The following is a list of commands for the Unidex 12 when operating via the IEEE-488 communication bus in the System Command mode.

NOTE: System Commands must be entered as upper case letters. An italic "A" or "a" represents any of the axes X,Y,U, and V. The letters "n", and "f" represent decimal numbers, 0 through 9. The symbol * represents an end-of-block.

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 12 after previous system command is executed.
D <CR> <LF>	: 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 12 in HOLD mode (Trigger required to execute programs). Hold mode cancelled by O <CR> <LF>.
I (string) * <CR> <LF>	: Execute program block (string) in the immediate mode.
J <CR> <LF>	: Set up Unidex 12 to send Service Request after execution.
K <CR> <LF>	: Cancel set up to send SRQ (default state)
L <CR> <LF>	: Go to Local with Remote enabled.
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)
PA <CR> <LF>	: Print <i>A</i> axis position register value
PD <CR> <LF>	: Print Directory listing
Pa <CR> <LF>	: Print <i>a</i> Encoder Position value
Pnn <CR> <LF>	: Print program # <i>nn</i>
P00 <CR> <LF>	: Print all programs in memory

PS <CR> <LF>	: Print Status bytes
Q <CR> <LF>	: Query (serial poll); Unidex 12 returns a byte
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
U <CR> <LF>	: Enable Joystick mode from Host, send RSQ on Limit
V <CR> <LF>	: Continued after programmed pause
W <CR> <LF>	: Complete move with Encoder Verification Error
X <CR> <LF>	: Quit program with Encoder Verification Error
Z <CR> <LF>	: Load Axis Calibration Table (with Axis Calibration option only)
 (hex 7F)	: Reset Unidex 12

MOTION PROGRAM COMMANDS

AB *	: Absolute Mode
AD nnn *	: Accel./Decel. ramp time in mil- liseconds
A4 *	: Enable velocity profile for <i>A</i> axis
B4 *	: Enable - limit for <i>A</i> axis
BXYUV *	: Enable - limit for all axes
CCA1vdddddddddd	
A2vdddddddddd	
I vdddddddddd	
J vdddddddddd *	
CO *	: Corner rounding mode
CWA1vdddddddddd	
A2vdddddddddd	
I vdddddddddd	
J vdddddddddd *	
C4 *	: Disable -limit for <i>A</i> axis
CXYUV *	: Disable -limit for all axes
DT *	: Disable tracking display
DWnnn.n *	: Dwell nn.n seconds
ET *	: Enable Tracking Display
E4 *	: High run current for <i>A</i> axis
EXYUV *	: High run current for all axes
F4 *	: Encoder verification for <i>A</i> axis
FXYUV *	: Encoder verification for all axes
Fli *	: Feedhold for Interrupt on line <i>i</i>

HA *	: Home A axis
HXYUV *	: Home all axes
IF fffff *	: Interpolation Feedrate
IN *	: Incremental Mode
IX vddddddddd	: 4 axis Linear move
Y vddddddddd	
U vddddddddd	
V vddddddddd *	
JS *	: Call Joystick
JA *	: Enable + limit for A axis
JXYUV *	: Enable + limit for all axes
KA *	: Disable + limit for A axis
KXYUV *	: Disable all axes + limit
LA vddddddddd *	: Load A position
LX vddddddddd *	: Load all axes position register
Y vddddddddd	
U vddddddddd	
V vddddddddd *	
MA vddddddddd *	: - limit for A axis
MX vddddddddd	: - limit for all axes
Y vddddddddd	
U vddddddddd	
V vddddddddd *	
NC *	: Non-Corner rounding mode
OR xxxx *	: Out/Run State
OS xxxx *	: Out/Stop State
PD (48 char) ^ *	: Print and Display message

PM (48 char) ^ *	: Display Message
PR (48 char) ^ *	: Print message
PA vddddddddd *	: + limit for A axis
PX vddddddddd	: + limit for all axes
Y vddddddddd	
U vddddddddd	
V vddddddddd *	
RD *	: Receive Message
RR *	: Reset all axes
RA *	: Start A axis
RXYUV *	: Start all axes
SSn *	: Send Service Request #n
SA *	: Stop A axis
SXYUV *	: Stop all axes
AF ffffff Dvddddddddd *	: Index for A axis
XF ffffff Dvddddddddd *	: Index for all axes
YF ffffff Dvddddddddd	
UF ffffff Dvddddddddd	
VF ffffff Dvddddddddd *	
AF ffffff R + *	: A Freerun
XF ffffff R -	: All axes Freerun
YF ffffff R -	
UF ffffff R -	
VF ffffff R +	
ZS *	: Disable velocity profile

I/O AND PROGRAM FLOW COMMANDS

AI nn I1 *	: Abort move on interrupt (i) and GoSub Label nn
BF *	: Beeper Off *
BI nn Ii *	: GoTo Label nn on interrupt (i)
BN *	: Beeper On
CI nn Ii *	: Abort move on interrupt (i) and GoTo Label nn
CP nn *	: Call Program #nn
CS nn Ivvvv *	: Conditional GoSub to Label #nn if Inputs = vvvv
CS nn Ldd *	: Conditional GoSub to Label #nn if 1 of 12 line #dd is Low
CS nn Hdd *	: Conditional GoSub to label #nn if 1 of 12 line #dd is High
CS nn Mx *	: GoSub to Label #nn if X is on Marker
CS nn MXYUV *	: GoSub to Label #nn if all axes on Marker
CS nn Bvvvvvvvvvvv *	: GoSub to Label #nn for Binary condition (vvvvvvvvvvv)
CT nn Ivvvv *	: Conditional GoTo to Label #nn if Inputs = vvvv
CT nn L dd *	: Conditional GoTo Label #nn if 1 of 12 line #dd is Low
CT nn H dd *	: Conditional GoTo to Label #nn if 1 of 12 line #dd is High
CT nn Mx *	: GoTo Label #nn if X is on Marker
CT nn MXYUV *	: GoTo Label #nn if all axes on Marker
CT nn Bvvvvvvvvvvv *	: GoTo to Label #n for Binary con- dition (vvvvvvvvvvv)

DB nnn *	: BCD Digital Output (nnn)
DD nnnn *	: Decimal Digital Output (nnnn)
DG *	: GoTo Label specified by DIO
DL dd *	: Set 1 of 12 Digital Output line #dd, Low
DH dd *	: Set 1 of 12 Digital Output line #dd, High
DI vvvv *	: Disable Interrupt for Inputs vvvv
DS *	: GoSub specified by DIO
GP nn *	: GoTo Program #nn
GS nn *	: GoSub Label #nn
GT nn *	: GoTo Label #nn
IT vvvv *	: Wait till Input vvvv
LB nn *	: Label #nn
LI vvvv *	: Enable Interrupt for Inputs
OT vvvv *	: Output vvvv
PS *	: Program Stop
RC vvvv *	: End Repeat Loop on Input condi- tions (vvvv)
RE *	: Repeat Loop End
RP *	: Repeat Program
RS nnnn *	: Repeat Loop Start nnnn times
SI nn Ii *	: GoSub Label #nn for Interrupt on Input i
SR *	: Subroutine Return
* or /	: End of Block (terminates Block)
%	: End of Edit (Downloading)

Appendix 2 : IEEE-488 BUS INTERFACE FUNCTIONS

The following are messages sent on the bus with the ATN line asserted. The bus functions listed below are valid only when Unidex 12 is in the Interface Active mode or the Communication Enabled mode. If Unidex 12 is in the Local mode, bus handshake protocol may not be completed.

BUS FUNCTION	UNIDEX 12 RESPONSE
DCL (Not addressed)	Goes into Power-up state (Unidex 12 requires about 2 seconds to come out of reset).
SDC (Addressed)	(Same as Above)
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)	These two components of a Serial Poll cause
SPD (Addressed)	Unidex 12 to put out the status byte and reset SRQ if set.
UNL	These are components of Data Transfers. They cause appropriate data transfer sequences
UNT	
MLA	
MTA	
LAG	
TAG	

IDY (Identify)
(Parallel Poll)

Puts the identity on the bus
as a response to a parallel poll;
transparent to Unidex 12.

TCT (Addressed)

Bus hangs up

LLO

No response

BUS FUNCTION

UNIDEX 12 RESPONSE

UUCG (Undefined Universal
Command Group)

No response

UACG (Undefined Addressed
Command Group)

No response

PART II
JOYSTICK

CHAPTER 1: INTRODUCTION

The Joystick option allows the Host computer controlling the Unidex 12 to enable it for Joystick operation or Remote (external) control through the Auxiliary inputs. Enabling Joystick or Remote control may be done through either Communication Interface RS-232C or IEEE- 488. Manual operation of the keyboard is no longer necessary to put Unidex 12 into the Joystick mode of operation, thus providing very efficient interactive control capability with a Host controller.

SECTION 1-1 REQUIRED HARDWARE

A Unidex 12 Joystick and a Communication Interface are required to implement this option. See the *Unidex 12 Motion Controller Programming Manual* for information on the RS-232 Interface, or Part I of this manual for information on the IEEE-488 Interface. Refer to the *Unidex 12 Motion Controller Hardware Manual* for hardware information.

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 2: OPERATION

SECTION 2-1 JOYSTICK CONTROL

The Unidex 12 Joystick can be enabled and disabled by simple system commands from the Host controller through the Communication Interface.

2-1-1 ENABLING THE JOYSTICK

The system command:

S <CR> <LF>

enters the Unidex 12 into the Joystick mode. The display will change to the Joystick mode Tracking display (Part II, Chapter 2 of the *Unidex 12 Motion Controller Programming Manual*).

jkm	X: 0000023456	step
jk	Y: -0002345.12	mm

In a system having more than two axes, the initially active axes will be X and Y. The Joystick may now be operated to move the axes.

When an axis encounters a limit, Unidex 12 indicates the condition by beeping and flashing the "Axis in Limit" message on the display. Reversing direction of the axis with the Joystick will remove the limit condition.

2-1-2 BUTTONS A AND B

The active axes pair may be changed by pressing Button A.

Pressing Button B will decrease the Joystick activated speed by a factor of 64. Pressing Button B a second time will return the axis speed to the original speed. (Refer to Part II: Chapter 2 of the *Unidex 12 Motion Controller Programming Manual*, for a detailed description of Buttons A and B.)

2-1-3 BUTTON C

Depressing Button C will initiate one of two different actions on the part of Unidex 12:

2-1-3-1 UNIDEX 12 IS IN SERVICE REQUEST MODE

(For information on Service Request and Serial Poll, refer to Part IV: Chapter 3, of the *Unidex 12 Motion Controller Programming Manual* for RS-232, or Part I: Chapter 3 of this manual for IEEE-488.

If Unidex 12 is in the Service Request Mode and Button C is pressed, a Service Request will be asserted as follows:

When using the RS-232C interface, this is implemented by sending a character from Unidex 12 and waiting for a Serial Poll command

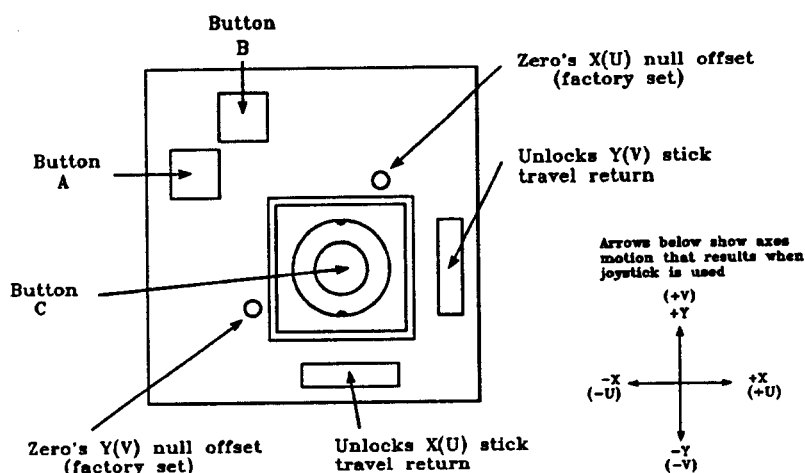
"Q <CR> <LF>".

When using the IEEE-488 interface, Service Request is a primary Bus function and Unidex 12 asserts the SRQ line and then waits for a Serial Poll by the controller.

When Unidex 12 asserts a Service Request, the Host controller is required to do a Serial Poll. After the Serial Poll, the Host must disable the Joystick prior to sending any system commands (except the Print Position commands).

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 12 asserts an SRQ. The device address of the Unidex 12 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 12 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 (optional)
260 ON INTR 7 GOSUB 200	! Return from Interrupt
@Enable INTR 7; 8 @	
Return	

2-1-3-2 UNIDEX 12 IS IN THE SERVICE REQUEST CANCEL MODE

When Unidex 12 is in the Service Request Cancel Mode, pressing Button C will cause Unidex 12 to send the Absolute axis position register values to the controller. Each position value is 12 characters (a space or a minus sign followed by 10 digits), terminated by a < CR > < LF > . The output from Unidex 12 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.

When using 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.

When using the IEEE-488 Interface, the Host is required to address Unidex 12 to talk before either of them can send data. Pressing Button C will cause Unidex 12 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 12
60 DISP X;Y;V	! Display data
70 GOTO 50	! Wait for next Button C
	! operation

2-1-4 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 prior to control being returned.

When the Joystick is active following an "S" command, the Unidex 12 will only recognize the "D" command, the Serial Poll command, and the "P" Print commands. All other system commands are ignored.

Position Register Values may be read with the "PX", "PY", "PU" and "PV" commands without disabling the Joystick.

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 12 unit. (Refer to the *Unidex 12 Motion Controller Programming Manual*, Part II: Chapter 2 and the *Unidex 12 Motion Controller Hardware Manual*, for more information.)

2-2-1 ENABLING REMOTE MODE

The system command:

R <CR> <LF>

enables the Unidex 12 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 12. The Unidex 12 keeps track of the axis positions during external control.

2-2-2 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 the Remote mode is active after an "R" command, Unidex 12 will only recognize the "D" command and the Serial Poll command. All other system commands will be ignored.

2-2-3 ENABLE JOYSTICK

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

a) Unidex 12 is forced into the SRQ mode

The three Joystick buttons are functionally the same as when the Joystick is enabled by the "S" command.

b) When an axis runs into a limit:

- The Joystick is disabled. (Unidex 12 automatically executes a "D" command).
- The clock pulses sent to the axes are shut off.
- The position registers are updated.
- Unidex 12 performs a service request.

The controller may now Serial Poll Unidex 12 to receive an error status and return Unidex 12 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 position registers may be read using the print position commands, "PX", "PY", "PU" and "PV".

Unidex 12 remains in the SRQ mode.

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

- Unidex 12 is put into the Joystick mode again, or
- When a motion command is executed.

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

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 3: DIGITIZING (RS-232 INTERFACE ONLY)

Joystick Digitizing provides the ability to obtain and remember one or more sampled axis positions or moves. Digitizing allows 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 12 Motion Controller Programming Manual*).

During the digitizing process, axis selection (X/Y or U/V) and speed range (Hi/Low) is set by the use of two buttons on the Joystick. The third button causes the position sample to be taken and recorded when pressed (see Part II: Chapter 2 of the *Unidex 12 Motion Controller Programming Manual*). All axis positions are displayed on the liquid crystal displays (LCDs) while digitizing.

A thorough understanding of the *Unidex 12 Motion Controller Programming Manual* (Part IV), is suggested prior to utilizing Joystick Digitizing and the RS-232.

SECTION 3-1 REQUIRED HARDWARE

A Unidex 12 Joystick and RS-232 Interface are required to operate Joystick Digitizing. (Refer to the *Unidex 12 Motion Controller Hardware Manual*.)

SECTION 3-2: OPERATION

3-2-1 INITIAL SETUP

Initial set-up for digitizing is done prior to entering the Digitizing mode. The set up depends on whether the digitized points are to be stored in Unidex 12 as a program or transmitted externally via RS-232.

If the digitized points will be stored in the Unidex 12, the available free memory space should be checked and additional space made free if more space is required.

If RS-232 is to be used to transmit the digitized points, refer to the set up requirements detailed in the *Unidex 12 Motion Controller Programming Manual*, Part IV.

If the "Start" position for digitizing is the Home position, or at another known position, the axis should be positioned at that point before entering the Digitizing mode. At the completion of the Initial set up, the user is ready to enter the Digitizing mode.

SECTION 3-3 DIGITIZING SET-UP

3-3-1 DIGITIZE SELECT

Entry into the Digitizing mode is through the Joystick screen (refer to the *Unidex 12 Motion Controller Programming Manual*). Selection of the Joystick mode (key #5), will display the following screens:

1. X/Y axis	3. DIVISOR
2. U/V axis	4. SEL PGM

5. Digitize

Press key #5 to select Digitize.

3-3-2 STORE/TRANSFER SELECT

After pressing key #5 of the Joystick Screen, the Store/Transfer Screen will be displayed. If RS-232 is to be used, the external device should be in the "Ready" mode:

- | |
|--|
| <ol style="list-style-type: none">1. Store in User Memory2. Transfer via RS-232 |
|--|

If key #1 is pressed, the Digitizing points will be stored in the Unidex 12 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.

3-3-3 PROGRAM # SCREEN

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

<p>Input Program</p> <p>00</p>

To enter the program #, key in 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 12 memory, depending on the Store/Transfer set-up (Item 3-3-2 of this Chapter). If the program is to be stored in the Unidex 12 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.

3-3-4 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.

3-3-5 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 Part II of the *Unidex 12 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 enter the Digitizing mode.

3-3-6 DIGITIZING TRACKING DISPLAY SCREEN

The Digitizing screen is continuously 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 12 Motion Controller Programming Manual*, Part II.

jd X : 0000000000 step jd Y : 0000000000 step
--

At this time, the joystick is active and digitizing is operative. See Item 3-3-7 of this manual for joystick operation. During this time, the following keys on Unidex 12 are active:

INSERT	Displays again the Digitized Feedrate screen (Item 3-3-5 of this Chapter) to allow feedrates to be changed for upcoming digitized blocks.
DELETE	<i>CLEAR ABSOLUTE POSITION.</i> (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	<i>QUIT.</i> See Item 3-3-8 of this Chapter.

3-3-7 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.

3-3-8 DIGITIZING TERMINATION

To terminate the Digitizing, press the BACK key when in the Digitizing Tracking Display screen (Item 3-3-6). When the BACK key is pressed, Digitizing will be terminated and Unidex 12 will transmit the End of Program characters if RS-232 is active (refer to Part IV of the *Unidex 12 Motion Controller Programming Manual*) The Unidex 12 will then go back to the Joystick screen. Pressing BACK again will send Unidex 12 to the second Main Menu screen (refer to Part II of the *Unidex 12 Motion Controller Programming Manual*).

SECTION 3-3: DIGITIZING FORMATS

3-3-1 STORE FORMAT

A brief sample program is shown below (in Unidex 12 "Menu" format) illustrating how sampled points are 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 (Items 3-3-3 through 3-3-6 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).

3-3-2 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 12 Motion Controller Programming Manual*.

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 4: TROUBLESHOOTING

Troubleshooting is divided into two sections, Software Malfunctions and Hardware Malfunctions. Software malfunctions are problems displayed as error messages or codes on the Unidex 12 display. Hardware malfunctions are problems that do not provide error messages or codes on the Unidex 12 display.

SECTION 4-1 SOFTWARE MALFUNCTIONS

Software malfunctions are displayed on the Unidex 12 display in the form of error codes. All error messages are described in the *Unidex 12 Motion Controller Programming Manual*, Part II.

SECTION 4-2 HARDWARE MALFUNCTIONS

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

- Joystick Control
- RS-232 Control (if used)
- Unidex 12 Editor and Memory

4-2-1 JOYSTICK CHECK

Joystick operation can be checked by going into Joystick operation without digitizing. Refer to the *Unidex 12 Motion Controller Programming Manual* (Part II). If operation is satisfactory, the Joystick is most likely not causing the problem and the next related area should be checked.

4-2-2 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 12 Motion Controller Programming Manual*, Part IV. If the RS-232 is working correctly check the next related area.

4-2-3 EDITOR AND MEMORY CHECK

The third section to check is the Editor and Memory. To check memory, see Part II, of the *Unidex 12 Motion Controller Programming Manual*. To check the Editor operation, key a small sample program into the Unidex 12 user memory. Review the program and verify that the program is present and all blocks are valid.

PART III

AXIS CALIBRATION

CHAPTER 1: AXIS CALIBRATION : INTRODUCTION

The Axis Calibration Option for the Unidex 12 provides the ability to correct static errors in the mechanical components of a positioning system using the software of the Controller. For example : A ball screw having an accuracy specification of 25 micron/100 mm may be calibrated by the use of a Laser Interferometer System. A correction table may then be stored in the Unidex 12 and during positioning the accuracy may be improved to 5 micron/100mm..

If the Unidex 12 is equipped with the Axis Calibration Option, upon power up the screen will display "/AC". This option must be specified at the time of purchase.

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 2: OPERATION

SECTION 2-1: INTERNAL COMPUTATIONS

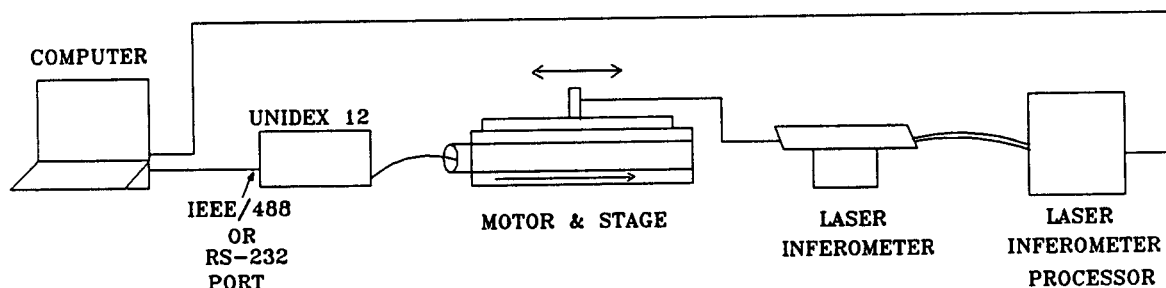
The difference between the distance (as determined by a Laser Inferometer System or equivalent) actually traveled for a programmed number of machine steps and the theoretical distance, is calculated and recorded in table form within the Unidex 12.

When subsequent positioning is performed by the Unidex 12, the correction value from the table within the Unidex 12 is added to the programmed distance. This results in a move end point devoid of Static Error. (See Chapter 3 for a sample Axis Calibration Table)

The Calibration Tables are stored in EEPROMS (Electrically Erasable, Programmable Read Only Memory) and are retained during power down. Two kilobytes of memory per axis is available for Calibration Tables. Each Axis Table may have a maximum of 1000 correction values for 1000 discrete positions of the axis.

SECTION 2-2: SET UP

A Calibration Table can only be generated for the Unidex 12 through one of the Communication Ports. An example of a Set Up required to enter a Calibration Table for a Positioning Table is shown below:



SECTION 2-3: OPERATION

For illustrative purposes, assume the Positioning Table is to be calibrated every 1 mm over a travel of 100 mm. The system resolution is 0.1 micron.

The computer commands the Unidex 12 to move the axis to the Home position. This position is established as the Home reference and is used to zero the Laser Interferometer System.

The Unidex 12 is then commanded to incrementally move the positioning table until the Laser Interferometer indicates 1.0000 mm. The computer then reads the absolute position from the Unidex 12, for example 9998 steps. The error is calculated to be - 2. At 2mm the Unidex 12's absolute position may be 20004 steps, the error then being +4. These error values are configured into a table and stored in the computer.

SECTION 2-4: LOADING CALIBRATION TABLES INTO THE UNIDEX 12

The computer must send the following command to load the previous Calibration Table into the Unidex 12.

ZX,10000,0,-2,4,.....,%

Z	the command to start the Axis Calibration Table.
X,	Axis name (Y,U,V)
10000,	Calibration Distance
0,	the first recorded value is always zero
%	End of Table

After the table is loaded, the Unidex 12 must be reset with either the <CR> <LF> or the (Hex) 7F command. This initializes Axis calibration.

SECTION 2-5: CHECKSUM

Upon a subsequent power up, the Calibration Table is verified for accuracy by a Checksum verification. If an error is detected the display will show:

<p>X AXIS CALIBRATION INVALID Press SELECT for a Menu</p>
--

Axis correction during positioning is automatically disabled.

SECTION 2-6: FACTORS INFLUENCING AXIS CALIBRATION

The user of the Axis Calibration Option should be aware of some of the factors affecting the Axis Calibration process and it's effectiveness.

2-6-1 TEMPERATURE (Thermal Expansion/Contraction)

Measurements are performed within a controlled ambient temperature. The mechanics of the system (Motor, Ball Screw, Positioning Table Base, etc.) are also at controlled ambient temperature. Thermal expansion or contraction will produce erroneous results and inaccuracies, therefore the Calibration process as previously described, must be done under conditions as similar to the actual working conditions as possible.

2-6-2 FIXTURING

The fixturing of the Positioning System during calibration must be identical to that of the working system. It is preferred that Calibration take place after the final installation of the Positioning system at the work place.

2-6-3 LINE OF MEASUREMENT

Every Positioning Table has mechanical imperfections contributing to the Yaw and Pitch of the Table top which are reflected in the measurements.

The Line of Measurement (distance) relative to the Positioning table, of the Laser beam to the Ball Screw (as measured from an edge of the table) must remain constant.

CHAPTER 3: EXAMPLE: AXIS CALIBRATION TABLE

The following is an example of Axis Calibration Data:

AXIS CALIBRATION TABLE - HAL:2000

LABEL ON EEPROM: _____ (Table on Page 2)

CUSTOMER ORDER # _____ AXIS _____

STAGE MODEL & SERIAL # _____

MOTOR MODEL & SERIAL # _____

DRIVE MODEL & SERIAL # _____

LASER INTERFEROMETER MODEL # _____

OTHER PARAMETERS

1. Torque on Bolts _____

2. Air Temp. _____

3. Motor Case Temp. _____

4. Motor Flange Temp. _____

5. Stage Base Temp. _____

6. Stage Far End Temp. _____

7. Flatness Spec. of Supporting Table _____

8. Calibration Distance _____

9. No. of Calibration Points _____

10. Distance from Ball Screw to Measurement Axis (Vertical) _____

11. Distance from Ball Screw to Measurement Axis (Horiz.) _____

12. Max. Operating Speed _____

13. Max. Duty Cycle _____

- NOTES:
1. Any frictional heating of the Ball Screw is to be avoided to maintain specified accuracies.
 2. Thermal expansion characteristics of the Ball Screw are as specified in the Data sheet on Page 3.

Calibration Done By _____ Date _____

CALIBRATION TABLE FOR _____
 NO. OF CALIBRATION POINTS _____
 DISTANCE BETWEEN POINTS _____
 CALIBRATION RESOLUTION _____

00, 05, 09, 07, 06, 09, 13, 14, 17, 19, 18, 17, 20, 24, 24, 24, 26, 28, 27, 26, 28, 30, 32, 31, 30,
 33, 36, 36, 37, 38, 40, 38, 36, 38, 40, 39, 44, 48, 52, 51, 49, 49, 48, 46, 44, 47, 51, 49, 47, 48,
 49, 47, 45, 48, 52, 50, 48, 48, 49, 47, 44, 47, 51, 49, 47, 48, 50, 48, 46, 49, 52, 50, 48, 49, 50,
 49, 48, 51, 54, 52, 49, 49, 49, 49, 50, 52, 50, 48, 49, 51, 50, 48, 51, 54, 51, 49, 49, 49, 47, 46,
 50, 54, 51, 48, 47, 46, 46, 45, 48, 51, 49, 47, 48, 49, 47, 45, 45, 45, 46, 47, 45, 44, 43, 44, 46,
 45, 44, 47, 50, 47, 45, 45, 45, 44, 43, 46, 49, 47, 44, 44, 43, 42, 41, 42, 43, 41, 39, 39, 39, 38,
 37, 38, 39, 35, 31, 34, 37, 35, 33, 35, 37, 35, 33, 33, 33, 31, 29, 31, 33, 31, 29, 30, 31, 30, 30,
 31, 32, 29, 26, 27, 28, 27, 27, 28, 29, 27, 24, 24, 25, 24, 23, 23, 24, 22, 20, 20, 21, 19, 18, 19,
 20, 20, 20, 19, 19, 19, 19, 18, 18, 16, 15, 15, 16, 15, 15, 15, 15, 14, 13, 13, 12, 12, 12, 12, 12,
 12, 12, 11, 11, 11, 10, 9, 9, 8, 8, 8, 8, 7, 5, 6, 6, 5, 5, 5, 3, 1, 1, 2, 1, 0, 1,
 2, 1, 0, 0, 0, 0, 0, 0, 0, -1, -1, -2, -3, -4, -4, -3, -3, -4, -4, -5, -5, -5, -4, -4, -5, -6,
 -6, -5, -5, -5, -5, -4, -7, -9, -9, -8, -9, -11, -10, -9, -10, -11, -12, -14, -14, -13, -12, -11,
 -13, -15, -17, -20, -19, -18, -17, -16, -19, -22, -21, -21, -21, -20, -19, -20, -21, -22, -24, -21,
 -19, -17, -16, -18, -19, -20, -21, -20, -19, -18, -17, -19, -1, -21, -22, -20, -19, -18, -17, -18,
 -19, -20, -22, -20, -18, -16, -14, -21, -28, -28, -28, -21, -14, -15, -16, -22, -28, -26, -25, -20,
 -16, -16, -15, -21, -27, -24, -22, -18, -14, -14, -15, -21, -27, -24, -22, -18

Calibration Done By _____ Date _____

THERMAL EXPANSION OF BALL SCREWS @ VARIOUS TEMPERATURES
 BASED ON 7×10^{-6} IN./IN. ° F THERMAL COEFFICIENT

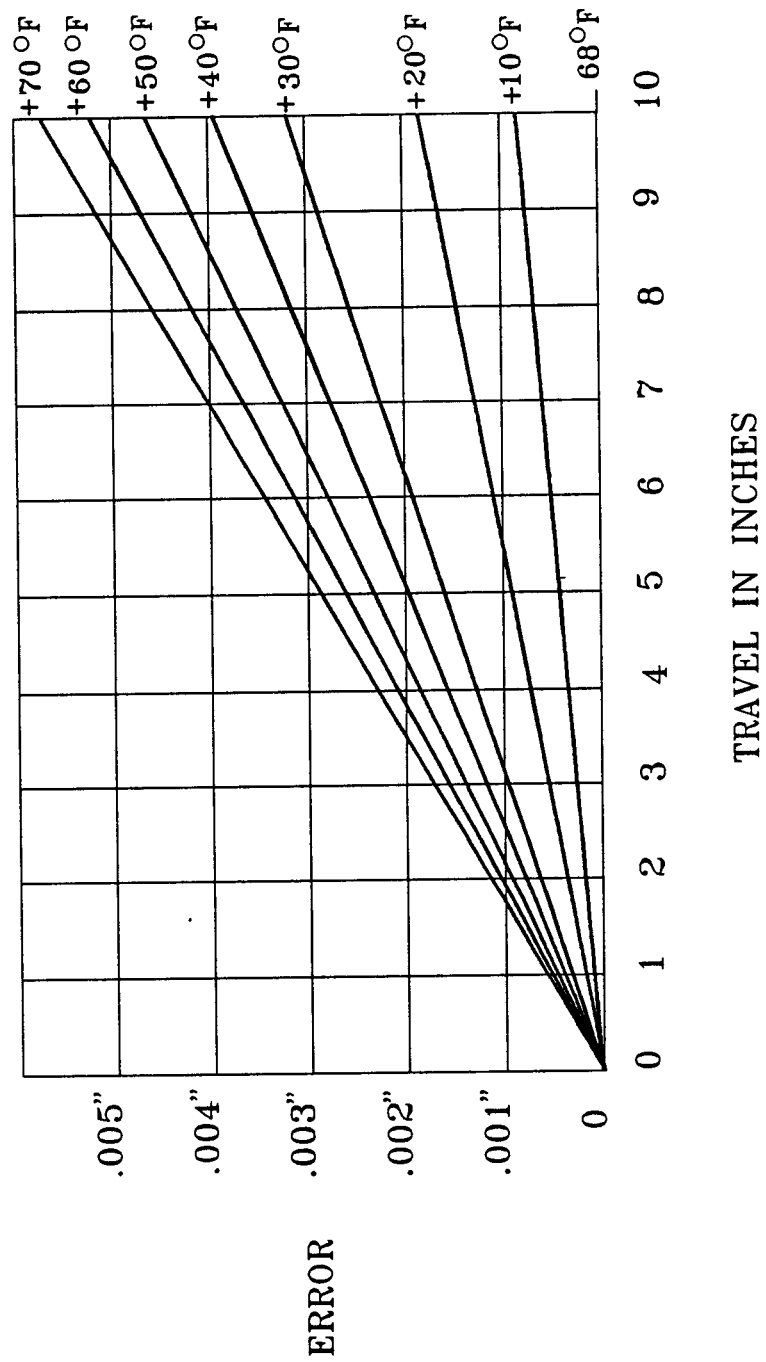


Figure 3-1: Example: Thermal Expansion

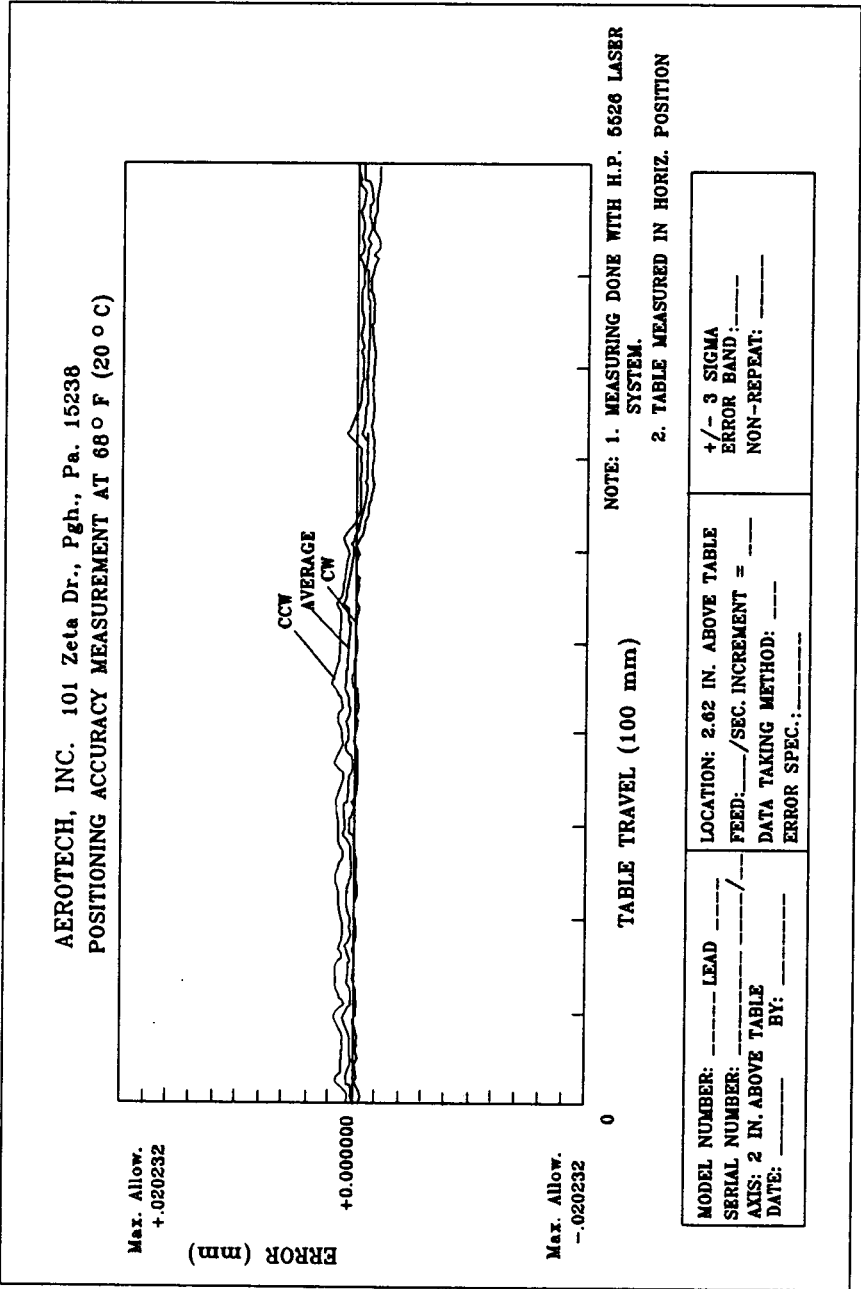


Figure 3-2 : Example: Error vs Travel Plot

PART IV

TWO DIGIT

THUMBWHEEL

CHAPTER 1: THE TWO DIGIT THUMBWHEEL

SECTION 1-1: INTRODUCTION

The Two Digit Thumbwheel Switch Assembly is a simple control device used to initiate programs that have been previously entered and stored in the Unidex 12. (Set for the Input mode – see Figure 4-2.)

SECTION 1-2: OPERATION

Connect the Two Digit Thumbwheel Switch Assembly (TDT) to the Digital I/O connector located on the rear of the Unidex 12 (see *Unidex 12 Motion Controller Hardware Manual*). Power up the Unidex 12. Set the Thumbwheel Switches to read "00". Press the TDT's STROBE Button and release. The TDT's READY LED will come ON. Unidex 12 is now configured to run programs called by the TDT.

Use the TDT's Thumbwheel Switches to indicate the desired program number. Press and release the STROBE Button.

The READY LED will go OFF and the Unidex 12 will execute the program if it exists in memory. After executing the program, the READY LED will come ON.

If an error condition occurs because the program called does not exist, or if there is a run-time error in the program, the Unidex 12 will cause the TDT's ERROR and READY LEDs to come ON simultaneously. Another program may be selected at this time.

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

NOTE: Refer to the *Unidex 12 Motion Controller Programming Manual* (Input Mode) for further details on running a program from an external device.

An electrical schematic diagram of the TDT Option is shown in Figure 4-1.

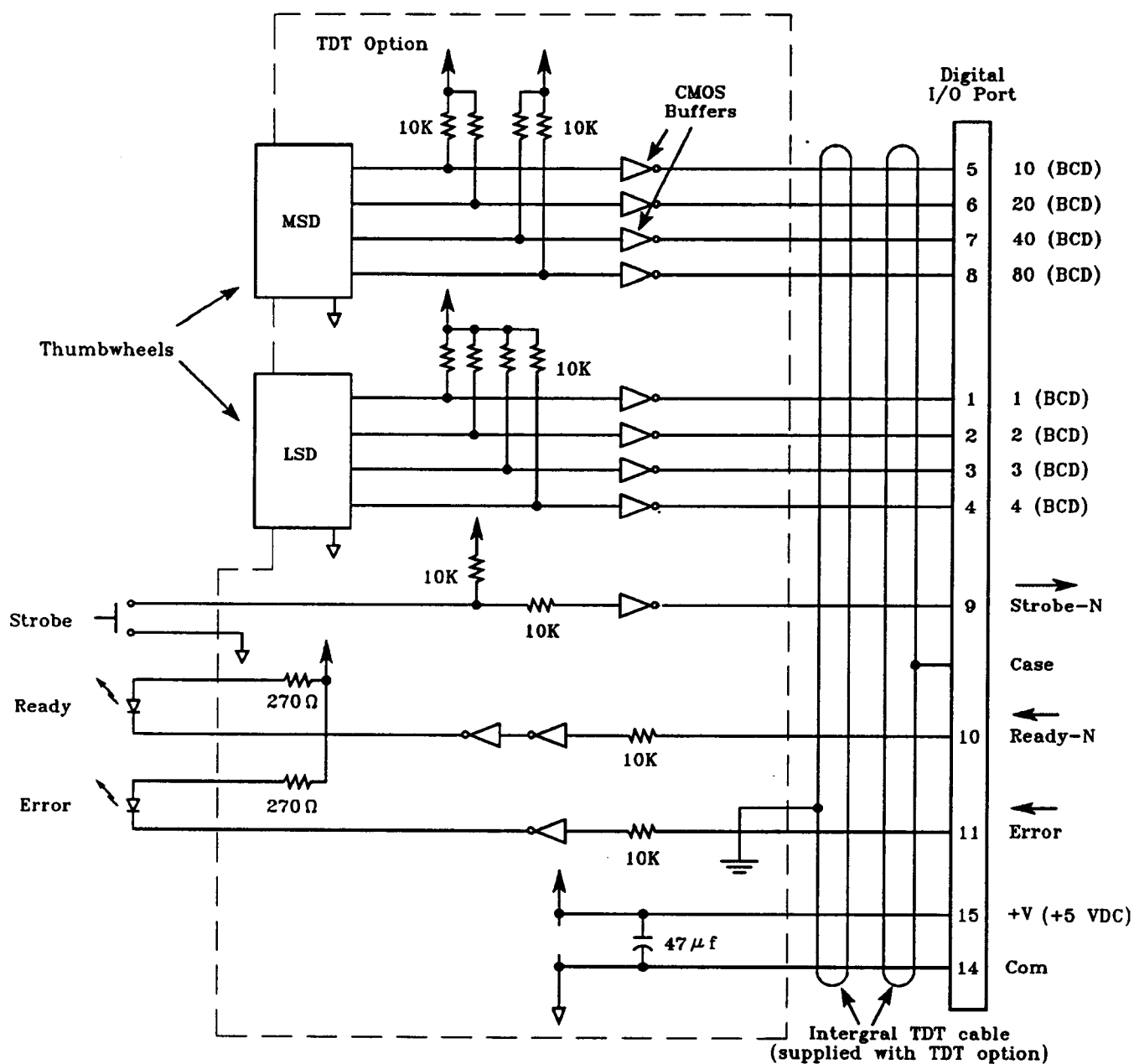


Figure 4-1: Two Digit Thumbwheel Switch Schematic

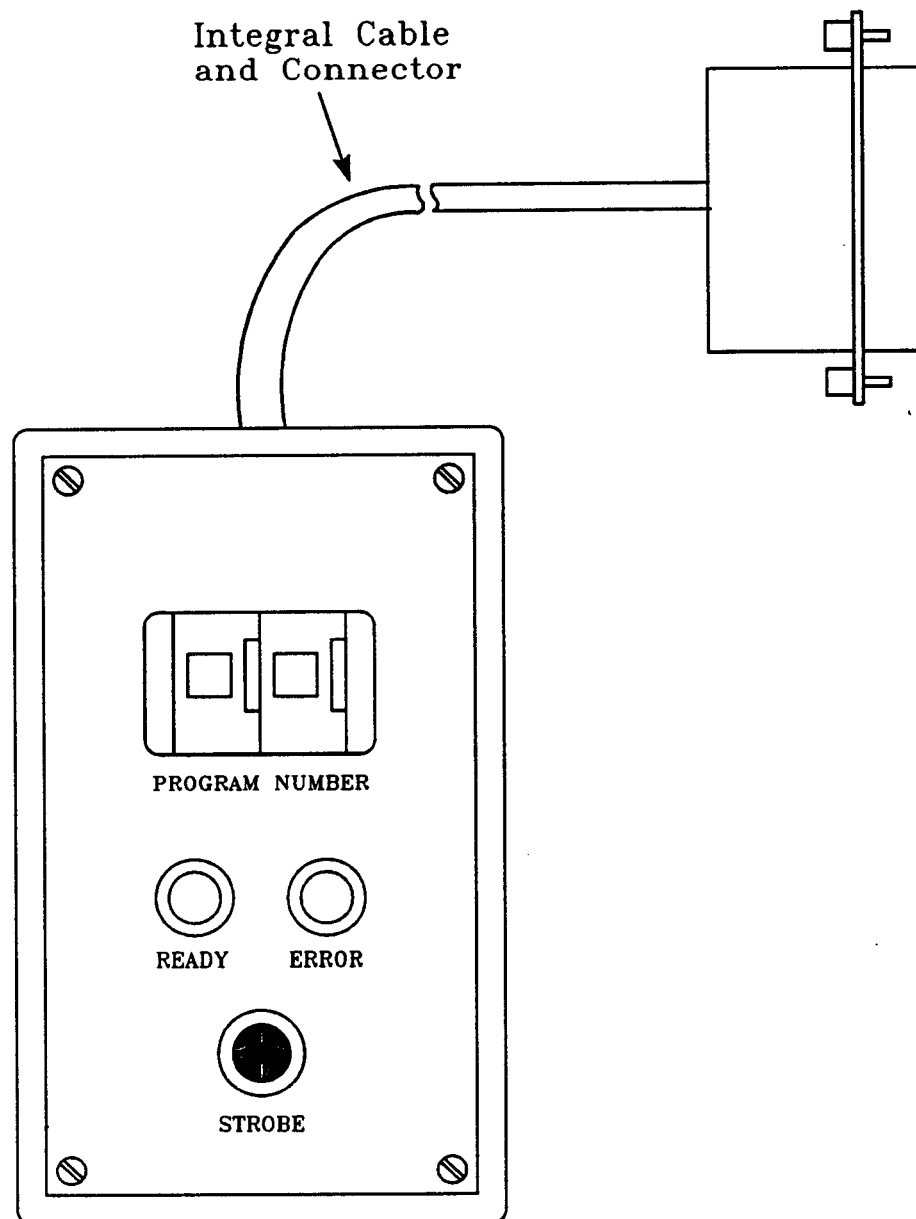


Figure 4-2: Two Digit Thumbwheel Switch Assembly

PART V
HIGH SPEED
BINARY

CHAPTER 1: INTRODUCTION

The Unidex 12 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 in binary format reducing the computational overhead time of the UNIDEX 12 between moves.

SECTION 1-1: HARDWARE INTERFACE

The High Speed Binary Interface is accessed through a 15 pin D-connector located at the rear of the Unidex 12 (see the *Unidex 12 Motion Controller Hardware Manual* for more information).

The pin configuration of this connector is shown in Figure 6-1 which also shows the hardware required to interface the Unidex 12 to a Metra-Byte PDMA-16 Digital I/O Board. No part 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 12 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 12, 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 12 then asserts Unidex 12 Ready (U12 RDY) line LOW after accepting the data. The Host may now write the next byte in the same manner.

When reading from Unidex 12, the Host negates the Host Ready (HOST RDY) line to a HIGH and waits for the U12 RDY line to be asserted LOW and, after accepting the data, asserts HOST RDY line LOW.

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 2: PROGRAMMING

SECTION 2-1 PROGRAMMING THE BINARY INTERFACE

Upon power up, the Unidex 12 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 12's menu screens.

1. COMM ENAB 3. PRINT
2. SET UP

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

1. RS-232/IEEE-488
2. TDT/HSB

Press key #2, Parallel/Binary, and Unidex 12 is ready to communicate through the Parallel port. You will see:

PARALLEL INTERFACE
COMMUNICATION ENABLED

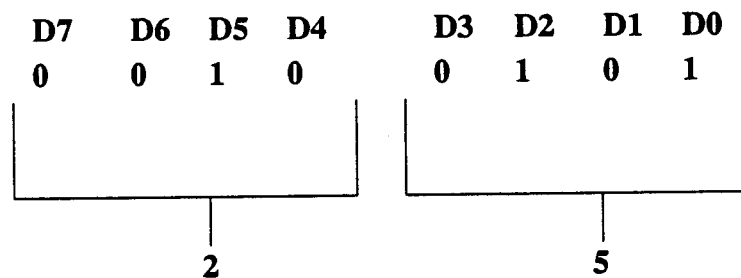
The parallel interface has two modes, the High Speed Binary (HSB) mode, and the Programmable Logic Controller (PLC) mode. Unidex 12 may be set to these modes of operation by writing the following ASCII codes. Writing to Unidex 12 is done by placing the data on the data lines and asserting HOST RDY low.

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

In the HSB mode the Unidex 12 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 2-2 : PROGRAMMABLE LOGIC CONTROLLER

The Programmable Logic Controller (PLC) mode allows a program contained within the Unidex 12's memory to be executed by writing the BCD program number into the Unidex 12. The program number is placed on the data lines D7 - D0, and HOST RDY is asserted low. Example (Program #25):



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

To return control to the keyboard, the Unidex 12 must be set to the Local/Comm. Enabled mode. This may be done only by setting the Unidex 12 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).

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 3: HIGH SPEED BINARY MODE

The High Speed Binary mode allows all of the RS-232 Immediate mode commands to be executed (see Chapter 6 of this part of the manual). It also allows the position and status to be read from the Unidex 12, the Unidex 12's LCD Tracking display to be updated, the Joystick to be enabled and disabled, and the remote clock and direction to be tracked (Refer to Chapter 4 of this part of the manual).

The commands are in a binary language having 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 the command block that follows. This requirement is due to the Unidex 12 waiting for a specified number of bytes, and not relying on microprocessor interrupt service routines. 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): U axis 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): X axis 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): Y axis feedrate and distance (7 bytes follow)
 79 (hex): Y axis distance only (4 bytes follow)

To specify distance use a 4 byte word representing the move in steps; such as a two's complementary binary number. To specify the feedrate use 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 μ sec)	8 bytes
Y - 5,000 steps at 1,000steps/sec(1000 μ sec)	8 bytes
U - Free run CCW at 10,000 steps/sec (100 μ sec)	<u>4 bytes</u>
Total	= 20 bytes

will 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 12 will begin pre-move computations immediately after receiving the number of bytes specified. Upon completion of these computations, Unidex 12 will wait for the "T" or trigger character, shown above as 54 hex. A feedrate of zero (00 00 00) will stop that axis from free running.

The IN-POS line will be taken to the logical low state as the trigger character is received. This means that the IN-POS line will go LOW as the trigger character is received but before the U12 RDY line is asserted, indicating acceptance of the byte. The IN-POS line will be asserted HIGH at the end of the move.

During this period, when the IN-POS line is LOW indicating a move is in progress, no other motion blocks may be transferred to the Unidex 12. This may be seen by the logic used for DMA transfers to the Unidex 12, shown in Figure 6-1. The motion in progress may be interrupted and aborted by writing to Unidex 12 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
D6	D7	D5	D4	D3	D2	D1	D0	
1	0	0	0	0	1	0	0	Stop U And Reset All Drives
1	0	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.

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 4: COMMUNICATING WITH UNIDEX 12

In the High Speed Binary mode, the HOST may also read data from Unidex 12. A 3-byte status may be read by writing to Unidex 12 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 12 the ASCII character "P" (50 hex or 80 decimal) or "p" (70 hex or 112 decimal).

SECTION 4-1: STATUS

To request status data from the Unidex 12 the HOST must write ASCII "S" (53 hex or 83 decimal), or ASCII "s" (73 hex or 115 decimal). Unidex 12 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 of Part V of this manual for a description of these bytes. See below for Handshake sequence of "S" or "s" entries.

SECTION 4-2: AXIS POSITION

To request axis positions from the Unidex 12, 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 12 will then output the bytes one at a time, asserting U12RDY 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 example would result in the following data from Unidex 12:

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

Unidex 12 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" (Figure 6-3)

HOST	UNIDEX 12
1. Places "S" or "P" on the data bus.	
2. Asserts HOSTRDY low. →	Reads "S" or "P" and asserts U12RDY 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 U12RDY high.
	Asserts IN-POS low.
6. Waits for U12RDY to go low to read data ←	Places first byte of data on the bus.
	Asserts U12RDY low
7. Reads Data Byte. ←	Waits for HOSTRDY low transition.
8. Asserts HOSTRDY low. →	Negates U12RDY high.
9. Negates HOSTRDY high.	
10. Waits for IN-POS to go high. ←	Changes Data Port to INPUT.
	Asserts U12RDY low.
	Negates IN-POS high.
11. Changes Data Port to OUTPUT	

Host can now send data to Unidex 12.

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

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

HOST		UNIDEX 12
1. Places "s" or "p" on the Data Port		
2. (HOSTRDY low when data valid)	→	Reads "s" or "p" and asserts U12RDY low.
3. Configures DMA interface for reading data. Sets AUX3 on MBI to enable DMA transfer		Changes Data Port to OUTPUT, Negates U12RDY high
4. Waits for U12RDY to go low	→	Places data byte on Port and asserts U12RDY low
5. Negates HOSTRDY high.		
6. Reads data byte		Waits for HOSTRDY low
7. (HOSTRDY low when data read)	→	Negates U12RDY high.
8. Configures DMA interface for writing data		Changes Data Port to INPUT

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

SECTION: 4-3 TRACKING DISPLAY

When updating the Unidex 12's LCD Tracking display to the current absolute position, the Host must write the character "D" (44 hex or 68 decimal) to Unidex 12. Unidex 12 then asserts U12RDY true, and updates 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.

SECTION: 4-4 ENABLING/DISABLING JOYSTICK

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

4-4-1 Enabling Joystick: Command "J"

The character "J" (decimal is 74 and hex is 4A) puts the Unidex 12 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 12 cannot enter the Joystick mode, the ERROR LINE along with the IN-POS LINE is asserted high.

Button A on the Joystick toggles between the X/Y and the U/V axis pairs. Button B on the Joystick toggles between normal and slow Joystick clock rate. The ratio between the two rates is 64.

Button C calls the attention of the computer. When Button C is pressed, the Joystick is disabled and absolute position registers are updated. Unidex 12 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 encounters a limit while using the Joystick, Unidex 12 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 had run into a limit at any time during the Joystick mode.

While in the Joystick mode the only command that the Unidex 12 will accept and respond to is the character "K" which disables the Joystick.

4-4-2 Disable Joystick: Command "K"

The character "K" (decimal is 75 and hex is 4B) may be written to cancel the Joystick mode. Unidex 12 disables the Joystick, updates the absolute position registers, and displays the screen:

PARALLEL INTERFACE COMMUNICATION ACTIVE
--

The Unidex 12 is now ready to receive further commands.

SECTION 4-5: ENABLE/DISABLE REMOTE CLOCK AND DIRECTION

Character "R" (decimal is 82 and hex is 52) sets up Unidex 12 to track remote clock and direction signals that may be applied to inputs provided on the rear panel connectors (refer to the *Unidex 12 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 the Remote mode will not be detected by reading the status. However, the limits are still functional and will prevent over-travel. The limit signals are available for user interface on the rear panel of the Unidex 12 (refer to the *Unidex 12 Motion Controller Hardware Manual*).

Command "K" (decimal 75 and hex 4B) is used to exit this mode. Unidex 12 disables the remote clock and direction inputs and updates the position registers before returning to receive further commands.

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 5: 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 runs into a limit.

The IN-POS line indicates Command Complete status and the Unidex 12 is ready for the next index or program.

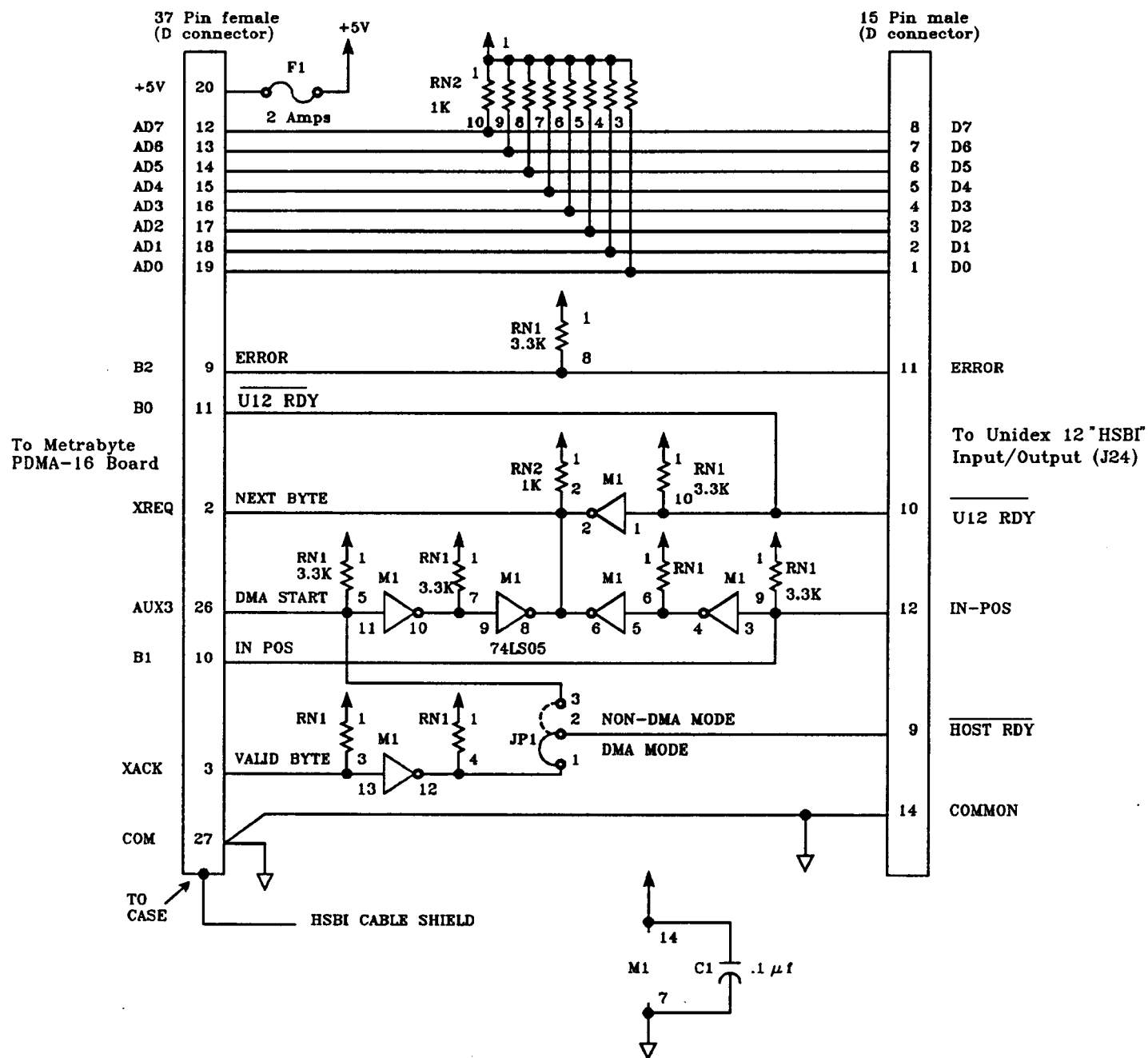
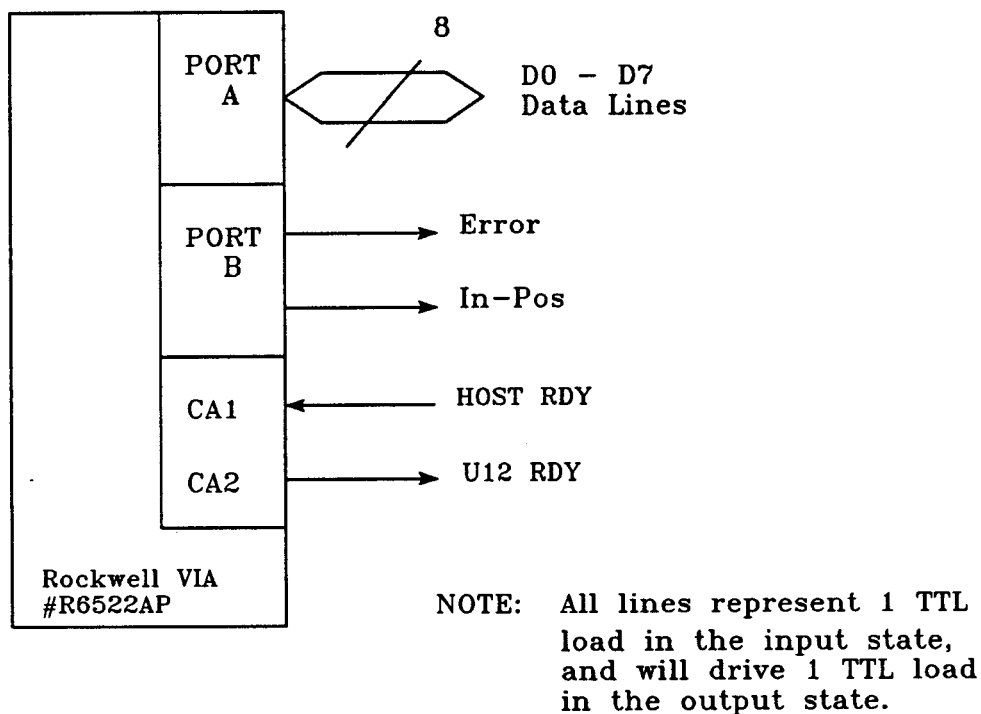


Figure 6-1: MBI (Metra Byte Interface) Adapter Schematic



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 12 while maintaining reliable operation.

Figure 6-2: HSB I/O Structure

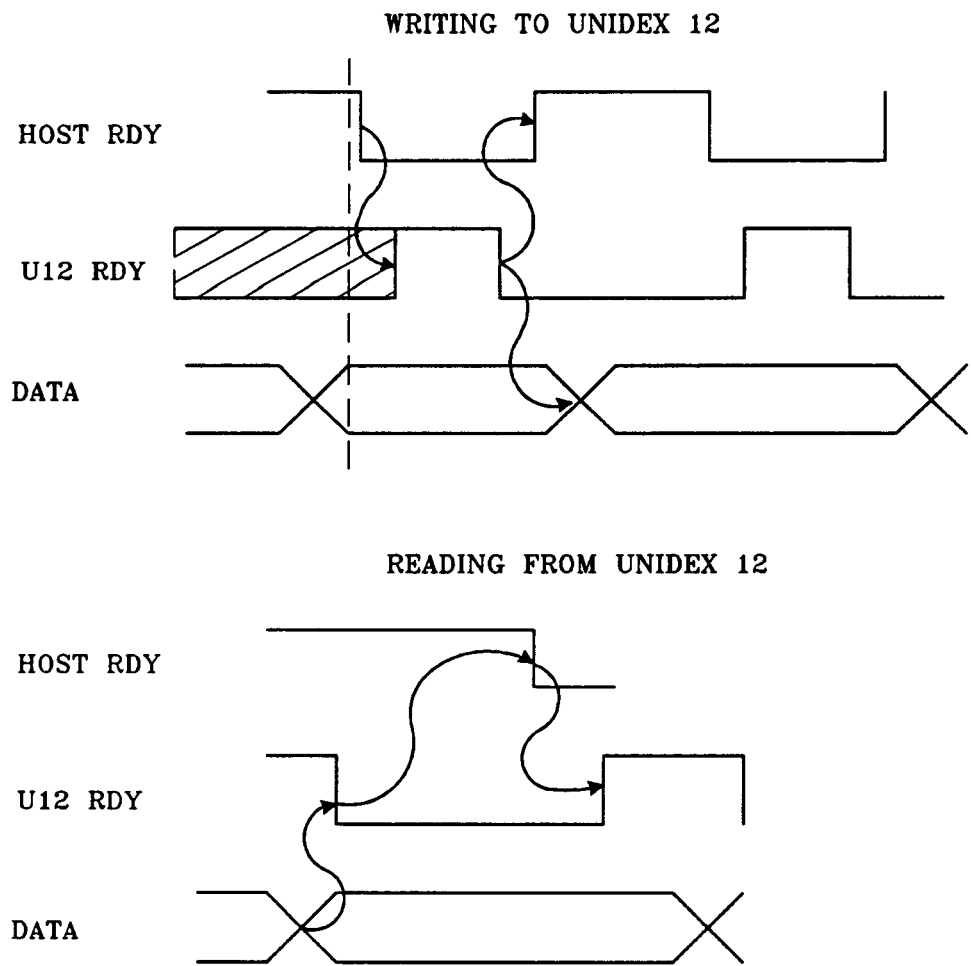


Figure 6-3 : HSB Timing Diagram

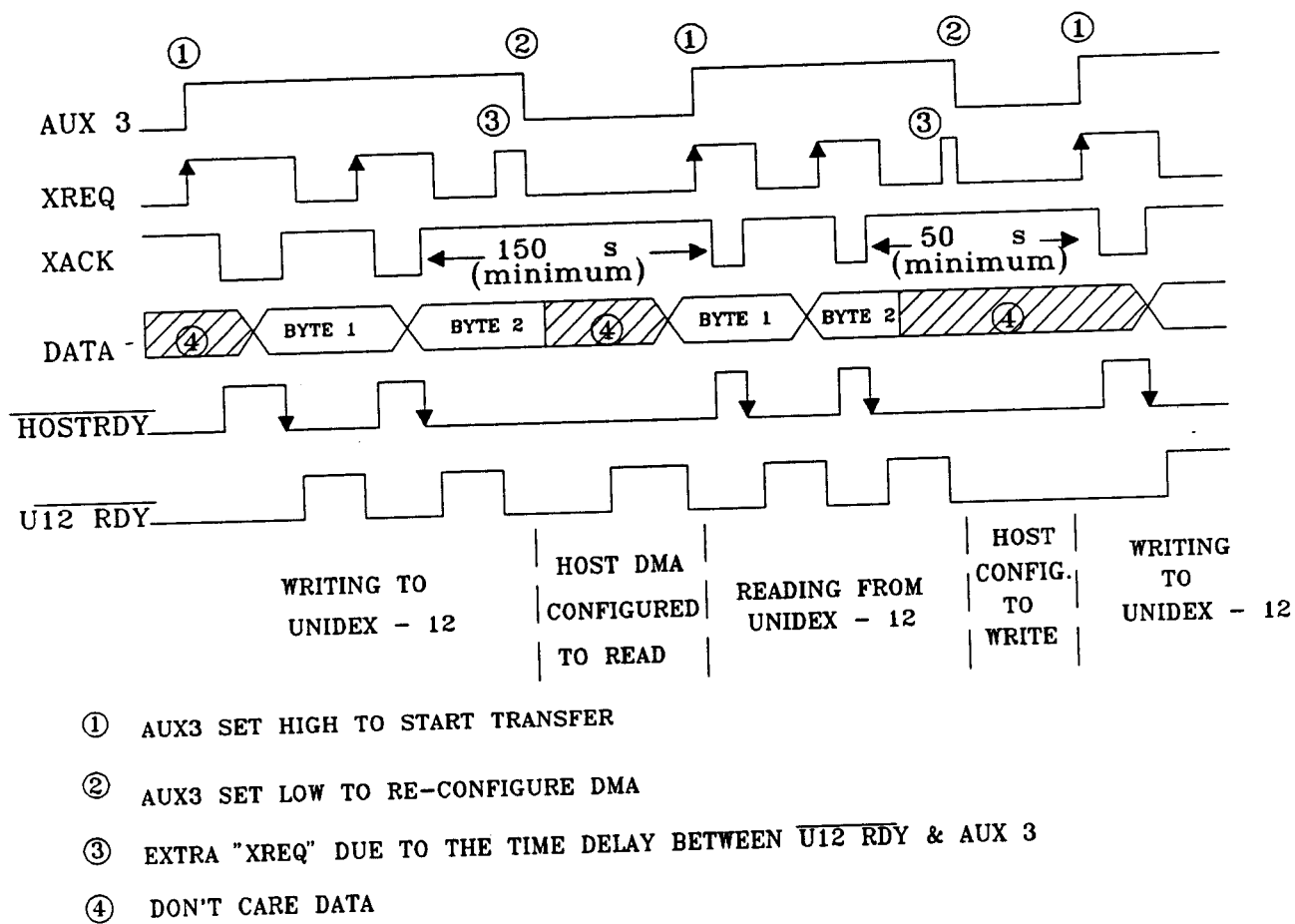


Figure 6-4: HSB-DMA Handshaking

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 6: 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.

6-1 Dwell

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

6-2 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)

6-3 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 corresponding 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)

6-4 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)

6-5 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 - running. The corresponding bits are set to 1.

Byte 4 = 54 hex (84 decimal)

6-6 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)

6-7 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 12.

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

6-8 Incremental Mode

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

6-9 Absolute Mode

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

6-10 Beeper On

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

6-11 Beeper Off

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

6-12 Corner - Rounding Mode

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

6-13 Non Corner - Rounding Mode

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

6-14 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.

(THIS PAGE LEFT INTENTIONALLY BLANK)

CHAPTER 7: 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 8: MBI (METRA-BYTE INTERFACE)

This section describes the MBI option required to interface a Metrabyte PDMA-16 digital I/O board to the Unidex 12 through the High Speed Binary Interface (see Figure 6-1).

WARNING: TO PREVENT THE POSSIBILITY OF ELECTRICAL SHOCK, PRIOR TO APPLYING POWER, MAKE CERTAIN THE MBI MOUNTING SCREWS ARE FASTENED SECURELY TO THE COMPUTER INTERFACE BOARD .

Unidex 12 is capable of data transfer rates of up to 80K bytes per second. High data transfer rates are most easily achieved by Direct Memory Access (DMA) data transfers. This technique bypasses the Host's processor and eliminates the associated overhead time by the use of a hardware controller. 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. The DMA controller must be initialized before receiving the transfer request (XREQ). This is not possible if the Host and the Unidex 12 are simultaneously powered up. Therefore, a provision exists 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 be initialized and the Unidex must be in position (IN POS) and ready (U12-RDY) for commands. The DMA start line (Refer to Figure 6-1) 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 repeated.

The MBI option also may be used in the non-DMA mode of operation (not requiring 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 12. After a valid byte has been placed on the data bus (Figure 6-3) this line is then negated. After receiving the data byte the U12 RDY line is asserted true. This mode is not normally used because of its low throughput.

The following 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 FILEEND
    INPUT #1,BYTE                ' Get byte
    POKE (COUNT),BYTE           ' Stick in memory

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

```

--MERGE METRABYTES DMA2.BAS PROGRAM HERE--
 DELETE LINES 210 AND 280 AND SET VARIABLES AS YOUR SYSTEM
 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
    GOTO DMATST              ' DONE?

```

(THIS PAGE LEFT INTENTIONALLY BLANK)

PART VI
PASSWORD

CHAPTER 1: PASSWORD INSTALLATION

SECTION 1-1: INTRODUCTION

This option allows the user to install a password protecting the Unidex 12 from unauthorized editing of system user parameters and user programs.

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

SECTION 1-2 INSTALLATION

Prior to Password installation a new Unidex 12, operates as a normal system, i.e., no password required. The password is installed in the Setup Mode. The procedure to install a password is as follows:

Select the Setup mode by pressing key #3 in the Third Main Menu screen. Unidex 12 will 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 number key may be pressed to enter the first digit of the password. After 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-number key (+/-, SELECT, INSERT, DELETE, BACK, ENTER) terminates the entry process. After typing in the desired password, press any non-numeral key to enter it. A 6 digit password is automatically entered upon entry of the 6th digit. The Unidex 12 will now require re-entry of this same password for confirmation.

The password entered the second time is compared to the first entry, if they are not identical, the Unidex 12 will require re-entry. 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 12 PASSWORD OPERATION

When a keyboard function such as RUN/SLEW is selected from the menu on the First Main screen, Unidex 12 displays the instruction:

ENTER PASSWORD
(up to 6 dig) :

Key in the password digits and press ENTER. If the correct password has been entered, the Unidex 12 will display the screen for RUN/SLEW data entry. If the password entered is not correct, the Unidex 12 will display:

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

Press key #1 for another opportunity to enter the password. Press key #2 to return to the First Main Menu screen (or the screen that initiated the Password Entry function).

SECTION 2-1 CHANGING THE PASSWORD

To change an existing password, select the Set Up mode from the Third Main Menu screen. Use the current password to enter the Set Up mode. Once in the Set Up mode, Unidex 12 displays the message:

CHANGE PASSWORD ?
1. Yes 1. No

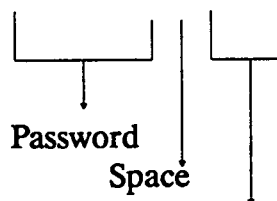
Press key #1 and repeat the procedures to enter a new password as described in Chapter 1.

SECTION 2-2 COMPUTER CONTROL OF UNIDEX 12

If equipped with the password option, it must be used in the Edit mode to download a program, 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*"



Program Number

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 : "Epppppp \$24*"

To clear all memory : "Epppppp \$00*"

SECTION 2-3 PASSWORD CORRUPTION

In the unlikely event of an alteration of the user memory, upon power-up, the Unidex 12, will display:

USER MEMORY CHECKSUM ERR
Press SELECT for a MENU

It is recommended that the user perform a LOAD DEFAULT operation to set the system parameters to the default values stored in the firmware. The LOAD DEFAULT function is available in the DIRECTORY mode from the First Main Menu screen.

The password may or may not be corrupted. To verify, invoke the password entry function by selecting the Setup mode. If the password is corrupted, the Unidex 12 will display:

**** PASSWORD CORRUPTED ****
Default PSWD :

The default password stored in system firmware is the number "012345". Enter this password correctly and Unidex 12 will display:

ENTER NEW PASSWORD
(up to 6 dig) :

The new password may be installed in accordance with the procedures of Chapter 1.

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

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

Press key #1 to return to the screen requesting the default password. Key #2 will return you to the Third Main Menu screen.

To reduce the possibility of undetected password corruption, Unidex 12 stores the logic equivalent of the password as well as the password itself. Whenever the password entry function is invoked, Unidex 12 verifies the password with its equivalent before requesting the user to enter it.

(THIS PAGE LEFT INTENTIONALLY BLANK)

SERVICE AND REPAIR

Customer repair of the equipment is limited to the items listed in the Troubleshooting Chapter in the *Unidex 12 Motion Controller Programming Manual*. The Translator and/or Control Board(s) may be removed and replaced if necessary, however, component level repair must not be attempted.

WARNING: DURING THE WARRANTY PERIOD, REPAIR OF ANY SUSPECTED DEFECTIVE ELECTRICAL COMPONENTS ON THE CONTROL OR TRANSLATOR BOARDS MUST NOT BE ATTEMPTED BY THE CUSTOMER. ANY ATTEMPT AT TRANSLATOR OR CONTROL BOARD COMPONENT REPLACEMENT OR REPAIR WILL VOID THE WARRANTY.

On-site service should be performed by an experienced electronic technician, preferably one 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 returning any equipment a "Return Authorization Number" must be obtained from Aerotech. (Be prepared to give the serial number of the equipment being returned.)
2. The equipment being returned must be encased in 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 TO THE EQUIPMENT DUE TO IMPROPER PACKAGING MAY VOID THE WARRANTY.

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

AEROTECH, INC. SERVICE CENTERS

World Headquarters

AEROTECH, INC.

101 Zeta Drive

Pittsburgh, Pa. 15238

Phone (412) 963-7470

FAX (412) 963-74

TWX (710) 795-3125

AEROTECH, LTD.

3 Jupiter House, Calleva Park

Aldermaston

Berkshire RG7 4QW England

Phone (07356) 7727459

FAX (07356) 5022

TLX 847228

AEROTECH GMBH

Neumeyerstrasse 90

8500 Nuernberg 10

West Germany

Phone (0911) 521031

TLX 622474

FAX (0911) 521235

INDEX

AXIS CALIBRATION

Checksum,3-5
Example,3-7
Influences,3-5
Introduction,3-1
Loading,3-4
Operation,3-3,3-4
Set Up,3-3

HIGH SPEED BINARY

Axis Position,5-11
Commands,5-7,5-23
Communication,5-11
Error,5-17
Joystick,5-14
Handshaking,5-12
Hardware,5-1
IN-POS,5-9
Introduction,5-1
Logic Controller,5-5
Metra-Byte Interface,5-31
Motion Interrupt,5-9
Programming,5-3
Remote Clock,5-15
Status,5-11,5-29
Tracking,5-13

IEEE-488

Accel./Decel.,1-41
Auto Mode,1-27
Axis Calibration,1-38
Block Mode,1-28
Block Numbering,1-30,1-31
Bus Controller,1-2
Beeper,1-53
Cable Restrictions,1-3
Cancel,1-31
Commands,1-14,1-27,1-39,1-53,1-62
Corner Rounding,1-42
Deleting a Program,1-30
DIO,1-55,1-57
Downloading,1-29
Dwell,1-43
Encoder Verification,1-44
End of Block,1-39
Error Codes,1-19,1-37
Feedrate,1-46

GOSUB,1-58
GOTO,1-58
Hardware,1-1
Hold,1-31
Home,1-45
Incremental/Absolute,1-40
Immediate Mode,1-32
Inputs,1-54
Interactive Control,1-1
Interface,1-1,ap2-1
Interrupt,1-45,1-51,1-53,1-58,1-61
Introduction,1-1
Joystick,1-36,1-37,1-46
Label,1-59
Linear Move,1-46
Limits,1-41,1-43,1-47,1-48,1-50
Listener,1-2,1-14
Loop End,1-60
Loop Start,1-61
Marker,1-55
Output,1-59
Out/Run,1-48
Out/Stop,1-48
Parallel/Serial Polling,1-3,1-10,1-15
Parameters,1-9
Position Registers,1-47
Print,1-34,1-49,1-50
Programming,1-13,1-54,1-55
Remote Mode,1-36
Remote Reset,1-28
Repeat Program,1-61
Reset,1-38,1-51
Service Request,1-15,1-33,1-37,1-51
Signal Lines,1-3,1-5
Status Byte,1-16,1-33
Stop Axis,1-52
Subroutine,1-61
Talker,1-2
Tracking,1-43,1-44
Trigger,1-36
Velocity Profile,1-41,1-52

JOYSTICK

Buttons,2-3,2-4
Control,2-3
Digitizing,2-11
Disable,2-7
Enable,2-3,2-9
Format,2-17
Hardware,2-1,2-11

Introduction,2-1
Position Registers,2-7
Remote,2-8
Service Request,2-4,2-6,2-9
Troubleshooting,2-21

PASSWORD

Changing,6-3
Computer Control,6-4
Corruption,6-4
Installation,6-1
Introduction,6-1
Operation,6-2

TWO DIGIT THUMBWHEEL

Introduction,4-1
Operation,4-1



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 warranted 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.

AEROTECH, Inc., 101 Zeta Drive, Pittsburgh, Pennsylvania 15238

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