
REFERENCE
DOCUMENT

**UNIDEX III
OPERATOR'S
MANUAL**

MAY, 1984

REV. 0-1A



AEROTECH, INC., 101 Zeta Drive, Pittsburgh, PA 15238

(412) 963-7470

TWX 710-795-3125

CONTENTS

CHAPTER		PAGE
1	<u>INTRODUCTION TO UNIDEX III</u>	1-1
1-1	UNIDEX III OPERATOR'S MANUAL	1-1
1-2	FEATURES OF UNIDEX III	1-2
2	<u>GETTING ACQUAINTED WITH UNIDEX III</u>	2-1
2-1	TYPES OF COMMANDS	2-1
2-2	TYPES OF PROGRAMMING	2-1
2-3	MODES OF OPERATION	2-1
A	EDIT MODE	2-2
B	AUTO MODE	2-2
C	SINGLE MODE	2-2
D	IMMEDIATE MODE	2-2
2-4	LOCAL PROGRAMMING MODE	2-3
A	EDIT MODE	2-3
	ENTERING NEW PROGRAM	2-3
	CHANGING OLD PROGRAM	2-4
B	AUTO MODE	2-8
C	SINGLE MODE	2-9

CHAPTER		PAGE	
	D	IMMEDIATE MODE	2-9
	E	MANUAL CONTROL	2-10
		STEP FUNCTION	2-10
		SLEW FUNCTION	2-10
		SLEW WITH JOYSTICK	2-11
2-5		REMOTE PROGRAMMING MODE	2-12
	A	EDIT MODE	2-12
	B	AUTO MODE	2-13
	C	SINGLE MODE	2-13
	D	IMMEDIATE MODE	2-14
3		<u>EQUIPMENT DESCRIPTION</u>	3-1
3-1		UNPACKING UNIDEX III	3-1
3-2		INSTALLATION	3-1
	A	A.C. POWER REQUIREMENTS	3-1
	B	SYSTEM POWER-UP	3-1
3-3		COMPONENTS	3-2
	A	REAR PANEL CONNECTORS	3-3
3-4		FRONT PANEL CONTROLS AND DISPLAY	3-4
	A	REMOTE SWITCH AND LOCAL LED	3-5
	B	RESET	3-5

CHAPTER		PAGE
	C IMMEDIATE	3-5
	D EDIT	3-5
	E AUTO	3-5
	F SINGLE	3-6
	G EXECUTE	3-6
	H PRINT	3-6
	I KEYBOARD	3-6
	J ALPHANUMERIC DISPLAY	3-6
	K X AND Y TRACKING DISPLAYS	3-6
	L X AND Y DISPLAYS CLEAR	3-6
3-5	JOYSTICK SPECIFICATIONS	3-7
4	<u>PROGRAMMING OF UNIDEX III</u>	4-1
4-1	POWERING-UP	4-1
4-2	LOCAL PROGRAMMING	4-1
	A EDIT MODE	4-1
	ENTERING A PROGRAM	4-2
	EDITING PROGRAM	4-3
	EDIT FUNCTIONS	4-3
	B AUTO MODE	4-7
	C SINGLE MODE	4-8
	D IMMEDIATE MODE	4-9

CHAPTER		PAGE
	E	MANUAL CONTROL
		4-10
		STEP FUNCTION
		4-10
		SLEW FUNCTION
		4-10
		SLEW WITH JOYSTICK
		4-11
4-3		REMOTE PROGRAMMING
		4-13
	A	EDIT MODE
		4-14
		ENTERING PROGRAM
		4-14
		STORING PROGRAM
		4-15
		DELETING PROGRAM
		4-16
	B	AUTO MODE
		4-16
	C	SINGLE MODE
		4-17
	D	IMMEDIATE MODE
		4-17
4-4		REMOTE INTERFACING
		4-18
	A	IEEE-488 INTERFACE BUS
		4-18
		DEVICE AS LISTENER
		4-19
		DEVICE AS TALKER
		4-19
		DEVICE AS CONTROLLER
		4-19
		SIGNAL LINES
		4-20
		CABLE RESTRICTIONS
		4-20
		PARALLEL AND SERIAL POLL
		4-20
	B	RS-232C/RS-422A INTERFACE
		4-21
		SIGNAL TRANSMISSION
		4-22
4-5		PROGRAMMING CAPABILITIES
		4-25
	A	UNCONDITIONAL JUMP
		4-25
	B	CONDITIONAL SKIP
		4-26
	C	ABORT OR INTERRUPT
		4-26
	D	SUBROUTINE CALL
		4-27
	E	REPEAT CYCLES
		4-27

CHAPTER		PAGE
	F SEND INFORMATION ON COMMAND	4-28
4-6	MEMORY CAPABILITIES	4-29
5	<u>COMMANDS FOR PROGRAMMING UNIDEX III</u>	5-1
5-1	RS-274-D DATA FORMAT	5-1
	A STRUCTURE OF RS-274-D	5-1
	B SEQUENCING OF COMMANDS	5-3
5-2	COMMAND GROUPS	5-4
	A X AND Y CODES	5-4
	B F CODES	5-7
	C M CODES	5-8
	D D CODES	5-10
	E N CODES	5-10
	F EOB CODE	5-12
	G G CODES	5-13
5-3	INTERFACING UNIDEX III	5-23
	A LOCAL INTERFACE	5-24
	B IEEE-488 INTERFACE	5-24
	C RS-232C INTERFACE	5-24
	D RS-422A INTERFACE	5-25

CHAPTER		PAGE
5-4	LOCAL SAMPLE PROGRAMS	5-26
A	EDIT MODE	5-26
	EDIT	5-26
	STEP AND BACKSTEP	5-26
	SEARCH	5-26
	SLEW	5-27
	CLEAR ENTRY	5-27
	CLEAR COMMAND	5-27
	EDIT SAMPLE PROGRAMS	5-27
B	AUTO MODE	5-29
	AUTO SAMPLE PROGRAM	5-29
C	SINGLE MODE	5-30
	SINGLE SAMPLE PROGRAM	5-30
D	IMMEDIATE MODE	5-30
	IMMEDIATE SAMPLE PROGRAM	5-31
	MANUAL MODE	5-32
E	PRINT-OUTS OF PROGRAMS	5-34
F	CLEARING PROGRAMS	5-35
5-5	REMOTE SAMPLE PROGRAMS	5-36
A	ADDRESSING UNIDEX III	5-36
	IEEE-488 INTERFACE	5-36
	RS-232C/422A	5-36
B	EDIT MODE	5-37
	IEEE-488 ENTRY	5-38
	RS-232/422 ENTRY	5-38
C	AUTO MODE	5-39
	IEEE-488 ENTRY	5-39
	RS-232/422 ENTRY	5-39
D	SINGLE MODE	5-40
	IEEE-488 ENTRY	5-40
	RS-232/422 ENTRY	5-40
E	IMMEDIATE MODE	5-41
	IEEE-488 ENTRY	5-41

CHAPTER		PAGE
	RS-232/422 ENTRY	5-41
F	PRINT	5-42
	IEEE-488 BUS	5-42
	RS-232/422 INTERFACE	5-43
G	ERASE	5-43
	IEEE-488 BUS	5-43
	RS-232/422 INTERFACE	5-43
H	RS-232/422 CODES	5-44
	CLEAR	5-44
	TRIGGER	5-44
	LOCAL	5-45
	CANCEL HOLD	5-45
	PROGRAMMABLE SRQ	5-45
	CANCEL PROGRAMMED SRQ	5-45
	SERIES RTS	5-45
	PARALLEL RTS	5-46
	QUERY	5-46
5-6	RS-274-D COMMAND SUMMARY	5-47
	X AND Y CODES	5-47
	N CODES	5-47
	M CODES	5-48
	F CODES	5-49
	D CODES	5-49
	G CODES	5-49
6	<u>SERVICE AND REPAIR</u>	6-1
APPENDIX 1	TABLES	
ADDENDUM 1	STATUS BYTES	
INDEX		
SALES		

ILLUSTRATIONS

FIGURES		FOLLOWING PAGE
1-1.1	FRONT PANEL OF UNIDEX III	1-1
1-1.2	REAR PANEL OF UNIDEX III	1-2
3-3.1	TYPICAL DC DRIVE CHASSIS	3-2
3-3.2	TYPICAL STEPPER DRIVE CHASSIS	3-2
3-3.3	PIN DEFINITIONS OF M-OUTPUTS, C-INPUTS AND JOYSTICK	3-3
3-3.4	INTERFACE BOARD (REAR PANEL)	3-3
3-4.1	FRONT PANEL CONTROLS AND DISPLAY	3-4
4-1.1	UNIDEX III LOCAL AND REMOTE MODES MODES OF COMMUNICATION	4-1
4-3.1	SELECTOR SWITCHES LOCATED ON DISTRIBUTOR CARD (BACK OF CHASSIS)	4-13
4-4.1	IEEE-488 ADDRESS SELECT SWITCHES LOCATED ON DISTRIBUTOR CARD (BACK OF CHASSIS)	4-21
4-4.2	UNIDEX III TO HOST COMMUNICATIONS USING RS-232 INTERFACE	4-22
4-4.3	ASSIGNMENTS FOR 37-POSITION RS-422 CONNECTOR USED IN UNIDEX III	4-22

TABLES

TABLES		FOLLOWING PAGE
4-3.1	ASCII CHARACTER SET	4-13
4-4.1	IEEE-488 AND RS-232/422 DAISY CHAIN	4-20
4-4.2	IEEE-488 STANDARD INTERFACE BUS SIGNAL LINE	4-20
4-4.3	IEEE-488 CABLE MANUFACTURERS	4-21

(ALL TABLES ARE ALSO INCLUDED IN APPENDIX 1)

CHAPTER 1

INTRODUCTION

Unidex III is a complete, extremely fast numerical motion controller, designed to govern two axis of DC servo or stepping motor drives. Unidex III's powerful set of motion control commands, battery-backed memory, and versatile interface capability gives it the competence to handle virtually all point-to-point motion control applications.

Unidex III's motion control capabilities include independent axis feedrates, joystick operation, programmable inputs, conditional skips, and interrupts during moves. Its internal position registers allow up to $\pm 2,000,000,000$ increments of system resolution for each axis. As a stand-alone motion controller, programs or individual motions can be entered, edited, and executed from the front panel controls. Its expandable memory (4 kilobytes standard, 32 kilobytes maximum) stores up to 99 programs which can be randomly accessed either locally on the Unidex III keyboard or remotely (by a host).

A host (computer or terminal) can be used to develop and edit a program, download it into the Unidex III and then be disconnected. Unidex III can execute a program from memory and communicate via interrupts with the host, or the host can download individual motions into the Unidex III. If used with a host, the communication bus can be any of the following three:

1. IEEE-488
2. RS-232C
3. RS-422A

Connections for all three interfaces, the 8 buffered outputs, 4 buffered inputs, and the joystick are found on the back panel of the Unidex III chassis. The RS-232C and RS-422A interfaces have multi-axis daisy-chain capabilities. These connections are also included on the back panel. A complete listing of all Unidex III features is included at the end of this section.

SECTION 1-1 UNIDEX III OPERATOR'S MANUAL

It is recommended that this manual be read before you attempt the installation or programming of Unidex III.

This chapter (Introduction) and chapter 2 (Getting Acquainted With Unidex III) will familiarize you with Unidex III and some basic programming techniques.

Chapter 3, Equipment Description, gives unpacking instructions and a description of the Unidex III components.

Chapter 4, Programming of Unidex III, is a more detailed description of local and remote programming.

Commands For Programming Unidex III, chapter 5, is a description of the language required for programming Unidex III, along with code summaries and sample programs.

Chapter 6, Service and Repair, is intended for use in the event that a problem should arise with your Unidex III. Included for your convenience, at the end of this manual, are an index and a list of Aerotech offices and sales representatives, as well as appendices, addendums and tables.

SECTION 1-2 FEATURES OF UNIDEX III

The features and capabilities of Unidex III are as follows:

- 2-axis control . . . $\pm 2,000,000,000$ steps maximum
- Independent feedrates for both axes . . . up to 150 KHz (actual value programmed will depend on the capability of the drives)
- Remote interface - RS232C, RS422A and IEEE-488
- Multi-axis daisy chaining capability for the remote interfaces

UNIDEX III

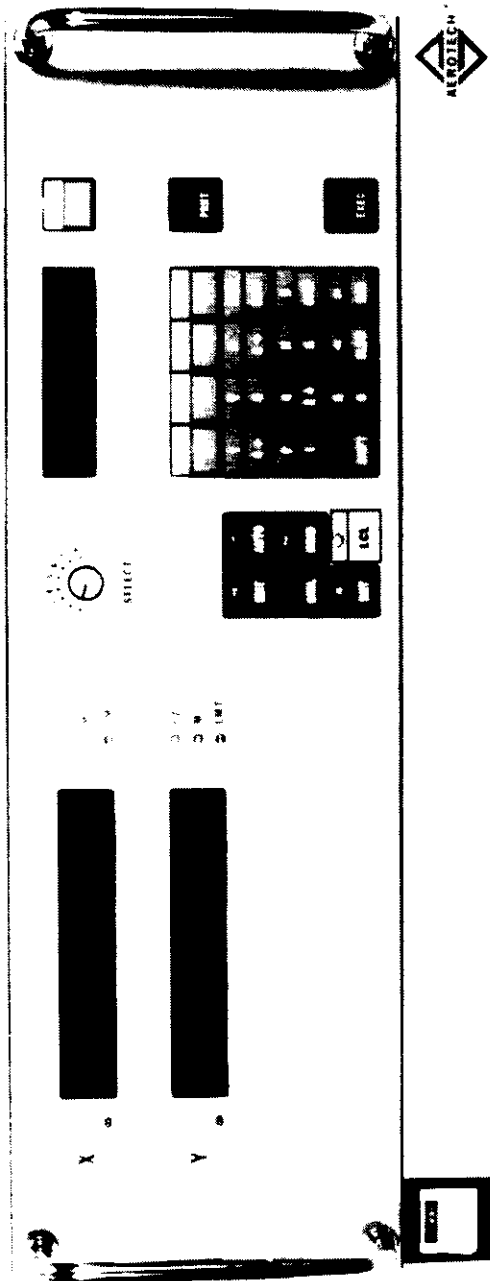


FIGURE 1-1.1 FRONT PANEL

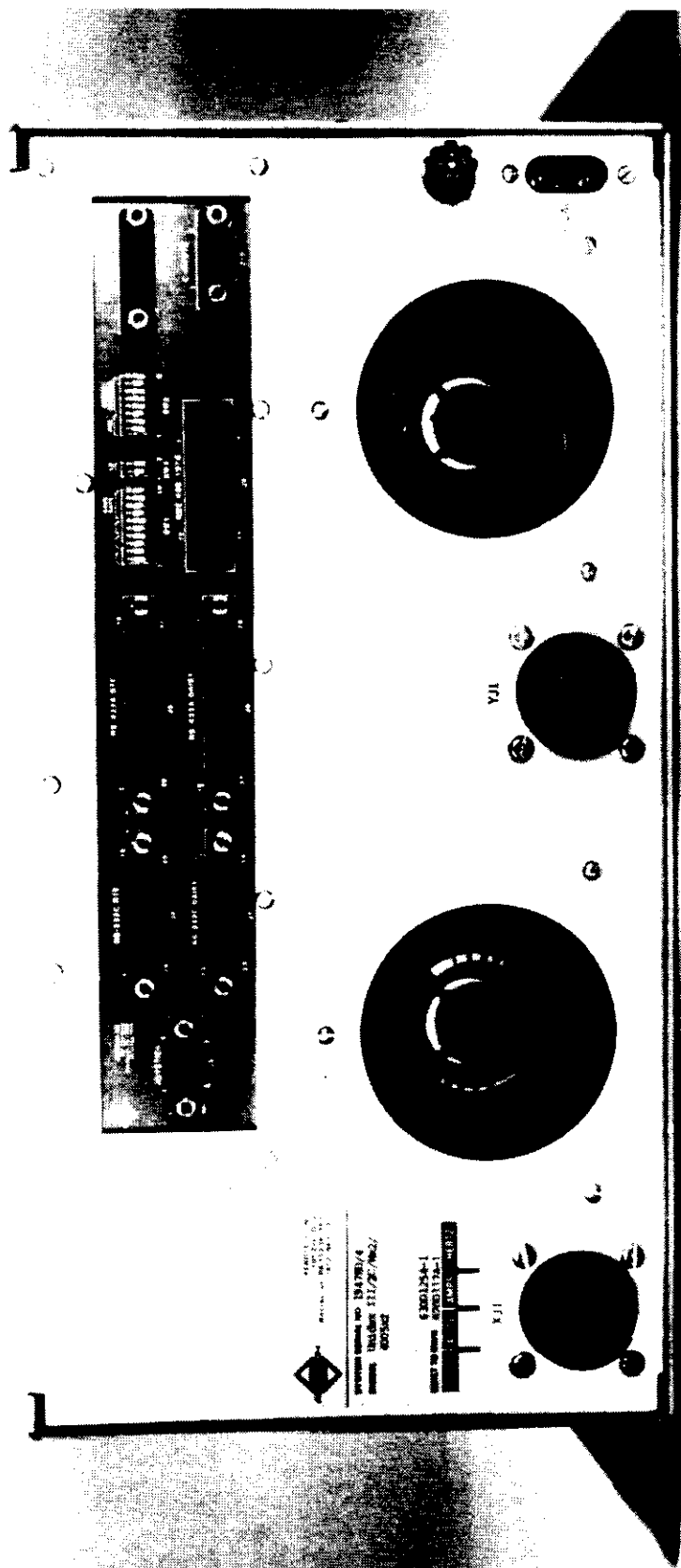


FIGURE 1-1.2 REAR PANEL (INTERFACE BOARD)

- 8 buffered outputs with strobe
- 4 buffered inputs
- Joystick operation
- Alphanumeric display - 8 characters long
- X and Y tracking displays
- Stored program capability (multiple programs randomly accessible)
- 4K storage - battery backup (up to 32K optional)
- Downloadable from remote (host computer)
- Incremental Mode - Absolute Mode
- Corner Rounding - Non-corner Rounding
- Programmable functions:
 - Repeat cycles
 - Dwell
 - Unconditional jumps
 - Conditional skips (hard and soft conditions)
 - Abort on interrupt
 - Subroutines (stack and jump with return)
 - Store (remember) positions on command or interrupt and recall
 - Send X, Y and M status to remote controller on command
 - Programs printed or sent to host from Unidex III
 - Remote programming
 - Trigger mode - allows multi-axis synchronization
 - Single front panel may control up to 8 axis pairs (optional)

- Edit functions:
 - Step
 - Backstep
 - Search
 - Clear entry
 - Clear command
 - Start

CHAPTER 2

GETTING ACQUAINTED WITH UNIDEX III

The purpose of this chapter is to allow you to begin programming Unidex III immediately. The details of operation and programming will be covered later in the manual. Our intention right now, however, is to give you a general view of how to operate your Unidex III so that you can get started. (A motion command summary can be found in chapter 5, section 6.)

SECTION 2-1 TYPES OF COMMANDS

There are two types of commands involved in programming the Unidex III. The first is mode commands. Mode commands are those which allow you to enter your program into Unidex III's memory and execute it in a specific manner.

The second type is the motion commands. Motion commands are those which control the action of the axes and comprise your actual program.

SECTION 2-2 TYPES OF PROGRAMMING

There are two ways to program your Unidex III. You may enter your commands by the front panel keyboard. This is called local programming.

The second method of entering a program into Unidex III is by a host (computer or terminal). This is called remote programming.

SECTION 2-3 MODES OF OPERATION

The various modes in which Unidex III can operate are accessible to you by either local or remote programming. These modes are:

- Edit mode
- Auto mode
- Single mode
- Immediate mode

A. EDIT_MODE

To enter or modify (edit) a motion program, you must be in the Edit mode. The editing of your program is made possible by the following functions:

1. EDIT - Allows you to enter a new program or call up an existing one.
2. STEP/BACKSTEP - Allows you to step through a program, in either a forward or a backward direction.
3. SEARCH - Allows you to locate a certain command or string of characters within your program.
4. CLEAR ENTRY - Allows you to erase an entry within your program.
5. CLEAR COMMAND - Allows you to erase a command within your program.

B. AUTO_MODE

When in the Auto mode, you may run a program from beginning to end without any further input.

C. SINGLE_MODE

When in the Single mode, you may execute a program one block at a time. Just press EXECUTE to run each block of commands. Unidex III will display the message "PGM END" (program end) when the program is finished.

D. IMMEDIATE_MODE

The Immediate mode allows you to enter a command block and run it immediately without storing it in the user memory.

SECTION 2-4 LOCAL PROGRAMMING MODE

Upon powering up, the front panel keyboard display will show the words System Ready (SYS RDY) and both the local and remote LED's will be lit. This is the "local with remote enabled" state. If at this point any key on the front panel keyboard is pressed, other than REMOTE, the remote LED will be extinguished and Unidex III will be in the local programming mode.

CAUTION: MAKE SURE DRIVES ARE CLEAR BEFORE BEGINNING A PROGRAM.

NOTE: When local sample programs are given, blocks will be used to indicate front panel keyboard entries. Unidex III's responses will be indicated by underscoring.

A. EDIT MODE

1. Entering A New Program

To enter a new program, program 25 for instance, the following sequence of mode commands are entered.

<div style="border: 1px solid black; padding: 2px; display: inline-block;">EDIT</div>	PGM__<1> (program 1 is default program)
<div style="border: 1px solid black; padding: 2px; display: inline-block;">2</div>	EDIT__2
<div style="border: 1px solid black; padding: 2px; display: inline-block;">5</div>	EDIT__25
<div style="border: 1px solid black; padding: 2px; display: inline-block;">EXEC</div>	_____>>

Program 1, the default program, appears upon power up. Program 25 is requested, EXECUTE is pressed, and Unidex III is now ready to receive its new commands.

Your motion commands may now be entered. For example:

SHFT	G	----->>G	
	7	----->>G7	(axes go home)
SHFT	X	---->>G7X	
	1	--->>G7X1	
	0	_->>G7X10	
	0	>>G7X100	(X axis, 100 steps)
SHFT	F	>G7X100F	
	1	G7X100F1	
	0	7X100F10	(feedrate, 10 steps/second)
	.	.	
	.	.	(program continues)
	.	.	
SHFT	M	Y100F10M	
	2	100F10M2	(end of program)

2. Changing An Existing Program

The edit functions help speed up the editing process. How to use these functions when altering a program is explained in the following paragraphs.

a. STEP and BACKSTEP

When you need to examine your program one command at a time, you will

need the Step/Backstep function. This allows you to step through the program either in a forward or backward direction.

Since the program "wraps around", you may start at the beginning of a program and by backstepping, come to the last command of the same program. Example (using the previous sample program):

<input type="button" value="EDIT"/>	EDIT__25
<input type="button" value="EXEC"/>	____>>GZ (first command)
<input type="button" value="STEP"/>	>>GZX100
<input type="button" value="STEP"/>	ZX100F10

Each time STEP is pressed, the next command is displayed.

Each time BACKSTEP is pressed, the previous command is displayed. Example:

<input type="button" value="BSTP"/>	>>GZX100
<input type="button" value="BSTP"/>	____>>GZ
<input type="button" value="BSTP"/>	____>> (last command)
<input type="button" value="BSTP"/>	100F10M2

b. SEARCH

When you must hunt for a particular command, you will need the Search function. Example:

<input type="button" value="EDIT"/>	EDIT__25
-------------------------------------	----------

```

      EXEC      ____>>GZ
SHFT SRCH      _____ (display goes
SHFT Y200      ____Y200    blank)
      EXEC      _LOOKING_   (searches)
                        OF10Y200 (finds it)

```

If Y200 is found, it will be shown on the display. If not, "WHAT?" will be seen on the display instead. Example:

```

SHFT SRCH      _____
SHFT Y200      ____Y200
      EXEC      _WHAT?_    (not found)

```

You realize you should have searched for Y250, not Y200, so:

```

SHFT SRCH      _____
SHFT Y250      ____Y250
      EXEC      _LOOKING_
                        OF10Y250

```

c. SLEW

During an Edit session, pressing the SLEW key brings you back to the first command from wherever you are in the program. Example:

```

SLEW      ____>>GZ

```

d. CLEAR_ENTRY_and_CLEAR_COMMAND

These two edit functions are needed when erasing an entry or a command. A new entry or command may be substituted by simply typing it in after the old one has been erased. Example (using previous sample program):

SHFT	SRCH	-----
SHFT	Y250	____Y250
	EXEC	_LOOKING
		0F10Y250
	CE	00F10Y25
	CE	100F10Y2
	0	00F10Y20
	0	0F10Y200 (Y250 is changed to Y200)

Example (using Clear Command instead of Clear Entry):

	EXEC	_LOOKING
		0F10Y250
SHFT	CC	1X100F10
SHFT	Y	X100F10Y
	2	100F10Y2
	0	00F10Y20
	0	0F10Y200

To complete the editing session, press any of the mode keys, ie., EDIT, AUTO, IMMEDIATE, SINGLE or REMOTE. Example (using the previous sample program):

RUN___25

Unidex III is now in the Auto mode instead of the Edit mode. The same program number is still displayed, however. To enter another program, simply type in the desired number. Example:

RUN___25
 RUN___50
 RUN___02 (must type in 0 before single digit numbers)
 _RUNNING
 _ _READY_ (program 2 complete)

B. AUTO_MODE

To enter the Auto mode, where the entire program runs at once, press AUTO. Enter the desired program number and then press EXECUTE. Example:

POWER UP SYS__RDY
 PGM__<1> (default program)
 RUN___0
 RUN___02
 _RUNNING

__READY__ (ready to run program again)

To run program 2 again, just press the EXECUTE key. In order to run another program instead, press the AUTO key. You will then be able to type in the new number.

C. SINGLE_MODE

Pressing SINGLE enters you into the Single mode, where the program will be executed one block at a time. Example:

<u>SINGLE</u>	<u>WALK__02</u> (still in program 2 from previous example)
<u>3</u>	<u>WALK__23</u>
<u>0</u>	<u>WALK__30</u>
<u>EXEC</u>	<u>__RUNNING</u>
	<u>__READY__</u> (ready to run next block. If at end of program, will display "PGM END")

D. IMMEDIATE_MODE

In order to enter the Immediate mode, where short programs are entered and executed immediately, simply press IMMEDIATE. Example:

	<u>IMMD</u>	<u>ENTER__>></u>
<u>SHFT</u>	<u>G</u>	<u>----->>G</u>
	<u>7</u>	<u>----->>G7</u>
<u>SHFT</u>	<u>X</u>	<u>----->>G7X</u>

EXEC_RUNNING__READY__(program
complete)

Every time EXECUTE is pressed, these commands will run again. To enter the next string of commands, you don't have to press IMMEDIATE again. You simply type in the commands and press EXECUTE. The program will run again. It is not retained in memory, however. Once any of the mode keys is pressed, excluding IMMEDIATE, the program is lost.

E. MANUAL CONTROL

The manual control operations are Step and Slew. They both operate in the Local Immediate mode.

1. STEP FUNCTION

The Step function is useful for precise positioning of the drives. After depressing the STEP key, the drives will move one step each time one of the arrow keys is pressed. Example:

IMMDSTEP(use arrow keys to
position drives)

2. SLEW FUNCTION

The Slew function is useful for rapid manual positioning of the drives. After entering a feedrate and depressing the SLEW key, pressing and holding one of the arrow keys will cause one of the drives to move in a given direction until the key is released.

The feedrate can be entered as either frequency (Fnnnn) or pulse period (F=nnnn). Example:

IMMD Fnnnn or F=nnnn SLEW

If no feedrate is entered, the default feedrate is 250 steps per second after power up. Any feedrate entered is stored and is valid until changed.

The X and Y feedrates can be programmed individually. Example:

IMMD X Fnn Y Fnn SLEW

3. SLEW FUNCTION USING JOYSTICK

Unidex III is always ready to accept joystick input when idle in the Slew mode. In the Slew mode, the joystick rather than the arrow keys may be used to control direction. The feedrate controlling the speed of travel is specified to be a fraction of the joystick frequency. This is done by entering command F-nnnnn (2 - 65535 range). This lets Unidex III know that the joystick will be used for positioning, and that the feedrate will be the joystick frequency divided by a given factor (keyboard slew overrides joystick, however). Example:

IMMD F-20 SLEW

If no joystick feedfactor is entered, the default feedfactor is 2 (F-2).

To exit the Step or Slew mode, press Clear Entry or Clear Command.

These manual functions are also usable whenever the system is ready after power up or after running a program or a block of program.

SECTION 2-5 REMOTE_PROGRAMMING_MODE

Upon powering up, the front panel keyboard will show that both local and remote LED's are lit. This is the "local with remote enabled" state. If the next command source is the host, the local LED will be extinguished and Unidex III will go into remote programming mode.

Before sending any commands, mode or motion, Unidex III must be addressed to listen. How it is addressed depends on what type of host is being used and is explained later in section 5-5A.

CAUTION: MAKE SURE DRIVES ARE CLEAR BEFORE BEGINNING A PROGRAM.

NOTE: All remote characters received from the host by Unidex III will be denoted by brackets. Long strings of commands will be enclosed in quotation marks.

A. EDIT_MODE

1. Entering_a_New_Program

To enter a new program, you must send the ASCII character for Edit, which is E, along with the appropriate number. Example:

```
"E99"                                (edit program 99)

[CR][LF]    (carriage return/line feed -
              EXECUTE)
"G90 X100 F10 . . .M30"    (motion commands)

[CR][LF]                                (execute)
```

Edit functions Step, Backstep, Search, Clear Entry and Clear Command are not available in remote. You cannot edit an existing program when in remote. You must instead download the program into Unidex III and edit it from the front panel.

If a program already exists in memory and you send its number over the lines, it will be erased. The new program will replace the old one.

B. AUTO_MODE

In order to run a program in the Auto mode, you simply send the ASCII character "A" along with the program number and execute by sending [CR][LF]. Example:

"A 30 CR LF"

In this example, program 30 runs from beginning to end. When this program is complete, Unidex III requests service and waits for a serial poll. To execute the same program again, program the host to do a serial poll of Unidex III and send [CR] [LF].

C. SINGLE_MODE

To enter the Single mode while in remote, send an "S" followed by the program number. Example:

"S 03 CR LF"

Program 3 runs to first end-of-block. At this point, Unidex III requests service and waits for a serial poll. To execute the next block, do a serial poll and send [CR][LF].

D. IMMEDIATE_MODE

To enter the Immediate mode, send the ASCII character "I". To begin executing a program, follow the character "I" by a command string, followed by a [CR][LF]. Example:

"I G7 X300 F1 Y250 F1 CR LF"

When Unidex III completes the command block, it will send a request for service and wait for a serial poll. To execute another command block, do a serial poll and then send [CR][LF].

The host you use may be a terminal or a computer. In fact, it can be any device which is capable of sending and receiving ASCII code. How you address Unidex III to listen to your host depends on the type of programming required by the host itself.

CHAPTER 3

EQUIPMENT DESCRIPTION

SECTION 3-1 UNPACKING UNIDEX III

Remove the system from its shipping container and, referring to the sales order, verify that all items are present. Save the packing material for storing and reshipping the system.

IMPORTANT: If the shipping container is damaged upon receipt, request that the carrier's agent be present while the system is being unpacked and inspected.

The system should be inspected upon receipt for broken, damaged or loosened parts. Retighten any loosened connectors.

SECTION 3-2 INSTALLATION

Install all motor connectors and cables before applying power to the system. If the system includes an Aerotech stage, refer to the stage manual and remove shipping clips. If the system does not include a stage, the drives will have to be adjusted for proper response to the load inertia and friction. Refer to these drive manuals for adjustment procedures.

A. A.C. POWER REQUIREMENTS

The standard system requires 115/230 VAC $\pm 10\%$ at 50 - 60 Hz. Unidex III itself requires 1.8 amps at 115 VAC, however, in a typical system. The chassis will be fused at 5 amps.

B. SYSTEM POWER-UP

On power-up, Unidex III will reset both drives, initialize itself, and perform a self-test.

The message "SYS RDY" (system ready) or an error message will be displayed, based on the results of the self-test.

If Unidex III passes the self-test, the SYS RDY message will appear and the local and remote LEDs will be lit (this indicates the local with remote enabled mode). Unidex III is now ready to accept commands in either the local or remote mode.

At this point, the system will be set to the following default conditions:

- Local mode with remote enabled
- Incremental mode (G91)
- Non-corner-rounding mode (G24)

SECTION 3-3 COMPONENTS

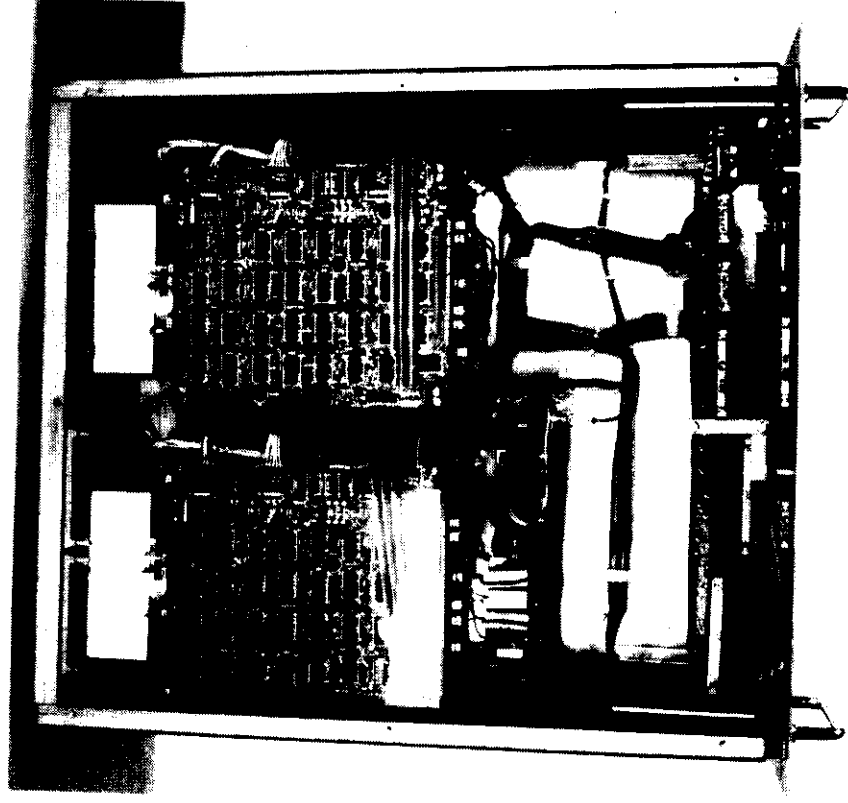
The five distinct parts of the Unidex III are:

1. Power supply
2. MPU board
3. Front panel board
4. Rear panel board
5. MPX board (optional)

In most applications, these components are shipped fully connected and packaged in a 19" rack mount chassis which includes one or two drives (stepping motor translators or DC servo systems). You will find all inputs and outputs on the rear of the chassis. Refer to figures 3-3.1 and 3-3.2 for internal photographs, showing two of Unidex III system variations.

The power supply generates all of the DC power required by the MPU board and the panel board, as well as a typical complement of serial load boards, ramper boards and encoders.

(CABLE REMOVED)



(WITH CABLE)

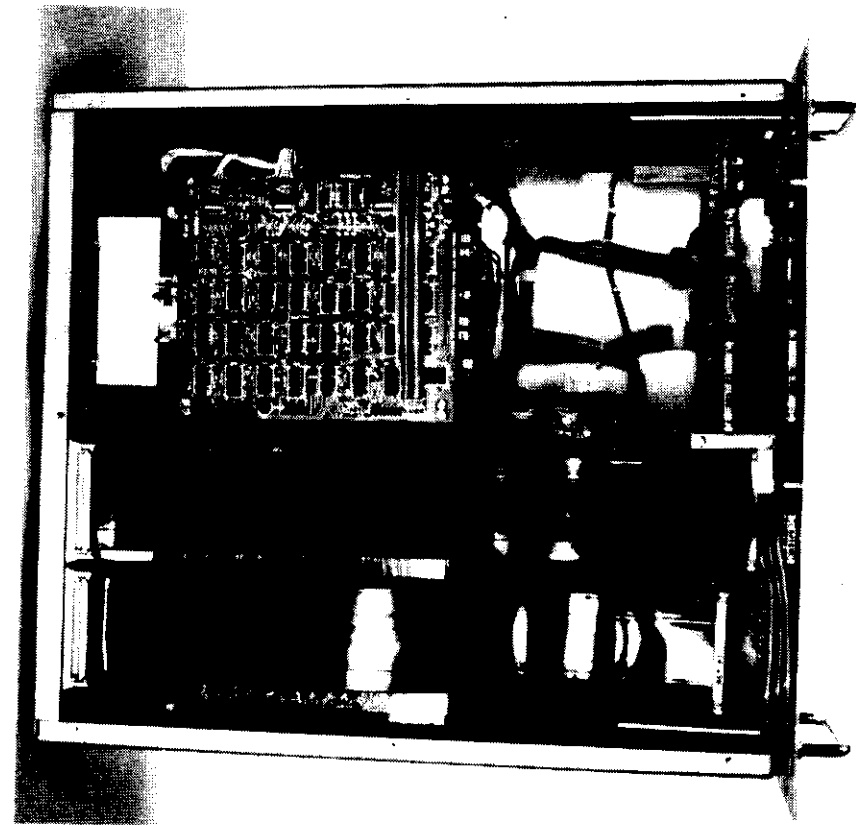
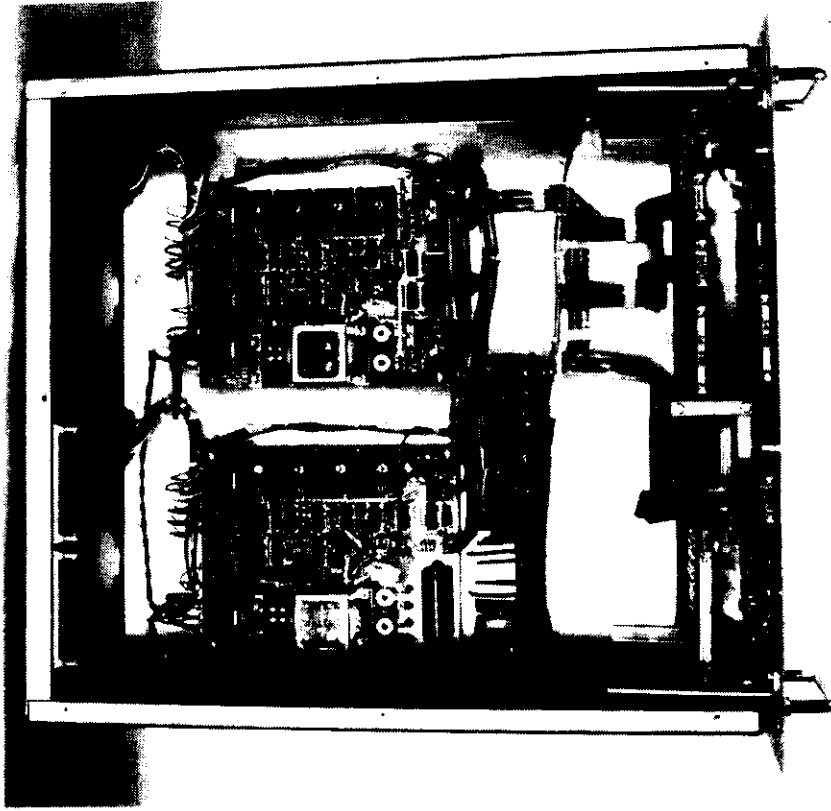


FIGURE 3-3.1 TYPICAL DC DRIVE CHASSIS

(CABLE REMOVED)



(WITH CABLE)

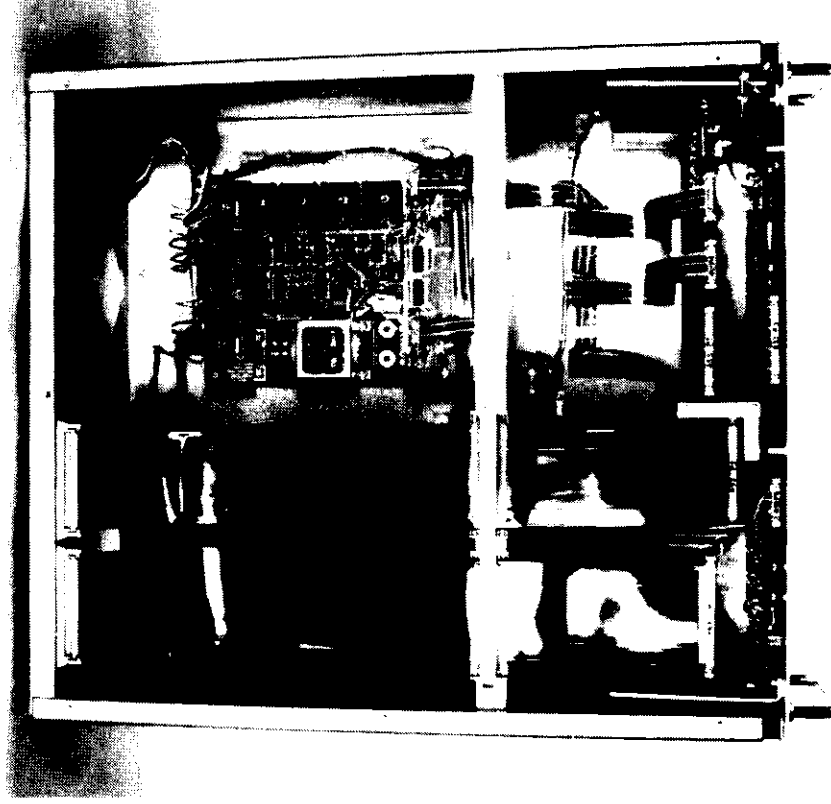


FIGURE 3-3.2 TYPICAL STEPPER DRIVE CHASSIS

The MPU board executes the program for axis control and stores up to 32K of user programs in battery-backed-up memory. In addition, the MPU board has the following features:

- Indexing logic and buffered indexer outputs
- Eight buffered "M" outputs with strobe: TTL compatible high current drivers (figure 3-3.3)
- Four buffered "C" (control) inputs: one CMOS load each (figure 3-3.3)
- Remote interfaces: IEEE-488, RS-232C, RS-422A
- Joystick interface (figure 3-3.3)

The MPU board has the resistor, R95, which controls the duration of the strobe. It is located on the lower right-hand side of the MPU board on stand-offs for easy removal. To substitute another resistor value for the present one, unsolder R95 and remove it. Solder the new resistor in place. Your Unidex III is shipped from the factory with an R95 value of 1K. This value gives a strobe pulse duration of 0.1 millisecond. R95 can range from 10 ohms to 10 Mohms, giving a pulse duration of from 1 microsecond to 1 second.

The front panel board has a display control and keyboard interface, as well as a keyboard, mode select switches, reset switch, alphanumeric display and two numeric tracking displays. In the multi-axis option, there is also an axis select switch. All front panel board controls are accessible from the front panel.

A. REAR PANEL (INTERFACE BOARD) CONNECTORS

The back of the chassis of the Unidex III (figure 3-3.4) has connectors for the following:

- IEEE-488 communication bus connector
- RS-232 communication bus connector

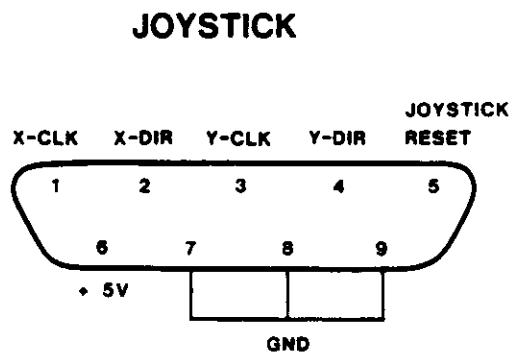
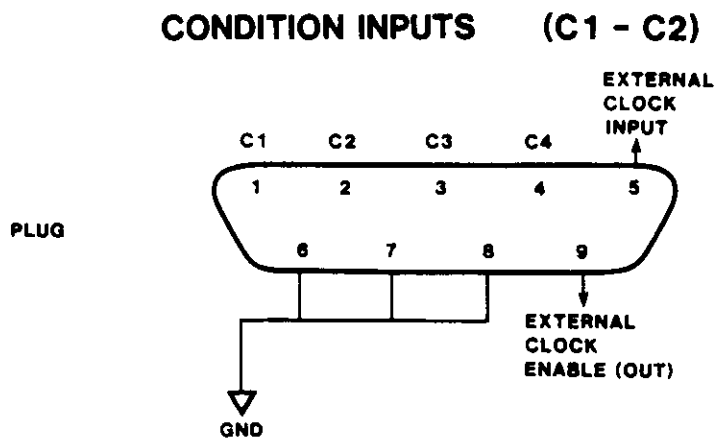
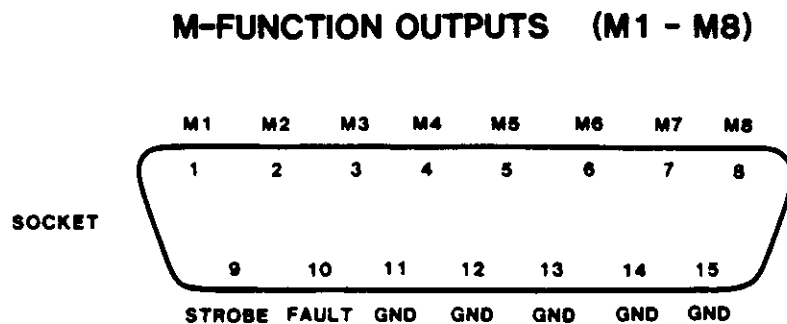


FIGURE 3-3.3 PIN DEFINITIONS OF M-OUTPUTS, C-INPUTS AND JOYSTICK

-
- RS-422 communication bus connector
 - Joystick connector
 - Input/output (M-outputs and C-inputs)
 - Motors/encoders (when ordered complete with drives)
 - Switches

Connecting any of the above keyed connectors is a simple plug-in procedure.

NOTE: The serial number, which is located on the rear panel, is the reference number needed when contacting Aerotech concerning the Unidex III. Please have it handy when calling our Customer Service Department.

SECTION 3-4 FRONT PANEL CONTROLS AND DISPLAY

The front panel (figure 3-4.1) has a 16 key alphanumeric keyboard and five mode select keys. The mode select keys are provided with LED indicators and comprise the following:

1. EDIT
2. AUTO
3. SINGLE
4. IMMEDIATE
5. REMOTE

The SHIFT key toggles and is also LED indicated. The remaining front panel LEDs are:

1. X axis (CZ, M, LMT)
2. Y axis (CZ, M, LMT)
3. Local

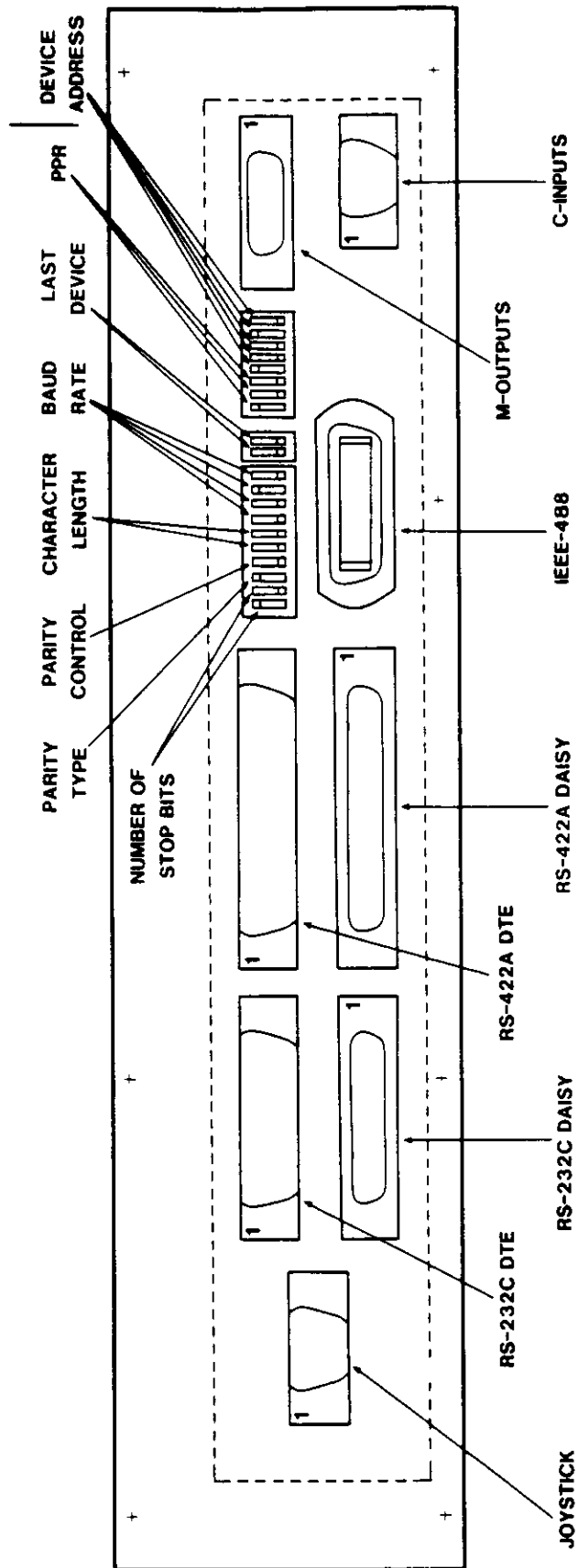


FIGURE 3-3.4 INTERFACE BOARD (REAR PANEL)

Also located on the front panel are:

1. EXECUTE key
2. PRINT key
3. RESET key
4. Eight character alphanumeric display
5. X and Y tracking displays with independent clear buttons

The front panel keys' functions are discussed here and on the front panel illustration (figure 3-4.1).

A. REMOTE_SWITCH_AND_LOCAL_LED

When first powered up, both remote and local LEDs will be lit; this condition is called "local with remote enabled". If you press any keyboard key, the remote LED will go out. Unidex III will be in local mode. You may switch to "local with remote enabled" state by pressing the REMOTE key, causing REMOTE and LOCAL LEDs to light up again.

B. RESET

When you press RESET, the state of the machine is restored to power-up conditions.

C. IMMEDIATE

Causes Unidex III to enter the Immediate mode.

D. EDIT

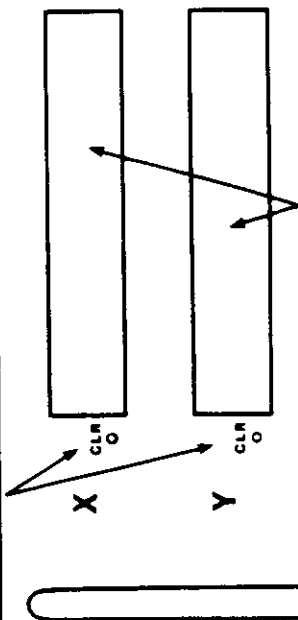
Causes Unidex III to enter the Edit mode.

E. AUTO

Causes Unidex III to enter the Auto mode.

UNIDEX III

12. X and Y TRACKING
DISPLAYS CLEAR



11. X and Y TRACKING DISPLAYS

☐ CZ
☐ M
☐ LMT

☐ CZ
☐ M
☐ LMT

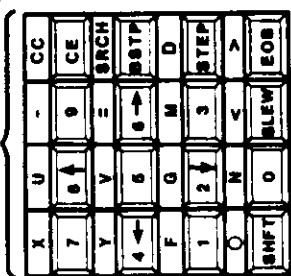


10. ALPHANUMERIC
DISPLAY

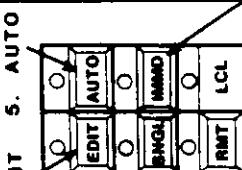
2. RESET. When RESET is pressed, the state of the machine is restored to power-up condition.



9. KEYBOARD



8. PRINT



6. SINGLE

4. EDIT 5. AUTO



3. IMMEDIATE



1. REMOTE KEY and LOCAL LED.
When first powered up, both Remote and Local LED's will be lit. This condition is called Local With Remote Enabled. Pressing any mode or keyboard key, will cause Unidex III to enter Local state.

7. EXECUTE. EXECUTE has two separate functions. In Edit, it is used to enter commands. During Edit "Search" function, EXECUTE key starts the search operation. In Auto, Single or Immediate, it causes the program or command string to run.

FIGURE 3-4.1 FRONT PANEL CONTROLS AND DISPLAY

F. SINGLE

Causes Unidex III to enter the Single mode.

G. EXECUTE

Executes program. In Immediate, Auto and Single, pressing EXECUTE causes the program to run. In other words, actual movement occurs.

Pressing EXECUTE when you enter the Edit mode causes the program to be displayed. If its a new program, two prompt signs will be displayed, which means you may begin to enter commands.

H. PRINT

When properly connected to a printer, Unidex III will give you a printout of your program. If connected to a host instead, PRINT, along with a program number, will cause the program to be sent over the lines to the host.

I. KEYBOARD

Enables you to enter commands.

J. ALPHANUMERIC_DISPLAY

As you type in your motion commands or call up a program stored in memory, the commands will appear on the alphanumeric display.

K. X_AND_Y_TRACKING_DISPLAYS

Show actual positions of the axes.

L. X_AND_Y_DISPLAYS_CLEAR

Clear X and/or Y displays.

SECTION 3-5 JOYSTICK SPECIFICATIONS

Unidex III can interface with any joystick with the following specifications:

X - Clock

X - Direction (0 is negative direction, 1 is positive direction)

Y - Clock

Y - Direction (0 is negative direction, 1 is positive direction)

Each of these inputs in Unidex III has an internal pull-up resistor of 100K ohms, connected to ground.

The signal levels on the outputs may be TTL or CMOS compatible.

CHAPTER 4

PROGRAMMING OF UNIDEX III

SECTION 4-1 POWERING_UP

The first thing that Unidex III will do upon power-up is a self-test (ROM check and RAM check). At the end of the self-test, it will display the message "SYS RDY" (system ready). The following states exist upon power-up:

- Local with remote enabled
- Incremental mode
- Non-corner-rounding mode
- Display enable mode

Unidex III can operate in either local mode or remote mode. Once in remote, it can use but one communication bus: IEEE-488, RS-232C or RS-422A. For a pictorial description, refer to figure 4-1.1.

SECTION 4-2 LOCAL PROGRAMMING

After power-up, if a key on the front panel other than REMOTE is pressed, Unidex III will go into local mode. In either local or remote enable mode, you may enter Auto, Single, Immediate or Edit mode from the front panel.

CAUTION: MAKE SURE DRIVES ARE CLEAR BEFORE BEGINNING A PROGRAM.

A. EDIT MODE

Unidex III must be in the Edit mode when you enter a new program or modify an existing one.

To create a program, you type in a sequence of motion control commands from the front panel keyboard. The commands are stored in the user RAM in a compressed format (redundant characters

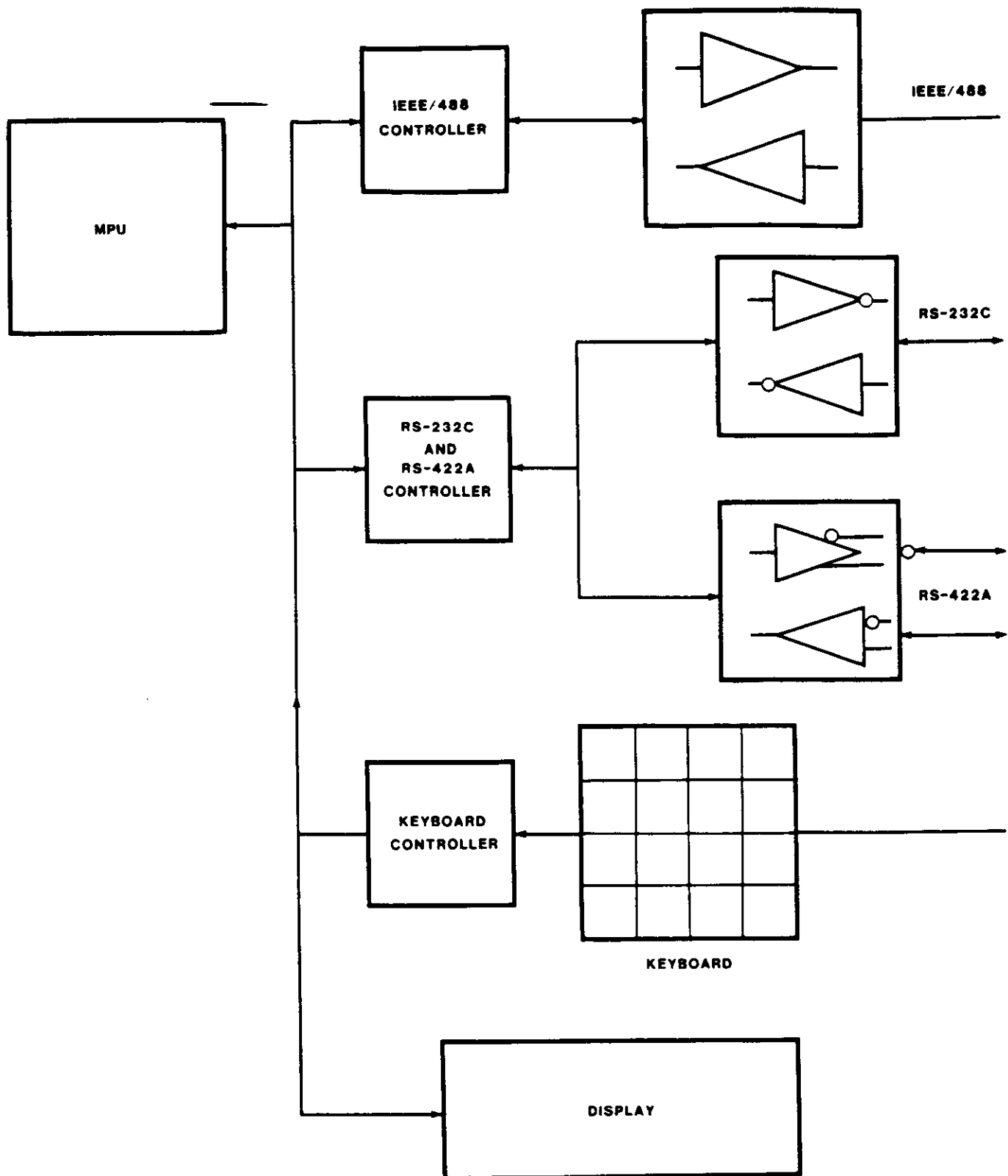


FIGURE 4-1.1 UNIDEX III LOCAL AND REMOTE MODES OF COMMUNICATION

are not stored). As data is typed, characters are displayed, moving right to left. Typing a letter (alpha) automatically enters the previous command (alpha followed by numerals as shown in chapter 5, section 6). EOB (end-of-block) will enter the previous command, as well.

The EDIT key provides the means of evoking, for the purpose of entering or editing a program, any specific program number (ranging from 1 to 99). To do so, you proceed as follows:

EDIT	PGM__<1>
2	EDIT___2
0	EDIT__20

NOTE: Program 1, the default program, will be displayed after power-up. Once a program has been entered, this is no longer the case.

1. Entering A New Program

If program 20 is a new program you will see the prompt sign on your display. Example:

EDIT	PGM__<1>
2	EDIT___2
0	EDIT__20
EXEC	----->>

You may now type in the new motion program.

2. Editing An Existing Program

If the program requested is an existing program which you wish to edit, the prompt sign, as well as the first command of the program, will be displayed. Example:

EXEC	----->>G7
------	-----------

3. Edit Functions

In order to facilitate your program editing, the following edit functions are available to you in the local mode.

a. STEP and BACKSTEP

These keys enable you to step through a program in a forward or reverse direction, command by command. The program "wraps around" so that stepping through the complete program in either direction will bring you back to the starting point. Example:

POWER UP	SYS__RDY
EDIT	PGM__<1>
0	EDIT__0
2	EDIT__02
EXEC	----->>G7
STEP	>>G7X100
BSTP	----->>G7
BSTP	----->>
BSTP	00Y100M2

b. SEARCH

The SEARCH key allows you to look for a particular string of characters (maximum length - 8 characters). To search for a string, press SHIFT and then SEARCH. When the display goes blank, type in the command for which you are searching. The word "LOOKING" will be seen on the display. When the command is found, it will be displayed on the screen. Example:

SHIFT	SRCH	-----
SHIFT	X150	____X150
	EXEC	__LOOKING
		QE20X150

If the command cannot be found, the word "WHAT?" will be displayed instead. Example:

SHIFT	SRCH	-----
SHIFT	X300	____X300
	EXEC	__WHAT?__

Once the command is found, any changes can be made. You can delete a command by pressing SHIFT followed by CLEAR COMMAND. You can add new commands by simply typing them in at the keyboard. To search for another string, repeat the above procedure. Example:

SHIFT	SRCH	-----
SHIFT	Y250	____Y250

EXEC		<u>LOOKING</u>	
		<u>OF10Y250</u>	
SHFT	CC	<u>OX100F10</u>	(delete command)
SHFT	Y	<u>X100F10Y</u>	
	4	<u>100F10Y4</u>	
	0	<u>00F10Y40</u>	
	0	<u>0F10Y400</u>	(add new command)

c. SLEW

Refer to chapter 2, page 6.

d. CLEAR ENTRY

The CLEAR ENTRY (CE) key enables you to erase a single character. The display moves from left to right, erasing the character on the extreme right. Example:

SHFT	SRCH	-----
SHFT	F200	----F200
	EXEC	<u>LOOKING</u>
		<u>1000F200</u>
	CE	<u>X1000F20</u>
	CE	<u>1X1000F2</u>
	7	<u>X1000F27</u>
	5	<u>1000F275</u>

e. CLEAR COMMAND

The CLEAR COMMAND (CC) key enables you to do one of three things:

1. CC allows you to erase the command on the far right side of your display screen. Example:

<u>SHFT</u>	<u>SRCH</u>	-----
<u>SHFT</u>	<u>Y5000</u>	___Y5000
	<u>EXEC</u>	<u>250Y5000</u>
	<u>CC</u>	<u>1000F250</u>

2. CC allows you to erase an entire program. In order to clear a program, you type in the program number upon request; then press SHIFT, CC and EXECUTE, in that order. Example:

<u>EDIT</u>	PGM__<1>	
<u>1</u>	EDIT___1	
<u>5</u>	EDIT__15	
<u>SHFT</u>	<u>CC</u>	CLR___15
<u>EXEC</u>		_CLEARED

3. CC allows you to erase the entire memory. In order to do this, type in zeroes for the program number; then press SHIFT, CC and EXECUTE, in that order. Example:

<u>EDIT</u>	EDIT__15
-------------	----------

	<input type="text" value="0"/>	EDIT__50
	<input type="text" value="0"/>	EDIT__00
<input type="text" value="SHFT"/>	<input type="text" value="CC"/>	CLR__00
	<input type="text" value="EXEC"/>	_CLEARED

B. AUTO_MODE

In the Auto mode, programs stored in the user RAM may be executed from beginning to end whenever the EXECUTE key is pressed.

In order to start a program, press the AUTO key. The program number will appear, which will be the default program number if you have just powered-up and the previous program number if you have already entered and/or executed one. Type in the intended program number and then press EXECUTE. Your program will run from start to finish without requiring any further input from you. When the program is completed, the word "READY" will be displayed.

If you require the program to be executed again, press EXECUTE. To run another program in the Auto mode, press AUTO and repeat the above procedure. Example:

POWER UP	SYS__RDY	
<input type="text" value="AUTO"/>	PGM__<1>	
<input type="text" value="0"/>	RUN___0	
<input type="text" value="2"/>	RUN___02	
<input type="text" value="EXEC"/>	_RUNNING	(program 2 runs)
	__READY__	
<input type="text" value="EXEC"/>	_RUNNING	(program 2 runs again)
	__READY__	

<input type="button" value="AUTO"/>	RUN___02	
<input type="button" value="1"/>	RUN___21	
<input type="button" value="0"/>	RUN___10	
<input type="button" value="EXEC"/>	_RUNNING	(program 10 runs)
	__READY__	

C. SINGLE_MODE

If the SINGLE key is pressed, execution is accomplished block by block. Each time you press EXECUTE, one block (up to end-of-block) is executed.

To start a program, press SINGLE. When the program number appears, type in the desired number. As in the Edit and Auto modes, upon power-up, program #1 will be displayed. Afterward, the program previously edited or executed will be the number to appear. After typing in the appropriate program number, press EXECUTE. Example:

POWER UP	SYS__RDY	
<input type="button" value="SNGL"/>	PGM__<1>	
<input type="button" value="2"/>	WALK___2	
<input type="button" value="0"/>	WALK__20	
<input type="button" value="EXEC"/>	_RUNNING	(runs first block of program)
	__READY__	
.	.	
.	.	
.	.	
<input type="button" value="EXEC"/>	_RUNNING	(executes last command)

PGM__END

As can be seen in this example, after the last block has been executed, the message "PGM END" (program end) will be displayed.

D. IMMEDIATE_MODE

Pressing the IMMEDIATE key puts Unidex III in the Immediate mode. This mode is like Edit, in that it enables you to enter a short program and run it immediately by pressing the EXECUTE key. All of the features of the Edit mode are available to you, except the ability to clear programs.

The program you enter in Immediate mode is stored in a special 256 byte buffer. When you press EXECUTE the program in this buffer will run. You may run it repeatedly by pressing EXECUTE. You can add commands to the program with or without pressing IMMEDIATE. Pressing any of the other mode keys however, will clear the buffer and your program will be lost. Example:

	IMMD	ENTER_>>	
SHFT	G	_____G	
	7	_____G7	
SHFT	X	_____G7X	
.		.	
.		.	
.		.	
	EXEC	_RUNNING	
		__READY__	(program complete)
	EXEC	_RUNNING	(runs again)

AUTOPGM__<1> (Immediate
program is lost)

As can be seen in this example, pressing any mode key, except IMMEDIATE, clears the buffer.

E. MANUAL_CONTROL

As mentioned in chapter 2, section 4E, the manual operations are STEP and SLEW. They are both accessible in the Local mode only.

1. Step_Function

The Step function is useful for precise positioning of the drives. After you depress the STEP key, the drives will move one step each time one of the arrow keys is pressed (figure 3-4.2). The Step function requires no feedrate. Example:

IMMD

ENTER_>>

STEP

__STEP__

You may now use the arrow keys for positioning the drives.

2. Slew_Function

The Slew function is useful for rapid manual positioning of the drives. The speed of this movement is determined by the feedrate(s) entered. You can enter the feedrate in either steps per second (frequency), which is the Fnnnn command, or microseconds (pulse period), which is the F=nnnn command.

To operate in Slew, enter the Immediate mode, enter the feedrate and depress SLEW. Pressing and holding one of the arrow keys

will cause one of the drives to move in a given direction until the key is released.

You may enter a feedrate which will apply to both axes (ie., a single feedrate). Example:

	IMMD		ENTER_>>
SHFT	Fnnn	or	SHFT F=nnnn
	SLEW		__SLEW__

The feedrate entered can also be applied to one axis only (ie., individual feedrates for X and Y). Example:

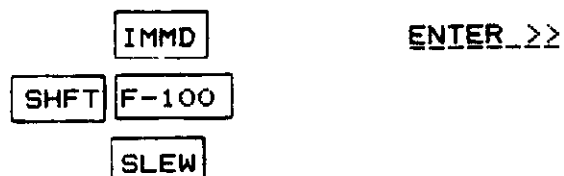
SHFT	IMMD		ENTER_>>
SHFT	X	Fnnn	Y Fnnn
	SLEW		__SLEW__

You may now position the drives through the use of the arrow keys.

If no feedrate is entered and an arrow key is pressed, the default feedrate automatically entered is 250 steps per second after power up. Any feedrate entered is valid until changed.

3. Slew Function Using Joystick

The Slew function may employ the joystick rather than the arrow keys for positioning. The feedrate for joystick operation is a fraction of the joystick frequency and is entered with the F=nnnn command. The range of dividing factors you may choose are from 2 to 65535. Example:



You may now position the drives through the use of the joystick.

If you do not enter a feedfactor and begin to operate the joystick, the default factor of F-2 is automatically entered, thereby dividing the joystick frequency by a factor of 2.

Pressing EXECUTE is unnecessary when in either mode.

To exit the Step or Slew mode, press either CLEAR ENTRY or CLEAR COMMAND.

These manual functions are also usable whenever the system is ready after power up or after running a program or a block of program.

SECTION 4-3 REMOTE PROGRAMMING

When Unidex III powers up or whenever the REMOTE key on the front panel keyboard is pressed, Unidex III goes into remote enable mode. The source of the next command (keyboard or host) determines to what programming mode (local or remote) Unidex III will switch.

With the exception of the Edit functions and the manual controls of Step and Slew, any of the modes available in local are available in remote as well.

Once in the remote mode, only one of the interfaces (IEEE-488, RE-232C or RS-422A) may be used.

Unidex III is factor-set to use RS232C or IEEE-488. To use RS-422A instead of RS-232C, jumper plug on socket M78 on the MPU board must be removed and inserted into socket M77. You then simply plug into the appropriate connector (refer again to figure 3-3.4).

If serial interface is employed, baud rate, character length, the number of stop bits, parity/non-parity and odd/even parity must be selected on the rear panel. The settings you choose should match the settings on your controller (host). For switch settings, refer to figure 4-3.1.

In remote, commands to switch to Edit, Auto, Single or Immediate mode may be sent over the lines, using characters "E" for Edit, "A" for Auto, "S" for Single and "I" for Immediate. Follow the character with CR LF (carriage return/line feed) to execute the command. For a complete ASCII character listing see appendix 1, table 4-3.1.

In the remote mode, transmitting the characters for carriage return/line feed (abbreviated CR LF or <CR><LF> in the following text), produces the same result as does pressing the EXECUTE key when in the local mode.

A special remote programming feature that applies when using multiple Unidex IIIs is the Trigger command. The Trigger mode makes all devices wait

BAUD RATE SELECTOR SWITCHES

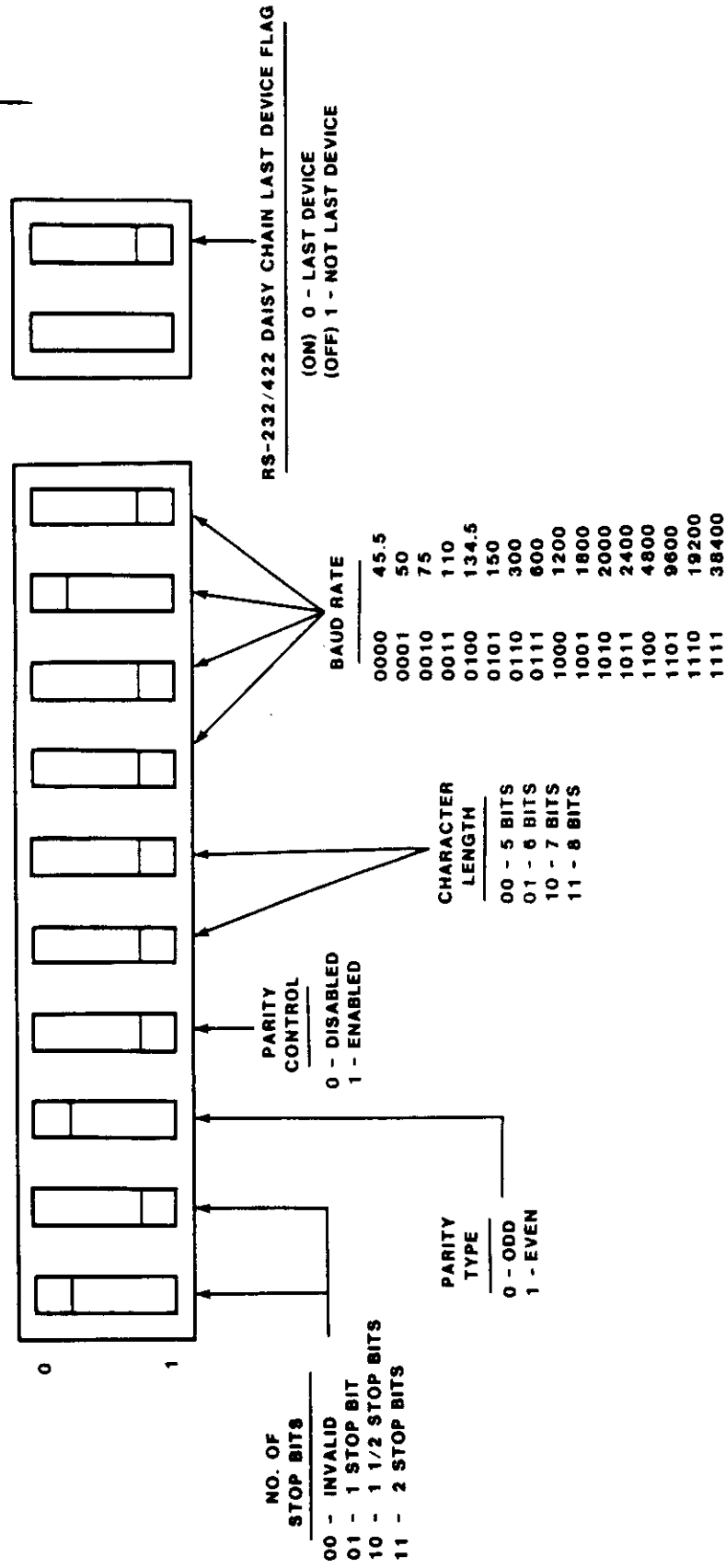


FIGURE 4-3.1 SELECTOR SWITCHES LOCATED ON DISTRIBUTOR CARD (BACK OF CHASSIS)

CHARACTER	MEANING	CHARACTER	MEANING	CHARACTER	MEANING
0	DIGIT 0	DEL (RUBOUT)	DEVICE RESET #	T	TRIGGER MODE #
1	DIGIT 1	A	AUTO MODE	U	U CODES
2	DIGIT 2	B	CONTINUE PROGRAM AFTER PROGRAM INITIATED SKO (G25)	V	V CODES
3	DIGIT 3	C	DEVICE CLEAR #	W	CONFIGURE SRO USING PROGRAMMABLE CHARACTER #
4	DIGIT 4	D	DWELL	X	X CODES
5	DIGIT 5	E	EDIT	Y	Y CODES
6	DIGIT 6	F	FEED	Z	CANCELS W COMMAND #
7	DIGIT 7	G	G CODES		
8	DIGIT 8	H	HOLD FOR TRIGGER		
9	DIGIT 9	I	IMMEDIATE MODE		
-	MINUS	J	CONFIGURE SERIES RTS #		
=	EQUAL TO	K	CONFIGURE PARALLEL RTS #		
>	GREATER THAN	L	GO TO LOCAL #		
<	LESS THAN	M	MISCELLANEOUS CODE		
!	START OF COMMENT FIELD	N	LINE NUMBER		
\$	ERASE (EDIT MODE)	O	CANCEL HOLD #		
*	END-OF-BLOCK	P	PRINT: Pnn - Print program nn PD - " " directory POO - " " all programs PX - " " X position PY - " " Y position PS - " " statuses		
CR LF	CARRIAGE RETURN/ LINE FEED (EXECUTE)	Q	QUERY (SERIAL POLL) #		
EOI	END OR IDENTIFY (EXECUTE)	R	REMOTE ENABLE (QUIT CURRENT MODE AND DISPLAY "RMT ENAB")		
ESC	BEGINS BUS INTER-FACE COMMANDS #	S	SINGLE MODE		

* (APPLIES TO RS-232/422)

TABLE 4-3.1 ASCII CHARACTER SET

(send "H" for the Hold command) until it is time for them to operate simultaneously (send "T" for the Trigger command).

A. EDIT MODE

In the remote mode of programming, Unidex III's detail editing functions, ie., Step/Backstep, Search, Clear Entry and Clear Command, are not accessible to you. You are limited to entering, storing and deleting programs.

1. Entering a Program

To enter a program into Unidex III via your host, transmit the mode commands "E" (edit), "nn" (program number) and CR LF (execute). Next, send the motion program commands followed by an M2 or M30 and CR LF at the end of the program.

NOTE: How you communicate with Unidex III depends on what type of host you are using for remote programming. This manual gives the commands needed to program Unidex III, not the language required by any particular host. Details of addressing Unidex III(s) on the bus is given in chapter 5, section 5-5A.

Example:

```
"E 20 CR LF"           (edit program 20)

"G7 X2000 F200 Y1000 M2 CR LF" (enter
                                commands)

"R CR LF" Quit Edit mode - you may now
                                address another Unidex III
```

You may include CR LF anywhere in the file, prior to M30. These will be ignored by Unidex III. You may want these extra CR LF commands so that the print file (for outputting on your host's printer) and the program file are one and the same.

2. Storing A Program

Once the program has been downloaded, it is stored in the Unidex III memory. You may now edit it locally, utilizing the edit functions that are available only in the local mode.

After you have modified the program from the front panel of Unidex III, you may read back the entire program into the host, if you should wish to store it there as well as in Unidex III. To do this, press PRINT on the front panel keyboard, enter the program number (if it is the program you were just editing, the correct number will come up automatically) and hit EXECUTE. Example:

<div>PRNT</div>	PRNT__20
<div>EXEC</div>	-----
	PRINTING

If you have just powered up, however, the following steps would apply:

<div>PRNT</div>	PGM__<1>
<div>2</div>	PRNT__2
<div>0</div>	PRNT__20
<div>EXEC</div>	-----
	PRINTING

If you require that all of the programs stored in Unidex III be stored in the host computer as well, the program number to be entered is zero-zero. Example:

PRNT	PRNT__20
0	PRNT__00
EXEC	-----
	PRINTING

3. Deleting A Program

In order to delete a program that is stored in Unidex III while in the remote mode, transmit an "E" followed by a dollar sign, the program number and CR LF. Example:

"E \$ 35 CR LF"

To erase the entire memory, send two zeroes for the program number. Example:

"E \$ 00 CR LF"

B. AUTO_MODE

Once a program is stored in Unidex III's memory, you may execute it in the Auto mode. The Auto mode enables the program to run straight through, beginning to end, without any further user intervention.

To get a program to run in the Auto mode, transmit "A" for Auto, program number and CR LF. Example:

"A 25 CR LF"

(program 25 runs)

When the program is complete, Unidex III will request service (SRQ) and wait for a serial poll. In order to execute the program again, do a serial poll of Unidex III (chapter 4, section 4A and 4B) and send CR LF. The same program will run again.

To run another program instead, do a serial poll of Unidex III and then transmit "A", program number desired and CR LF, as in the above example.

C. SINGLE_MODE

In the Single mode, Unidex III will set SRQ (service request) after each block. Therefore, in the Single mode, you must do a serial poll after each block. After the serial poll, send a CR LF on the lines in order to get the Unidex III to execute the next command. When you first enter the Single mode, send the program number and CR LF. Example:

"S 22 CR LF" (executes first command block)

After the execution of the first command block, simply do a serial poll and send another CR LF for each following block of commands.

D. IMMEDIATE_MODE

The Immediate mode in remote is similar to the remote operation of Unidex II, in that execution begins as soon as Unidex III receives CR LF. After execution of the block is complete, Unidex III interrupts the host and waits for a serial poll. As mentioned previously, commands are not part of a prewritten program and are not stored in memory in the Immediate mode. Example:

"I"

"G7 X5000 F1000 Y2500 . . ."

"CR LF"

(command block executes)

Unidex III will now request service. After the host takes a serial poll, the next command string may be entered and executed.

SECTION 4-4 REMOTE_INTERFACING

Any of the three interfaces (IEEE-488, RS-232C or RS-422A) transfer control signals (mode commands) and data signals (motion commands) to and from the host (controller) and Unidex III (or multiple Unidex IIIs).

These signals and data are transferred in the ASCII format. ASCII (American Standard Code for Information Interchange) is a standard means of representing characters as digital codes. Each of the letters (upper and lower case) of the alphabet, as well as various symbols, are represented by 128 binary combinations, 7 bits each. Refer to table 4-3.1 for the ASCII character set used by Unidex III. (Tables are located in appendix 1 as well as in text.)

A. IEEE-488_INTERFACE_BUS

A bus is a resource shared by all devices connected to it. In order to avoid confusion, the bus has a precise organization and specific protocol.

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

1. Listener
2. Talker
3. Controller

1. Device_As_Listener

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

2. Device_As_Talker

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

3. Device_As_Controller

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

Listener, talker and controller capabilities can occur individually or in combinations. For instance, devices such as the Unidex III 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.

Table 4-4.1, appendix 1, lists all of the possible address combinations for Unidex III. Which address is selected for each Unidex III within your system is left to your discretion.

4. Signal Lines Of The IEEE-488 Bus

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

Eight of the lines are for the transfer of data (DIO1 to DIO8).

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

The remaining five lines, for example "ATN" (attention) and "SRQ" (service request), are for bus management. Each line, when asserted low (ground), represents a single line message sent on the bus. A description of these lines is given in table 4-4.2.

5. Cable Restrictions Of The IEEE-488 Bus

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

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

The maximum number of devices on the bus is limited to 14. The total length of the cable is limited to 20 meters or 2 meters multiplied by the number of devices (whichever is shorter in length). For a complete cable listing, refer to table 4-4.3.

6. Parallel And Serial Polling

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

SWITCH					DECIMAL ADDRESS
5	4	3	2	1	
OFF	OFF	OFF	OFF	OFF	31*
OFF	OFF	OFF	OFF	ON	30
OFF	OFF	OFF	ON	OFF	29
OFF	OFF	OFF	ON	ON	28
OFF	OFF	ON	OFF	OFF	27
OFF	OFF	ON	OFF	ON	26
OFF	OFF	ON	ON	OFF	25
OFF	OFF	ON	ON	ON	24
OFF	ON	OFF	OFF	OFF	23
OFF	ON	OFF	OFF	ON	22
OFF	ON	OFF	ON	OFF	21
OFF	ON	OFF	ON	ON	20
OFF	ON	ON	OFF	OFF	19
OFF	ON	ON	OFF	ON	18
OFF	ON	ON	ON	OFF	17
OFF	ON	ON	ON	ON	16

SWITCH					DECIMAL ADDRESS
5	4	3	2	1	
ON	OFF	OFF	OFF	OFF	15
ON	OFF	OFF	OFF	ON	14
ON	OFF	OFF	ON	OFF	13
ON	OFF	OFF	ON	ON	12
ON	OFF	ON	OFF	OFF	11
ON	OFF	ON	OFF	ON	10
ON	OFF	ON	ON	OFF	9
ON	OFF	ON	ON	ON	8
ON	ON	OFF	OFF	OFF	7
ON	ON	OFF	OFF	ON	6
ON	ON	OFF	ON	OFF	5
ON	ON	OFF	ON	ON	4
ON	ON	ON	OFF	OFF	3
ON	ON	ON	OFF	ON	2
ON	ON	ON	ON	OFF	1
ON	ON	ON	ON	ON	0

* NOT USED

TABLE 4-4.1

IEEE-488 AND RS-232/422 DAISY CHAIN

IEEE-488 STANDARD INTERFACE BUS SIGNAL LINE

MNEMONIC	NAME	FUNCTION
<u>BUS MANAGEMENT LINES (UNILINE MESSAGES)</u>		
IFC	INTERFACE CLEAR	System controller alone can assert this line, to place all devices in the unaddressed state. Devices go into Talker idle/Listener idle state. If control has been passed to another device, system controller again becomes active by asserting IFC.
ATN	ATTENTION	Asserted true by active controller to send bus interface messages on the bus. When ATN is asserted, signals on the data lines are interpreted as messages. ATN asserted with EOI to do a parallel poll. When ATN is false, data may be sent over the bus by a designated talker.
REN	REMOTE ENABLE	Asserted to program devices on the bus remotely. Any device addressed to listen while REN is true, is placed in Remote mode of operation.
SRQ	SERVICE REQUEST	Asserted by a device to indicate its need for interaction with the controller.
EOI	END OR IDENTIFY	When asserted, indicates the termination of flow of data. Asserted when the last data byte is placed on the bus.
<u>DATA HANDSHAKE LINES</u>		
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 the 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.
<u>DATA LINES</u>		
DIO-1 - DIO-8	DATA LINES	Used for sending data (ATN line false) or Bus interface messages (ATN line true).

a. Parallel Polling

In the parallel poll, you may assign each of the eight data lines of the bus to as many as eight different devices.

The parallel poll bit assigned to each Unidex III may be selected via the dipswitch labeled PPR, located on the back panel. Address selection may be between 0 and 7 in binary numbers (figure 4-4.1).

b. Serial Polling

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

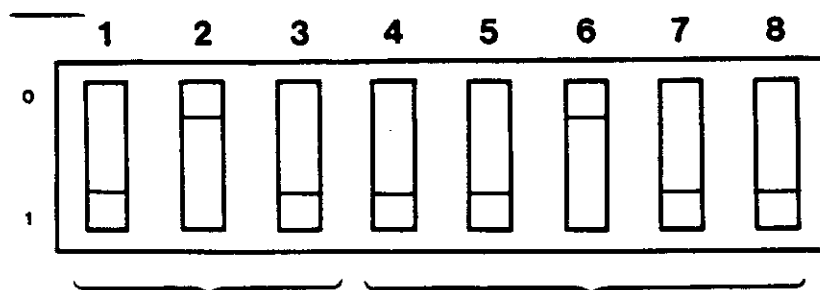
A Unidex III will request service (set SRQ) at specific times, such as when a program is completely executed. At such a time, further operations will be suspended until Unidex III is serial polled by the controller (host). Upon being polled, the Unidex III will transmit its status.

B. RS-232C/RS-422A INTERFACE

Either RS-232C or RS-422A can be implemented as an interface instead of IEEE-488. Including the host, up to 31 devices may be "daisy chained" on the serial line. The possible address combinations are listed in table 4-4.1, found earlier in this section and in appendix 1.

As mentioned in the previous section, the number assigned to each Unidex III is selected by you.

IEEE - 488 ADDRESS SELECT SWITCHES



PARALLEL POLL CONFIGURE DEFINES PARALLEL POLL RESPONSE

000 - PPR 1
 001 - PPR 2
 010 - PPR 3
 011 - PPR 4
 100 - PPR 5
 101 - PPR 6
 110 - PPR 7
 111 - PPR 8

IEEE-488 BUS ADDRESS VALID ADDRESSES

0 DECIMAL TO 30 DECIMAL
 00000 TO 11110

IEEE - 488 PARALLEL POLL REGISTER

(SIGNAL ON THE DATA LINES)

8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---

PPR 1	0	0	0	0	0	0	0	1
PPR 2	0	0	0	0	0	0	1	0
PPR 3	0	0	0	0	0	1	0	0
PPR 4	0	0	0	0	1	0	0	0
PPR 5	0	0	0	1	0	0	0	0
PPR 6	0	0	1	0	0	0	0	0
PPR 7	0	1	0	0	0	0	0	0
PPR 8	1	0	0	0	0	0	0	0

**FIGURE 4-4.1 IEEE - 488 ADDRESS SELECT SWITCHES
 LOCATED ON INTERFACE BOARD
 (BACK OF CHASSIS)**

IEEE-488 CABLE MANUFACTURERS

Hewlett-Packard

Palo Alto, California 94304

<u>PN</u>	<u>Length</u>
HP 10833D	.5 Meter
HP 10833A	1 Meter
HP 10833B	2 Meters
HP 10833C	4 Meters
HP 10834A	Adapter

Belden Corporation

Richmond, Indiana 47374

<u>PN</u>	<u>Length</u>
9642	1 Meter
9643	2 Meters
9644	4 Meters
9645	8 Meters
9646	16 Meters

1. Signal Transmission

Transmitted and received data, as well as five control signals, are accommodated by the 2 lines. The five control signals are:

1. RTS (request to send)
2. CTS (clear to send)
3. DTR (data terminal ready)
4. DSR (data set ready)
5. DCD (data carrier detect)

When using RS-232 interface, refer to figure 4-4.2 for Unidex III to host connections. When using RS-422, refer to figure 4-4.3.

The five control signals are handled by the interface software and require no user intervention, except for RTS (request to send) which is used as a service request.

RTS can be arranged by the use of a special code (J/K) into either of two configurations. These alternate arrangements are:

1. SERIES - ALL Unidex IIIs must raise their RTS lines before a raised RTS is presented to the host. This arrangement is established by using the J code.
2. PARALLEL - The RTS line will be raised to the host when ANY Unidex III raises its own RTS line. This arrangement is established by the use of the K code.

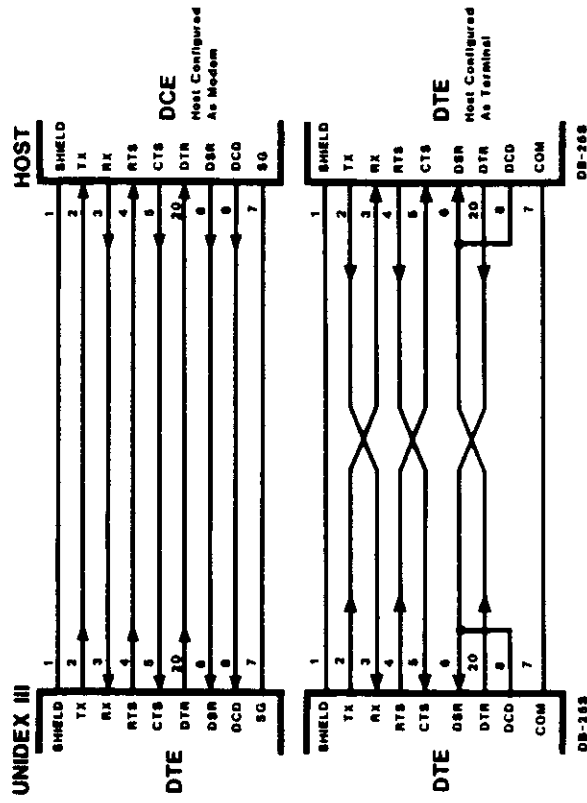


FIGURE 4-4.2 RS-232 CONNECTION

FIRST SEGMENT ASSIGNMENT			SECOND SEGMENT ASSIGNMENT				CIRCUIT CATEGORY	DIRECTION		RS-232 EQUIVALENT PROTECTIVE GROUND (AA)
CONTACT NUMBER	CIRCUIT	INTERCHANGE POINTS	CONTACT NUMBER	CIRCUIT	INTERCHANGE POINTS			TO DCE	FROM DCE	
1	SHIELD									
2			20							
3			21							
4	SD	A-A'	22	SD	B-B'		I	X		TX (BA)
5			23							
6	RD	A-A'	24	RD	B-B'		I		X	RX (BB)
7	RS	A-A'	25	RS	B-B'		I	X		RTS (CA)
8			26				I			
9	CS	A-A'	27	CS	B-B'		I		X	CIS (CB)
10			28							
11	DM	A-A'	29	DM	B-B'		I		X	DSR (CC)
12	TR	A-A'	30	TR	B-B'		I	X		DIR (CD)
13	RR	A-A'	31	RR	B-B'		I		X	DCD (CF)
14			32							
15			33							
16			34							
17			35				I	X		
18			36							
19	SG	C-C'	37	SC	C-B'		I	X		SG (AB)

NOTE: BLANK ENTRIES IN TABLE ABOVE ARE NOT USED.

FIGURE 4-4.3 ASSIGNMENTS FOR 37-POSITION RS-422 CONNECTOR USED IN UNIDEX III

You will most likely want the Series arrangement if, for example, you are using the host to control several Unidex IIIs in a multi-axis remote/immediate mode under interrupt control.

The parallel configuration emulates operation of the IEEE-488 bus more closely in that each Unidex III on the bus is essentially an independent device.

If you have only one Unidex III on the RS-232/422 line (plus a host), parallel and series will function identically.

a. Addressing Unidex III

You may address one Unidex III.
Example:

<ESC> nn <CR LF>

You may address multiple devices on the line. The following example addresses three devices:

<ESC> nn, mm, ll <CR LF>

For more details, refer to chapter 5, section 5-5A.

b. Programming RTS

It may be necessary for you to use a character for SRQ instead of RTS. If you need to program an RTS, send the codeword "W", followed by any ASCII character you choose. Example:

"ESC nn CR LF W % CR LF"

The user programmable ASCII character that may represent RTS may be a letter, number or symbol, but it must be limited to one character (in other words, you cannot use a double digit).

The RS-232 interface will support full duplex operation in accordance with RS-232 protocol. When you use a programmed character for RTS, Unidex III will not raise RTS for SRQ.

Unidex III uses the DC1 (X ON), DC3 (X OFF) protocol for controlling transmissions.

b. Switch Settings On Back Panel

When implementing RS-232/422, the switch settings on the back panel of your Unidex III must match those of your controller.

Baud rate is switch selectable from 45.5 to 38400. Character length is variable from 5 bits to 8 bits with 1, 2 or 1 1/2 stop bits.

Parity may be selected to be on or off. If on, odd or even parity may be selected. All of these switches are located on the rear panel of the chassis (refer again to figure 4-3.1).

Multiple Unidex IIIs may be daisy chained, using the serial port. The units connected in a daisy chain fashion may be individually addressed by the host.

Two RS-232 connectors are provided on the rear panel. You simply connect the host to the connector marked "DTE" on the Unidex III that is to be first in a chain of devices. You then connect the "daisy" connector on this

first Unidex III to the "DTE" connector on the second device of the chain using a one to one cable as in the top illustration of figure 4-4.1. Continue in this fashion for the entire chain of devices.

NOTE: The "last device" switch on the last device of the chain should be "ON".

If you have only one Unidex III on the RS-232/422 line, be sure to set it as the last device. To do this, set the "last device" switch to the "on" position (refer again to figure 4-3.1).

SECTION 4-5 PROGRAMMING CAPABILITIES

Unidex III is programmed in standard X, Y, F, D, M, N and G codes. The codes are extended to provide a full array of motion control capabilities (see chapter 5, sections 2 and 6).

Programming features which reduce the length of your program while increasing its versatility include:

A. UNCONDITIONAL JUMP

The unconditional jump allows the program to jump to a given line, regardless of internal or external conditions. Example:

"X2000 F100 Y2000 N>150 G10 . . ."

The N>150 command causes the program to unconditionally jump to line 150.

B. CONDITIONAL_SKIP

Skip to the end-of-block, depending on the status of one of the inputs (C1 - C4). These are program selected conditions. Example:

```

                                C1 true
                                |
                                v
"G7 X100 F10 Y200 F20 G271 X5000 F500 Y2000 *
                                |
                                C1 false_ _ _ _ _ _ _ _ ^

```

The G271 command causes the program flow to continue if the C1 input is high and to skip to the end-of-block if it is low. Therefore, the conditional skip depends on an existing condition. It also requires the EOB command to set-up program direction.

C. ABORT_ON_INTERRUPT

Abort the current move and go to the end-of-block, when one of the inputs, C1 - C4 (program selected), causes an interrupt because of changing from logic true to logic false (high voltage to low). Example:

```

"G7 G301 X1000 F500 Y2000 F1000 * G531 M-58"
      |
      ^
      C1 goes T to F_ _ _

```

The G301 command allows you to abort a move, stop or dwell immediately upon C1 going from high to low, causing an interrupt. In the above example, the move "X1000 F500 . . ." is being executed. As soon as the input C1 goes from logic true to logic false, the move is aborted and the program skips to end-of-block. The Absolute Position registers are updated with the actual segment of the move performed. G531 tests the flag set by the interrupt and skips to end-of-block if set.

D. SUBROUTINE_CALL

Call to subroutine starting at given line number. If the N-ddd command for a subroutine is given, all statuses (feedrate, corner-rounding/non-corner-rounding, absolute mode/incremental mode) of the subroutine are preserved when returned to main program.

If the N=ddd command is given, the main program statuses, not those of the subroutine, are maintained upon return to the main program.

Subroutine example:

```
"G7 X1000 . . .N-10000 . . .M2  
N10000 X10 F10 Y10 F10 M99 *"
```

To avoid confusion, it is best to put all subroutines after M2 (end of program) as shown above.

At the end of the subroutine, place an M99 command (return from subroutine) and an EOB.

If, in the above example, N=10000 had been used instead of N-10000, all of the main program statuses would have been preserved after the subroutine. Otherwise, as shown above, the subroutine statuses will become the main program statuses after the subroutine is completed.

E. REPEAT_CYCLES

Execute a specific block of user program a predetermined number of times. Repeat counters (down-counting) are provided. Example:

```
". . . G661=10 *  
N100 *  
X100 F100 Y100 F100 *
```

```
G671 N>100 *
```

```
M=217 . . ."
```

The G661=10 command loads the repeat counter #1 with the number 10. Line number 100 begins the program loop, which will be repeated 10 times before the program continues.

The G671 command decrements the repeat counter and if zero, skips to the EOB. N>100 causes the program to jump back to line 100. When the decrement counter reaches zero, the cycle will not be repeated. Instead the program will skip to EOB and continue.

F. SEND_INFORMATION_ON_COMMAND

Provides a Service Request upon command. A G code command (G25) which, when executed, will cause Unidex III to request service and wait for a serial poll from the host.

Unidex III will complete execution of motion program up to command G25, before requesting service. Send character to continue program. Example:

```
"<B> <CR LF>"
```

In order to receive information on statuses and on positions of X and Y, send a "P" along with X, Y or S. Example:

```
PX - print X position  
PY - print Y position  
PS - print statuses
```

Please refer to addendum 1 for a detailed explanation of the status bytes.

SECTION 4-6 MEMORY CAPABILITIES

Unidex III is able to store up to 32K bytes of user program. Four sockets provide the following:

- 2 x (2K x 8) - 4K bytes - standard
- 4 x (2K x 8) - 8K bytes - optional
- 2 x (8K x 8) - 16K bytes - optional
- 3 x (8K x 8) - 24K bytes - optional
- 4 x (8K x 8) - 32K bytes - optional

Expanding memory will require changing a specifically programmed PAL (programmable array logic device) as well as some jumpers.

Battery back-up will support up to 32K bytes of RAM.

When loading a program from either the keyboard or a host, an end-of-memory warning will be provided 64 bytes before the real end of memory. In local, you will receive the message "FREE 64B". In remote, Unidex III will set SRQ.

CHAPTER 5

COMMANDS FOR PROGRAMMING UNIDEX III

The language used for programming Unidex III is discussed in this chapter. Included also are code summaries and sample programs.

SECTION 5-1 RS-274-D DATA FORMAT

RS-274-D is the standard elementary language adopted for programming a computerized numerically-controlled (CNC) machine. Adopting such a standard leads to uniformity within the industry and helps to avoid confusion. It permits easily effected interchangeability of data between motion control devices of the same order. RS-274-D uses a straightforward and easy-to-understand programming format.

Each command is represented by a code consisting of an alphabetic character, followed by one or more digits. Because of the letter-number(s) structure, this code is easy to write and read, both for the programmer and for the machine that executes it. This simplicity of language structure permits rapid execution of the code, which is essential in motion control.

For the sake of convenience in describing the rules of RS-274-D, we group the commands into seven categories:

1. Axis commands (X and Y codes)
2. Feedrate commands (F codes)
3. Miscellaneous commands (M codes)
4. Dwell commands (D codes)
5. Sequence numbers (N codes)
6. End-of-block (EOB) command
7. Preparatory commands (G codes)

A. STRUCTURE OF RS-274-D

The structure of the RS-274-D code is self explanatory: letter followed by number(s).
Example:

"G7 X1000 F200 Y5000 F1000 M-58 D100 M47"

The first code, G7, sends all axes home. The second code, X1000, states that the X axis is to move 1000 steps, while the third code, F200, states that the feedrate for the X axis is to be 200 steps per second. The fourth and fifth codes state that the Y axis is to move 5000 steps at a feedrate of 1000 steps per second. The sixth code, M-58, calls for a data output of the number 58 at the miscellaneous function output port. The seventh code, D100, calls for a dwell of 100 milliseconds and the last code, M47, calls for a reexecution of the program.

The example shows spaces between commands for the sake of clarity. In actuality, spaces are neither needed nor provided within the language. You need not use them when putting codes into Unidex III.

The program is executed as follows:

G7	All axes sent home
X1000	X axis moves 1000 steps
F200	at 200 steps per second and
Y5000	Y axis moves 5000 steps
F1000	at 1000 steps per second
M58	When move is complete, specified data (58 in this case) is output on the M output connector
D100	Dwell, where program executes a 100 millisecond pause
M47	Return to program start, where program is executed again.

B. SEQUENCING OF RS-274-D COMMANDS

The format of RS-274-D includes writing commands in the sequence in which you wish them to be executed. The only exception to this is the feedrate (F) commands; they cannot stand alone. An F command must be preceded immediately by an X command or a Y command. There can be no substitution. With this sole exception, any command is legal.

In order to have X and Y commands execute simultaneously, no command can be inserted between them. There are several structures that allow for simultaneous movement, however. Examples:

Xnnnn Ynnnn (X and Y use previous feedrates)*

Xnnnn Fnnnn Ynnnn (X assumes new feedrate, Y
retains previous feedrate)*

Xnnnn Ynnnn Fnnnn (Y assumes new feedrate, X
retains previous feedrate)*

Xnnnn Fnnnn Ynnnn Fnnnn (X and Y assume new
feedrates)

* (If no previous feedrate has been entered,
you will get a feedrate error.)

In order to have X and Y commands execute successively, insert another command between the X and the Y commands. Example:

Xnnnn Fnnnn NO Ynnnn Fnnnn

NO is good for this purpose because the zero will not get confused with any other line number reference within the program.

As mentioned earlier in this section, any command is legal, with the exception of the

proper placing of the feedrates. However, not every code written represents a legal command. Which are legal commands will be discussed by group.

Each group will be explained on the following pages, followed by a complete listing of all commands used by Unidex III.

SECTION 5-2 COMMAND GROUPS

A. X AND Y CODES

The axis functions (X and Y codes) are those which command motion from the X and Y axes or, in connection with certain G codes, specify position related quantities. Code Xnnnnnnnnnn specifies an X axis destination (where nnnnnnnnnn \leq 2,000,000,000) and code Ynnnnnnnnnn specifies a Y axis destination.

In the absolute mode (G90), the destination represents absolute position, as recognized in the absolute position register.

In the incremental mode (G91), the destination is recognized as an increment added to the current value of the absolute position register. When discussing X and Y codes, various other codes and conditions which effect the axes' mode of travel must be mentioned. They are:

1. Each axis may move from 1 to 2,000,000,000 steps (± 1 to $\pm 2,000,000,000$).
2. Feedrates (F codes), in steps per second, apply to each axis individually. An F code must be preceded immediately by an X or Y command. If not, an F-error (feedrate error) will occur. If the X code is followed by a feedrate and the Y code is not, X will use the new feedrate while Y

will retain the previous one. If no previous one has been entered, an F-error will occur. If the Y code is followed by a feedrate, and the X is not, then Y will assume the new feedrate, while X retains the old one.

3. The axes may be moved individually or simultaneously. The following codes are for simultaneous moves:

- a. Xnnn Ynnn
- b. Xnnn Fnnn Ynnn
- c. Xnnn Ynnn Fnnn
- d. Xnnn Fnnn Ynnn Fnnn

In the above examples, if an X or a Y is followed by a feedrate, the new feedrate applies to that axis only. If no feedrate follows an X or Y, the old feedrate is retained.

4. The axes may be programmed to move consecutively. If anything other than a feedrate command is programmed between an otherwise adjacent X and Y pair, the moves will take place individually. Example:

```
X1000 F100 NO Y1000 F100
X1000 F100 * Y1000 F100
```

When using a EOB for this purpose, as in the second example, practice caution. It could interfere with a "skip to EOB" command, causing the program to skip to the wrong place.

5. Absolute mode, G90, is the option that commands the axes to travel to absolute positions. For instance, in the absolute mode, X100 means "go to position 100". Once this position is attained, further

assertions of X100 will have no effect, since the X axis is already at position 100.

Incremental mode, G91, is the option that commands the axes to travel to incremental positions. For example, X100 in the incremental mode means to "move 100 steps". Every time the X100 command is executed, the X axis will move another 100 steps.

The incremental mode is the default option. This means that every time you power-up, the G91 mode will be implemented.

6. Corner-rounding (G23) as opposed to non-corner-rounding (G24) is an option available in axis motion. G24 is the default option, asserted at power-up.

The distinction between these two codes is nonexistent for a typical non-ramped stepping motor, but is important for a DC servo system where a lag may be substantial. It is also important in ramped stepping motor drives where, again, there is a lag in system response.

In the corner-rounding mode, Unidex III considers a move to be complete when it has output all of the pulses called for in the move command.

In the non-corner-rounding mode, Unidex III waits for an "in-position" signal from the servo drive or ramped stepper system. It does not consider the move complete until this signal is received. (More details in chapter 5, section 2(G).)

7. Xnnn and Ynnn codes may be used together with certain G codes which require axes codes to form compound commands. For example, the code for G92 is required to be followed by either an X and/or Y command to

specify how the absolute position registers are to be preset. Example:

- a. G92 X1000
- b. G92 Y2000
- c. G92 X1000 Y2000

The first example command tells Unidex III to preset X axis absolute position register to 1000.

The second example command tells Unidex III to preset Y axis absolute position register to 2000.

The third example accomplishes both the presetting of X and of Y at the same time.

B. F_CODES

The feedrate functions (F codes) determine the rate of speed at which an axis will move.

Movement can be measured in steps per second or in periods of time between steps.

1. Feedrate frequency is measured in steps per second. It ranges from 1 to 150,000 steps per second. Example: F25000
2. Pulse period is measured in real time. The time can range from 6 microseconds to 1 second (1,000,000 microseconds). Example: F=40

Feedrate commands cannot stand alone. A feedrate command must be placed after an X or a Y code. An F code applies solely to the X or Y code immediately preceding it.

As mentioned earlier in this chapter, if no F code follows a given X or Y code, the previous

feedrate will apply, assuming one has been entered.

The feedrate functions include:

1. The feedrate frequency (Fnnnn) measured in steps per second. The range is from 1 to 150,000 steps per second.
2. The pulse period (F=nnnn) measured in microseconds. The range is from 6 to 1,000,000 microseconds.
3. The joystick feedrate (F-nnnn). It is the divisor for the joystick frequency when Unidex III is in the Slew mode. The dividing factors range from 2 to 65535.

If you do not enter the F-nnnn feedfactor when in the Immediate/Slew mode, a default feedfactor of 2 will apply. This would mean that the joystick input frequency would be divided by 2.

For a high degree of feedrate resolution, choose a high dividing factor.

C. M_CODES

The miscellaneous functions (M codes) have, as their name indicates, a variety of purposes. Such commands as "program stop", "return to program start" and "return from subroutine" are M functions.

Eight buffered outputs, plus a strobe, are M functions as well. An M-output may cause a motion to occur or may signal you that a certain function has been completed.

The strobe can be set in order to apprise you of the fact that an M-output function has occurred. No other M-output should be programmed

to arrive before the strobe has gone low again. To do so would mean that you would not receive a signal from the strobe that the second M-output function has been activated.

For instructions on how to adjust the strobe pulse, refer to chapter 3, section 3.

A complete listing of the M codes employed by Unidex III is as follows:

1. M0 is the code for a stop executed by the program. The stop is subject to an interrupt or abort command. The result of aborting a stop is to continue the program. You may use a G301 code which, as mentioned before, will "arm" C1 with an interrupt when it goes from high-to-low. When this occurs, the program will skip to the EOB and continue. In other words, the stop (M0) is interrupted by G301.
2. M2 is the code which indicates the operating end of the current program, where execution comes to a halt. M2 must be placed at the end of a program.
3. M30 is identical to M2.
4. M47 is the code for the "return to current program start" command. Upon receiving the M47 command, execution restarts from the beginning of the current program.
5. M99 is the code which indicates the "return from subroutine" command.
6. M-99 indicates that the eight M-outputs switch to form two BCD digits corresponding to "nn" in negative (active low) logic (nn = 00 through 99).
7. M=nnn indicates that the eight outputs switch to form the binary code for "nnn" ("nnn" = 000 through 255).

D. D_CODES

The dwell commands (D codes) are inserted when a pause or a count is needed within the program. The pause may last only a millisecond or can endure until an event has occurred numerous times. It depends on which dwell command has been used.

1. Dnnnnnnn indicates a dwell. The digits "nnnnnnn" represent the required time in milliseconds ($nnnnnnn < 4,000,000$).
2. D=nnnnnn indicates the external event counter. Unidex III's C-input connector (back of chassis) has an external clock input. This input can be any given event, and can be used as a dwell or a counter.

Unidex III counts each negative going edge (high-to-low transition) at this input, until D=nnnnnn is satisfied. The only restriction is that each negative-going edge must be at least two microseconds apart. External clock frequency < 500 KHz.

E. N_CODES

The sequence numbers (N codes) are commands associated with "line numbering" and program flow. For instance, an N code may be used to number a location at random, or to define a particular point in the program to which program flow may be directed, by the use of jumps, skips and subroutine calls. The various N codes are as follows:

1. Nnnnnnnn is a passive location number used to label a specific location. Use of these is not essential, unless you want to refer to these locations from elsewhere in the program. The "nnnnnnn" numbers are decimal digits, as in N1200.

When Nnnnnnn is encountered in the course of program flow, it is treated as a "no-op" or label.

2. N-nnnnnn is a call to a subroutine starting at line Nnnnnnn. For instance, the command N-1200 would direct program flow to a subroutine that starts on line 1200.

The end of the subroutine is designated by M99, which affects the return to the main program. After returning to the main program, the statuses of the subroutine, and not those existing within the main program before the subroutine execution, are now in effect.

There are eight levels of nesting (together with the Nnnnnnn command). Example:

N-1200 (call to N1200, lose statuses
of main program upon return)

3. Nnnnnnn is a subroutine call that preserves the following statuses: feedrate, absolute/incremental mode and corner-rounding/non-corner-rounding mode. In other words, it calls to a subroutine as does N-nnnnnn, but the statuses existing before subroutine execution, ie., those of the main program, are maintained after the return from the subroutine.

As in the previous example, eight levels of subroutine nesting are possible. Example:

N=1200 (call N1200, preserve statuses)

4. N>nnnnnn is the command given to "jump to line nnnnnn". Example:

N>1200 (unconditional jump to N1200)

F. EOB_CODE

The end-of-block (EOB) is represented by the asterisk (*) in the Unidex III language. In the Unidex III implementation of RS-274-D, blocks are of arbitrary length. You may place an EOB essentially anywhere within the program.

The EOB can be used to aid in documentation, to single-step through an operation or to control program flow. There is no required command order with respect to end-of-block.

As discussed previously, there are numerous G codes which affect conditional or abortive skips to the nearest following EOB.

When an EOB is executed by Unidex III, it is treated as a no-op. In fact, EOB is a local label used to:

1. Act as a no-op.
2. Define "blocks" for single-step-mode program execution.
3. Define the extent of a conditional or abortive skip.
4. Provide a new line when printing programs from Unidex III.

As previously discussed, the code: "X1000 F100 Y5000 F500" will cause Unidex III to make the X and Y moves simultaneously. If you wish to make them one after the other, you may break them up with a no-op, as in one of the following examples:

1. X1000 F100 * Y5000 F500
2. X1000 F100 NO Y5000 F500

If you use EOB (first example) for the purpose of separating the X move from the Y move, be careful you do not upset the control you may wish to achieve. Use the second example in cases where EOB would delimit a skip prematurely. Example:

1. G301 X1000 F100 * Y5000 F500 *
2. G301 X1000 F100 NO Y5000 F500 *

The first example would abort only the X move. If you wish the abort to be armed during both the X and the Y move, code as in the second example.

G. G_CODES

The preparatory functions (G codes) are those which prepare the system environment before any axis movement occurs. The G commands fall into either of two categories:

1. Modal
2. Non-modal

Modal commands, once asserted, stay in effect until countermanded. The modal commands are recognizable by the fact that they come in pairs. For each command there is an opposite command. In other words, each of the pair of commands cancels the other and asserts itself. An example would be G90/91, "absolute mode" and "incremental mode" respectively. Once asserted, G90 will stay in effect, making the axes move to absolute positions until G91 is asserted, cancelling G90. With G91 asserted, the axes will move incremental distances. One of each pair (in this case G91) will always be the one selected by the system to be "on" at power-up. This is the "default option".

Non-modal commands, however, are commands of a one-time nature. They are "self-cancelling", and will not occur more than once without being reasserted. An example of a non-modal command would be G10, where position read-outs and drives are commanded to reset. Once this command has been executed, G10 cancels itself and a reset will not recur until G10 has been asserted again.

There are not many modal G codes. They are listed below, together with their default options:

1. G23 Corner-rounding mode
2. G24 Non-corner-rounding mode (default)
3. G90 Absolute mode
4. G91 Incremental mode (default)

NOTE: Feedrates also are considered modal, in that one feedrate will stay in effect until another is entered to take its place.

The distinction between the corner-rounding and non-corner-rounding mode is non-existent when Unidex III is connected to a lowspeed stepping motor system; whichever code you may assert, the result will be the same. Not so, however, when a DC servo system is connected to Unidex III.

The DC servo drive (motor, servo amplifier and serial load system) will lag in its response to the clock pulses which Unidex III outputs. Because of this, when Unidex III has completed its output of pulses for a given X, Y move, the drive system will not simultaneously have come to rest at the designated coordinates.

Depending on the speed of the move and upon how the drive is tuned (the bigger the load the greater the lag), several tens of milliseconds may lapse between the conclusion of the pulse output from Unidex III and the system's

attainment of its final destination. To accommodate this effect, virtually all DC servo drives provide an output, typically called "in position", that signals when the drive has "run down" all of the clock pulses that have (thus far) been the input, and signals also that the system has come to rest.

Unidex III is provided with an input which accepts the "in position" output of the servo drive. In the non-corner-rounding mode (G24) this "in position" input supersedes the input of the Unidex III counter. Therefore, the Y move won't begin until the X move has been completed and the drive has come to rest. The result is, of course, non-rounded corners.

As the name implies, "corner-rounding" will produce rounded corners if, for example, you are scribing a rectangle. In the case of the corner-rounding mode (G23), Unidex III does not wait for the "in-position" input, but proceeds to execute subsequent commands as soon as it has completed outputting the clocks for a given move. For certain speeds this effect may be useful and, hence, has been incorporated into Unidex III.

Whether you use this for corner-rounding or not, you will want to keep G23/G24 in mind when you have coded for a move, interrupt or abort. The two codes, G23/G24, operate the same way under an aborted move condition as under G24. Under G24, Unidex III "carries on" immediately (upon the abort).

G90 and G91 are almost self-explanatory. In absolute mode (G90), a move command will take you to the spot that you have designated by the command. In incremental mode (G91), Unidex III will interpret the move command as an increment of your current position. These commands may be asserted arbitrarily within your program.

The following are the non-modal codes:

G7 Go home (both axes)
G10 Reset drives

```

G11  Reset X drive
G12  Reset Y drive
G25  Interrupt the host computer
G60  X axis only go home
G61  Y axis only go home

```

Only slightly more complex are the G codes that enable conditional skips to be made. These codes make use of the end-of-block (EOB) character "*". Described more completely in the previous subsection (EOB Code), the EOB character is a marker which can be used to break up a program into segments, or "blocks". Typically, one will want to segment a portion of a program that is to be executed conditionally. For example, consider the following program:

```

C1 true
|
|   ▼
"G7 G271 X1000 F200 Y5000 F1000 * M-58 D1000 M47"
|
|   ▲
|   _ _ _ _ _ C1 false _ _ _ _ _
|

```

The code G271 (following G7) calls for a test of the C1 input (Condition 1 input). If, on test, C1 is logic true (positive voltage at C1 input), the code "X100 F200 Y5000 F1000" will be executed. If, on test, C1 is logic false (ground at C1 input) execution will skip to the EOB character (*) and proceed from there. In either case, "M-58 D100 M47" will be executed.

The EOB character serves as a delimiter for the G271 command. When, on test, C1 is logic false, all code, up to the first EOB encountered after the G271, will be skipped. In other words, G271 is a skip to EOB when C1 is false. G272 through G274 provide similar skips for condition inputs C2 through C4. Codes G281 through G284 provide skips to EOB when C1 through C4, respectively, are logic true (positive voltage).

Codes G301 through G304 provide for interrupt-type skip conditions. The code G301,

for example, "arms" an interrupt to take place when input C1 makes a logic high-to-low transition. If such a transition occurs, a flag (flag 1) is set, and execution skips to EOB. When EOB is encountered, the interrupt is disarmed. The flag is left set however, until you clear it or until another G301 is asserted. C1 through C4 interrupts are recognized only during programmed stop (M0), dwell or move commands.

The idea behind the G301 interrupt is to allow you to abort a stop, a move or a dwell. These are commands which take "real time" to execute and which you may want to be able to "abort" on an immediate basis. For example, consider M0, the "halt" or "stop" command. If we program "G301 M0 *" when G301 is executed, an interrupt on C1 is armed; after G301 is executed, M0 is executed. M0 brings further execution to a halt. Nothing happens until C1 input makes a transition from true to false; M0 is then aborted and execution proceeds from EOB.

We could also have programmed "G301 D10000 * G531 N>500 *". In this case, a ten second dwell will execute until aborted or completed, whichever comes first. The code "G531 N>500" shows how you can test a flag associated with the C1 interrupt and make a jump according to whether the interrupt occurred or not. The EOB followed by G531 is necessary to update position registers correctly and to accurately control the program flow.

The most interesting and useful application of the G301 interrupt is its capacity to abort a move in progress. Consider the previous move example, with G271 replaced by G301:

```
"G7 G301 X1000 F200 Y5000 F1000 * G531 M-58 D100 M47"
      |
      | ▲
      | _C1 goes T to F_!
```

The code G301 arms C1 for a high-to-low interrupt. After executing G301, Unidex III

executes the move "X1000 F200 Y5000 F1000". Now suppose C1 goes from high-to-low before the move is complete. If this occurs, Unidex III immediately stops putting out clock pulses. This is done in hardware. At the most, one system clock cycle (1 microsecond) will elapse between the transition from high-to-low on C1 and the cessation of clock pulses. Execution of code now jumps to end-of-block.

In G23, as well as in G24, the G531 execution occurs "instantly", as quickly as the firmware can accomplish it. In effect, Unidex III does not wait until "in position" is output by the drive, but moves on. Naturally, these considerations are of consequence only if you are using a DC drive system.

In general, if a drive has a lag of T seconds and it is up to speed, receiving N counts per second, the position lag will be NT counts. This may or may not be too small to consider. There is a way to cut the lag out of the picture, however. Consider this code:

```
"G7  G301 X1000 F200 Y5000 F1000 * G531 G10 M-58 D100 M47"
      |
      |   ^
      |   |
      |   | C1 goes T to F _ _!
```

Here is what happens. Unidex III executes G531 immediately, which resets the drives and brings the system to a halt. In the above example you will lose, at the most, 500 microseconds from the time C1 goes high-to-low until G10 is asserted. This lost time is a function of how much code appears between G301 and EOB. The above amount of code is considered small because there is no additional code between the "move" code and EOB.

It is certainly legal to have additional code following the move. Here are two examples:

```
"G301 X1000 F200 Y5000 F1000 M-26 * G531 M-58"
```

"G303 X1000 F200 Y5000 F1000 * G533 N>150 *

How these will execute is determined by the rule that only stops, dwells and moves are interrupted. If an interrupt occurs during a "non-move", it will be retained by the system until:

1. The next stop, dwell or move occurs
2. The next EOB disarms it
3. The next arm instruction clears and resets it

Consider the first example. If the interrupt occurs during the move, the move will be aborted; M-26 will be skipped and execution will resume with M-58. On the other hand, suppose the move completes and then the interrupt occurs. Even if, in time, the interrupt occurs before the execution on M-26 gets started, M-26 will be executed anyhow. Since the interrupt did not occur during the move, the interrupt flag will not be set. Interrupt flags only record aborted moves, not merely the occurrence of the interrupt.

Consider the second example. N>150 will be executed if the interrupt occurred during the move.

If and when the interrupt occurs, the skip will be made to the first EOB that occurs during running program flow. The reason for this, the guiding principle, is simple: an interrupt cannot create a flow path, it can only bypass an existing flow path. If a jump, condition test, subroutine call or return to start command is programmed before the EOB and flag test, erroneous program flow could result in unexpected program operation.

The codes associated with the C1 input, when it is used for interrupt purposes, are as follows:

-
- G301 Arms interrupt (abort) on C1 input for true-to-false transition. If this transition occurs before the next EOB is executed, execution will skip to that EOB. Only moves, halts and dwells are aborted; if an abort is made, flag 1 is set.
- G311 Same as G301, except for the fact that a false-to-true transition of C1 is recognized.
- G501 Clear flag 1 (interrupt flag for input C1).
- G511 Set flag 1.
- G521 Test flag 1 and continue if set. If clear, skip to EOB.
- G531 Test flag 1 and continue if clear. If set, skip to EOB.

The codes G302, G303 and G304 are like G301, but they arm interrupts on C2 through C4. Similarly, the codes for setting, clearing and testing flag 1 listed above have their counterparts for flags 2 through 4. See the list of codes at the end of this chapter.

As a final example, the code "G301 X2000 F250 Y1000 F125 M-52 *" arms C1 for an interrupt and proceeds to make a move. If the interrupt on C1 (C1 goes from high-to-low) occurs before the move is complete, the move is aborted and execution of code skips to EOB (the M-52 code is not executed). On the other hand, if the move runs out before the interrupt occurs, M-52 will be executed and the EOB will disarm the interrupt.

To summarize, G301 arms the interrupt, the next EOB disarms it. The interrupt occurs on a high-to-low transition of C1 only if this transition takes place during a move, stop or dwell. When the interrupt does occur, the move,

stop or dwell is aborted, flag 1 is set and a skip is made to EOB.

The other G codes of this order, ie., codes which are interrupts, set flags, clear flags and test flags, are as follows:

G312-G314 Arm interrupts on C2 through C4 inputs for false-to-true transitions, just as G311 does on C1.

G502-G508 Clear flag 2 through flag 8.

NOTE: Although flag 1 through flag 8 are all user programmable, only flag 1 through flag 4 are common to interrupt inputs C1 - C4.

G512-G518 Set flag 2 through flag 8, just as G511 sets flag 1.

G522-G528 Test flag 2 through flag 8 (continue if set, skip to EOB if clear), just as G521 tests flag 1.

G532-G538 Test flag 2 through flag 8, just as G531 tests flag 1.

The remaining G commands are as follows:

G60 X axis only go home (Y remains in position).

G60=nnnnnn Loads X axis go home feedrate (1 ≤ nnnnnn ≤ 150,000).

G61 Y axis only go home (X remains in position).

G61=nnnnnn Loads Y axis go home feedrate.

G661=nnnnnn Loads repeat counter 1. This command is used in conjunction with G671 (decrement/skip if zero

repeat counter). It loads the repeat counter with "nnnnn" (nnnnn <= 65535). Together with G671, it allows repeat loops to be made. Example:

```
"G7 X1000 F200 Y1000 F200 G661=50
N100 X10 F10 Y10 F10 G671 N>100 *
```

In this program, the axes go home and then go to X=1000 and Y=1000 position. The repeat counter is loaded with the number 50. N100 labels the line X10 F10 Y10 F10 with the number 100 (as a reference point). G671 decrements the repeat counter by 1. So for 50 counts, line 100 will be repeated. When the decrement command brings the repeat counter to zero, the program will go to EOB.

G662=nnnnn
to
G668=nnnnn

Load repeat counters 2 through 8. These are associated with the "decrement repeat counter" codes G672 - G678.

G671

Decrement/skip if zero repeat counter 1. If it is zero, skip to end-of-block. This is associated with the repeat counter, G661=nnnnn.

G672
to
G678

Decrement repeat counter 2 through counter 8.

G90

Absolute mode

G91

Incremental mode.

G92

Preset X position register (XPSN) and Y position register (YPSN). These registers may be preset in

order to establish a home reference point at a place other than absolute home. Both of these registers are cleared when the axis is sent home (G7). G92 is a compound command, ie., it does not stand alone, but must be followed by commands saying where the position registers must be set. Thus, "G92 X1000 Y200", for example, presets the X and Y absolute position registers to 1000 and 200 respectively. The absolute position registers, XPSN and YPSN, always keep track of the true position of the device you are controlling unless you deliberately change them. Example:

G7 Go home (clears them)

G92 Xnnnn (presets them)

The true theoretical position of the device and the actual position agree after a move has been executed, if you are in non-corner-rounding mode.

If in corner-rounding mode, the actual position will lag, at least in a DC drive system.

SECTION 5-3 INTERFACING UNIDEX III

As mentioned throughout this manual, the four ways to communicate with Unidex III are:

1. Locally, via the front panel keyboard and display.
2. Via IEEE-488 interface bus

-
3. Via RS-232C serial line interface
 4. Via RS-422A differential serial line interface

A. LOCAL_INTERFACE

In some cases, it may be more efficient and economical to use Unidex III without a host. Unidex III possesses edit capabilities that allow you to enter and edit short programs efficiently. There is, additionally, a PRINT key that allows you, after connecting Unidex III to a printer, to print out programs from the user memory.

B. IEEE-488_INTERFACE

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

C. RS-232C_INTERFACE

RS-232C has 2 data lines and 20 additional lines for control. The control lines implemented in Unidex III are:

1. RTS (request to send, CA)
2. CTS (clear to send, CB)
3. DSR (data set ready, CC)
4. DTR (data terminal ready, CD)
5. DCD (data carrier detect)

Since RS-232 has only 2 data lines, data flows in a serial fashion, that is, one bit at a

time. RS-232 is a widely used form of communication between various devices.

D. RS-422A INTERFACE

RS-422 implements the same serial line communication protocol as RS-232 with differential line drivers.

Unidex III has connectors on the rear panel of the chassis for any of the above mentioned communication interfaces, ie., the IEEE-488, RS-232C and RS-422A.

If a host computer is used with Unidex III, as mentioned in sections B, C, and D, there are three methods of preparing and downloading a program that can be employed. They are as follows:

1. The program is prepared on the host and then loaded into the Unidex III memory for execution either immediately or at a later time. The host and Unidex III need not be connected during operation.
2. The program is prepared on the host and is then loaded into the Unidex III memory and executed. Unidex III and the host computer communicate throughout the procedure via the use of service requests from Unidex III and other commands from the computer.
3. The program is prepared on the host and is then loaded into Unidex III one block at a time. Unidex III executes on a block-by-block basis, apprising the host when each of these is completed. The host can then communicate to Unidex III which operation to execute next. This is the Immediate mode.

SECTION 5-4 LOCAL_SAMPLE_PROGRAMS

When you power-up Unidex III and the message "SYS RDY" (system ready) is displayed, the choice of going into local or remote is now available to you. As mentioned earlier, at this point Unidex III is in local with remote enabled. The source of the next entry lets Unidex III know which it's to be.

At this point, if you operate any key on the front panel other than REMOTE, Unidex III will go into the local mode. If you should wish to switch back to the remote enabled state, simply press the REMOTE key. When the next entry comes from the host, Unidex III will be in the remote mode.

A. EDIT_MODE

The Edit mode, as discussed previously, is the only mode in which you may write, edit or change a program. It offers these features:

1. Edit

Edit mode asks for the program number and calls up the start of the program requested.

2. Step_and_Backstep

Step and Backstep allow you to step through your program. The program "loops around", so that if you step or backstep through a program, you will eventually come back to where you started.

3. Search

Search allows you to search for a command within your program.

4. SLEW

Takes you back to the start of the program.

5. Clear_Entry

Clear Entry (CE) allows you to clear one entry. If you wanted to clear X10, hitting CE once would clear the zero only.

6. Clear_Command

Clear Command (CC) allows you to clear one command. If you wanted to clear X10, hitting CC once would clear the whole command.

7. Edit_Sample_Programsa. Edit_Function

POWER UP	SYS__RDY
<input type="text" value="EDIT"/>	PGM__<1>
<input type="text" value="2"/>	EDII__2
<input type="text" value="5"/>	EDII__25
<input type="text" value="EXEC"/>	____>>6Z

You will see the first command of program 25. If program 25 is not yet in memory, but a new program you wish to enter, you will see only the prompts upon execution. Example:

<input type="text" value="EXEC"/>	____>>
-----------------------------------	--------

When you see the prompts, you may start to enter the new commands.

b. Step/Backstep_Function

In order to edit a program employing the edit functions, you would proceed as follows:

EDIT	EDIT__25
EXEC	____>>G7
STEP	>>G7X100
STEP	X100F100

You can step through the whole program. When you come to the end, you will encounter the beginning again:

STEP	____>>G7
------	----------

Backstepping will accomplish the same thing, only in a backward direction.

c. Search_Function

If you need to search for a command, in order to change it for instance, proceed as follows:

SHFT	SRCH	_____
	Y200	____Y200
	EXEC	_LOOKING

OF20Y200

If the command being sought is not in the program, the message "WHAT?" will be displayed.
Example:

EXEC	_WHAT?_
------	---------

B. AUTO_MODE

As you know, the Auto mode is used when you want to have the program executed, beginning to end, without user intervention. You have only to press AUTO, give the desired program number and then hit EXECUTE. If for any reason you must stop the action of the drives, hit the RESET key.

1. Auto_Sample_Program

POWER UP	SYS__RDY
AUTO	PGM__<1>
5	RUN____5
7	RUN____57
EXEC	_RUNNING (runs program)
	__READY__

After Unix III has completed the program it will display "READY" on the screen. If you want to execute the same program again, simply press EXECUTE. If a new program is needed, you must press AUTO again and enter the new number.

The default program, program 1, is the one evoked upon power up. If you do not enter another number, program 1 will run.

If you have already been working on a program, however, that is the program number that will be displayed when you press a mode key.

C. SINGLE_MODE

In the Single mode, the program is executed block-by-block. Each time the EXECUTE key is pressed, the next block is executed. Unlike the Immediate mode, Single mode is a way of stepping through and executing, one block at a time, a prewritten and stored program.

1. Single_Sample_Program

<u>SNGL</u>	<u>WALK__57</u>	
<u>2</u>	<u>WALK__72</u>	
<u>5</u>	<u>WALK__25</u>	
<u>EXEC</u>	<u>_RUNNING</u>	(executes one block)
	<u>__READY__</u>	
<u>EXEC</u>	<u>_RUNNING</u>	(executes next block)

The Single mode will allow you to continue in this fashion until the end of the program is reached (M2 or M30) when "PGM END" (program end) will be displayed. For another program, you must again press SINGLE and give the new program number.

D. IMMEDIATE_MODE

In the Immediate mode, each string of commands is entered and executed immediately. No program is prewritten and stored in memory. You can, however, enter a command, execute it and then repeat the command string by simply pressing

EXECUTE again. To enter further commands, just type them in; feedrate and X/Y positions are retained.

Pressing IMMEDIATE will not cause the loss of the former commands. It allows you to edit the previous string of commands by pressing BSTP first. Pressing EXECUTE causes the entire thing to run again. It can be seen from this that the Immediate mode program is kept in a very temporary "memory" of sorts. Entering a new character without pressing BSTP overwrites the Immediate memory.

If any other mode key is pressed, ie., EDIT, AUTO or SINGLE, the Immediate program is lost. This will also occur, of course, if the RESET key is pressed.

1. Immediate_Sample_Program

POWER	UP	SYS__RDY
	IMMD	ENTER_>>
SHFT	G	_____>>G
SHFT	7	_____>>G7
.	.	
.	.	
	EXEC	_RUNNING
		__READY__
	G	_____G
	9	_____G9
	0	_____G90
.	.	
.	.	

EXEC	__RUNNING
	__READY__
AUTO	RUN___25

As can be seen, whether or not you press IMMEDIATE, you may enter further commands. However, once another mode key is pressed, Unidex III exits Immediate and loses the program.

2. Manual Mode

In the manual mode, you can move the two axes by three methods:

1. Step
2. Slew using keyboard
3. Slew using joystick

With Step, you can move the drives one step at a time, pressing one of the arrow keys each time a step in a given direction is desired.

With Slew, rapid manual positioning of the drives is possible by holding down one of the arrow keys. Movement will stop when the key is released. The speed of movement in the Slew mode can be determined by the feedrate.

With Slew using the joystick, the joystick instead of the arrow keys is used for positioning.

a. Step

In Step, no feedrate is necessary, since the Step mode dictates its own rate of speed. The drives will travel

one step each time an arrow key is pressed. Example:

<div style="border: 1px solid black; padding: 2px;">IMMD</div>	ENTER_>>
<div style="border: 1px solid black; padding: 2px;">STEP</div>	(when arrow key is pressed, axis moves one step)

b. Slew_Using_Keyboard

For Slew a feedrate is necessary, since Unidex III needs to know at what rate of speed to move once an arrow key is pressed. Example:

<div style="border: 1px solid black; padding: 2px;">IMMD</div>	ENTER_>>
<div style="border: 1px solid black; padding: 2px;">F</div>	_____F
<div style="border: 1px solid black; padding: 2px;">2</div>	_____F2
<div style="border: 1px solid black; padding: 2px;">0</div>	_____F20
<div style="border: 1px solid black; padding: 2px;">SLEW</div>	(when arrow key is pressed movement will occur at 20 steps/second until arrow key is released)

If you neglect to enter a feedrate, the default feedrate is 400 steps per second after power up, or the previously entered slew feedrate.

c. Slew_Using_Joystick

If the joystick is used, a feedfactor of from 2 to 65535 may be entered. This number becomes the factor by which the joystick frequency is divided. If none is entered, F-2 becomes the feedfactor by default, once you begin to operate the joystick. The feedrate is the joystick frequency divided by the factor.

IMMD	ENTER_>>
F	-----E
SHFT -	-----E-
8	-----E-8
SLEW	(when joystick is operated, movement occurs)

The joystick will move at a rate that is equal to its own frequency divided by 8 in the above example.

E. PRINT-OUTS OF PROGRAMS

In order to get a print-out of any or all of your programs, or to get a print-out of your directory (list of programs on file), you must connect Unidex III to a printer. This connection must be made at the RS-232 DTE or RS-422 DTE port, depending on the interface you are using.

If you should need a hard copy of one of your programs, simply type in the appropriate number. Upon execution, your printer will respond. Example:

PRINT	PRNT_<1>
2	PRNT___2
5	PRNT__25
EXEC	PRINTING

If a complete print-out of all of your programs is required, type in zero-zero for the program number. Example:

PRINT	PRNT__25
0	PRNT___0
0	PRNT__00
EXEC	PRINTING

If a print-out of the directory is needed, you would enter D for the program number. Example:

PRINT	PRNT_<1>
D	PRNT_DIR
EXEC	PRINTING

F. CLEARING PROGRAMS

Eventually, you may wish to erase a program from Unidex III's memory. You may possibly wish to erase the entire memory.

To erase one program, simply type in the program number, after entering the Edit Mode and pressing CLEAR COMMAND key. Example:

EDIT	PGM__<1>
SHFT CC	CLR____1
0	CLR____0
5	CLR___05
EXEC	_CLEARED

For clearing the whole memory, you would enter zero-zero instead of any particular program number. Example:

	EDIT	PGM__<1>
SHFT	CC	CLR____1
	0	CLR____0
	0	CLR___00
	EXEC	_Cleared

SECTION 5-5 REMOTE_SAMPLE_PROGRAMMING

This section will concentrate on examples of remote programming. The discussion of how the various interfaces (IEEE-488, RS-232C and RS-422A) transmit data and control signals was covered in chapter 4, section 4. How your host implements communication with Unidex III depends upon the programming language your host employs. This section only deals with the commands Unidex III needs to operate.

A. ADDRESSING_UNIDEX_III

1. IEEE-488_INTERFACE

Addressing of devices connected to the IEEE-488 interface bus is a primary function of the bus and is specified in the bus specifications. Different computers implementing the interface effect the various bus functions, using different commands or statements and languages. The user is requested to refer to the specific computer manual.

2. RS-232C/422A

The serial line daisy chain protocol is unique to Unidex III and a specific character sequence explained below is required to address Unidex IIIs on the line.

The device address is set using the switch SW3 on the rear panel as shown in table 4-4.1. The sequence:

<ESC> nn, mm, ll <CR LF>

addresses the devices nn, mm and ll to listen. The ESCAPE code <ESC> unlistens all Unidex IIIs on the line and restarts the addressing operation. A sequence of two digit numbers separated by a space, comma or semicolon, represent the addresses of the devices to be made listeners. <CR LF> ends the addressing sequence.

It is also important that the last device switch be set correctly to ensure proper communication.

B. EDIT_MODE

When you power-up the system, Unidex III is in the local with remote enabled mode. When the first entry it receives is from the host, Unidex III will go into the remote mode. To go back to local, transmit an "L" over the lines.

Once in remote, and the listener addressed, transmit an "E" to enter the Edit mode. Next send the number of the program you are about to enter, followed by a carriage return/line feed (CR LF). You may now send the program's motion commands over the lines. When the program is complete, end it with an end-of-program (M2) or an end-of-tape (M30) command.

When using the RS232/422 serial interface, to do a new address sequence, the currently addressed devices must be taken out of the Edit mode. This is done by sending on the line, "R CRLF"

The Edit functions of Unidex III are not available from remote. To change your program,

it must be downloaded into Unidex III where it can be edited locally.

1. Program Entered On IEEE-488 BUS

When using IEEE-488, address any and all Unidex III's according to your system. In the following examples, we will assume that there are five Unidex IIIs on the bus. Address each Unidex III to which you wish to send a particular program.

For instance, assume you wish to send program 99 to four of your Unidex IIIs. Example:

Address Unidex IIIs at addresses 10, 12, 13 and 14, for instance:

"E99 CR LF"

"G7 G91 X1000 F10 Y1000"

"M30 CR LF"

Program 99 is now stored in the Unidex III at each of four locations. The unaddressed Unidex III (remember, there were five on the bus) does not have program 99 in memory.

2. Program Entered Over RS-232/422 Line

When using RS-232/422 interface, the ESCAPE key (ESC) must be pressed before any and all Unidex IIIs are addressed. As you recall, RS-232/422 needs certain control commands which IEEE-488 handles internally.

Again, let us assume there are five Unidex IIIs on the line, and you wish to send program 99 to four of them. The five Unidex IIIs are at addresses 10, 11, 12, 13 and 14. Example:

```

"ESC 10, 12, 13, 14 CR LF" (address-
                           ing sequence)
"E99 CR LF"
"G7 G91 X1000 F100 Y1000 F10 . . . ."
"M2 CR LF"
"R CR LF" (come out of Edit mode)
"ESC 11 CR LF" (address device 11)
"E 52 CR LF"
"G7 . . . M2 R CR LF"

```

Program 99 is now stored in the Unidex III at each of four locations. The fifth Unidex III has program 52 in memory.

C. AUTO_MODE

In order to have a program execute in the remote Auto mode, transmit an "A", followed by the number of the program to be executed and CR LF. After the program has been executed, Unidex III will send an SRQ (service request).

1. Program Entered On IEEE-488

Address Unidex III(s);

```
"A99 CR LF"
```

Serial poll Unidex III(s). (Every serial poll gives status of one Unidex III - lowest address to highest.)

2. Program Entered Over RS-232/422

```
"ESC 12, 13 CR LF" (address devices
                    12 and 13)
```

```

"A99 CR LF" (program is executed by
              the Unidex IIIs)
"Q CR LF" (serial poll Unidex III at 12)
"Q CR LF" (serial poll Unidex III at 13)
"ESC 11 CR LF" (address device 11)
"A 52 CR LF" (run program 52)

```

In the above example, you must do a query (serial poll) for as many Unidex IIIs as were addressed. They will answer from the lowest address to the highest.

D. SINGLE_MODE

In order to have a program execute in the remote Single mode, transmit an "S", followed by the number of the program to be executed and CR LF. After each block is executed, Unidex III will send an SRQ.

1. Program Entered On IEEE-488

```
Address Unidex III(s);
```

```
"S25 CR LF"
```

Serial poll Unidex III(s) after each SRQ is sent (which is after each block execution).

2. Program Entered Over RS-232/422

```
"ESC 11 CR LF"
```

```
"S25 CR LF" (execute first block)
```

```
"Q CR LF" (serial poll after each
            block execution)
```

"CR LF" (execute next block)

"Q CR LF"

E. IMMEDIATE_MODE

In order to execute commands in the remote Immediate mode, transmit an "I".

You may now enter each command string followed by a CR LF. After execution of the block is complete, Unidex III will interrupt the host (SRQ) and wait for a serial poll. Using the RS-232/422 interface to address Unidex III to run a second Immediate command, requires the first addressed device to be taken out of Immediate mode before doing a new <ESC> sequence. This is accomplished by sending the character "R" on the line.

As mentioned before, the Immediate mode program is not a prewritten or stored program.

1. Program_Entered_On_IEEE-488

Address Unidex III(s);

"I"

"G7 X2000 F200"

"CR LF" (block executes)

Serial poll Unidex III(s) after each
SRQ is sent - after each block execution.

2. Program_Entered_Over_RS-232/422

"ESC 10, 12 CR LF" (address devices 10
and 12)

"I G7 X2000 F200"

"CR LF"

"Q CR LF" (serial poll of Unidex III
at address 10)

"Q CR LF" (serial poll of Unidex III
at address 12)

"R CR LF" (takes Unidex IIIs out of
Immediate mode)

"ESC 13, 14 CR LF" (address devices 13
and 14)

F. PRINT

When the ASCII character "P" (for print) is sent over the lines to Unidex III, Unidex III responds by sending the requested program, programs, positions or statuses to the host.

1. IEEE-488_Bus

a. Print_a_Program:

Address Unidex III(s);

"P25 CR LF"

If the requested program is not in memory, Unidex III will do a Service Request.

b. Print_a_Status:

Address Unidex III(s);

"PS CR LF"

Every time "PS CR LF" is sent, the status of the Unidex III at the lowest address requested will be sent to the host.

2. RS-232/422 Interface

a. Print a Position:

"ESC 14 CR LF"

"PX CR LF"

At this point, the X axis position of the Unidex III at address 14 will be sent to the host. The format of the data returned is as shown below.

<- sign or space> nnnnnnnnnn CR LF

When using RS232/422 let <ETX> follow CR LF.

G. ERASE

When it is necessary to erase a program or programs, send an "E" (for edit) and a dollar sign (for erase) followed by the program number(s) and CR LF.

1. IEEE-488 Bus

Address Unidex III(s);

"E \$ 25 CR LF"

2. RS-232/422 Interface

"ESC 10, 12, 13 CR LF"

"E25 \$ CR LF"

H. RS-232/422_INTERFACE_CODES

Some ASCII codes have been assigned for control when using RS-232/422 interface. There is a complete list in table 4-3.1 in appendix 1.

Some of these commands are programmed as follows:

1. C_-_Clear_Device

"ESC 13, 15 CR LF"

"C CR LF"

Unidex III(s) at 13 and 15 are cleared. If you wish to clear all devices, you may just program CLEAR this way:

"ESC C CR LF"

All devices will be cleared. You may neglect listing addresses if the command to be sent is a "universal command", ie., one that may be sent immediately after <ESC> without specific addresses listed. Universal commands apply to all devices. Other universal commands will be listed within this section.

Once all Unidex IIIs are cleared, any future listener must be reconfigured.

2. I_-_Trigger

"ESC 10, 13, 15 CR LF"

"T CR LF"

The Unidex IIIs at 10, 13, and 15 will now execute their programs simultaneously. You may also program:

"ESC T CR LF" (universal)

3. L--Go To Local

"ESC L CR LF" (universal)

or "ESC 10, 13 CR LF L CR LF"
(addressed)

4. O--Cancels H

"ESC O CR LF" (universal)

"ESC 10, 13, 15 CR LF O CR LF"

5. W--Configures SRQ

"ESC 12 CR LF W % CR LF"

Unidex III at address 12 will now send a % for SRQ.

6. Z--Cancels W

"ESC 12 CR LF Z CR LF"

7. J--Configures Series RTS

"ESC J CR LF" (universal)

8. K_-_Configures_parallel_RTS

"ESC K CR LF" (universal)

NOTE: J and K have to be universal, since all of the Unidex IIIs on the line must be configured as either series or parallel. You can't mix the two types of RTS.

9. Q_-_Query (serial poll)

"ESC 13, 14 CR LF"

"Q CR LF" (poll Unidex III at 13)

"Q CR LF" (poll Unidex III at 14)

"Q CR LF" (poll Unidex III at 13
again)

SECTION 5-6 RS-274-D COMMAND SUMMARY

USER ACCESSIBLE REGISTERS

XPSN Relative X position. Position of X axis in number of steps relative to the current home reference point. Home reference point equals home limit marker plus home offset. This register is cleared when the axis is sent home (G7), and an X-home-offset has been made. In data transfer commands, "X" refers to this register.

YPSN Relative Y position. Same as above, for Y.

X AND Y CODES

X and Y refer to the X axis and Y axis respectively. The following command syntaxes for X are also valid for Y.

Xnnnnnnnnnn Specifies an X axis move in Incremental mode. Specifies the X axis destination in Absolute mode (nnnnnnnnnn ≤ 1,999,999,999).

N CODES

. Nnnnnnn Line number used as label. Not essential unless you want to refer to the line from elsewhere in the program. The letters "nnnnnn" represent decimal digits. Example: N5500

N-nnnnnnn Call to subroutine starting at line number nnnnnn. M99 effects return from subroutine. Status before subroutine is not preserved.

N=nnnnnn Subroutine call that preserves the following statuses: feedrate, absolute/incremental, corner-rounding/non-corner-rounding. M99 effects return. Eight levels of nesting, including the command above. Example: N=5500

N>nnnnnn Jump to line number Nnnnnnn. Example: N>5500

M_CODES

M0 Programmed stop. The program executes a stop. The stop may be interrupted and aborted as any other move. The result of aborting a stop is to continue. (See G codes for commands to interrupt.) In local, display "M0 - stop".

M2 Operating end of current program. Must place this at the end of each program. You can, however, use M30 in its place. Halts execution of program.

M47 Return to current program start. Execution restarts from the beginning of current program.

M99 Return from subroutine.

M-99 M function command. The eight \bar{M} outputs switch to form two BCD digits corresponding to "nn" in negative logic (nn = 00 through 99).

M=nnn M function command. The eight \bar{M} outputs switch to form the binary code for "nnn" (nnn = 000 through 255).

E_CODES

Fnnnnnn Feedrate frequency ($1 \leq \text{nnnnnn} \leq 150,000$)
in steps per second.

F=nnnnnn Pulse period in micro-seconds ($6 \leq \text{nnnnnn} \leq 1,000,000$).

F-nnnnn Divisor for joystick input frequency in the
SLEW mode ($2 \leq \text{nnnnn} \leq 65535$). F-2 is
as low a divisor as can be entered.

D_CODES

Dnnnnnnn Dwell. The digits represent the required
time in milliseconds ($\text{nnnnnnn} < 4,000,000$).

D=nnnnnn External event counter. The digits represent
the required count in external events (input
located on C-input connector, back of
chassis).

G_CODES

G7 Go home - both axes.

G10 Reset both drives.

G11 Reset X drive.

G12 Reset Y drive.

G23 Corner-rounding mode.

-
- G24 Non-corner-rounding mode.
- G25 Interrupt host computer, if in remote. Display "G25 - stop" if in local. The Service Request (SRQ) line asserted when the IEEE-488 interface is active. When the RS-232C/RS-422A interface is active, a programmable ASCII character (section 4-2B) which is configured by sending code W, is sent to the host computer or RTS line is turned on. For example, by sending W %, the character % would be configured as the SRQ. In local, EXEC may be pressed twice in succession to continue program. In remote, send "B [CR][LF]" on line to continue.
- G271 If input C1 is high, continue; if C1 is low, skip to the end-of-block. Example:
- "G7 X100 F20 G271 N-800 N>900 *"
- If C1 is high: the program will execute the subroutine, starting on line 800, and after returning from the subroutine will jump to the line numbered 900. If C1 is low: the above routine will be skipped over and program will go directly to EOB.
- G272 Same as above, but for C2.
- G273 Same as above, but for C3.
- G274 Same as above, but for C4.
- G281 If input C1 is low, continue; if high, skip to the end-of-block.
- G282 Same as above, but for input C2.
- G283 Same as above, but for input C3.

-
- G284 Same as above, but for input C4.
- G301 Interrupt move, set flag-1 and skip to the end-of-block for a negative edge (1 to 0 transition) on C1. The position at the time of interrupt is not lost. The XPSN and YPSN are updated upon interrupt, based on the number of pulses output on each axis.
- G302 Same as above, but for C2. Sets flag-2.
- G303 Same as above, but for C3. Sets flag-3.
- G304 Same as above, but for C4. Sets flag-4.
- G311 Same as above in G301 code, only for the positive edge (0 to 1 transition) for C1.
- G312 Same as above, but for C2.
- G313 Same as above, but for C3.
- G314 Same as above, but for C4.
- G501 Clear flag-1 (set by interrupt input C1 following command G301 or G31).
- G502 Clear flag-2 (set by interrupt input C2).
- G503 Clear flag-3 (set by interrupt input C3).
- G504 Clear flag-4 (set by interrupt input C4).
- G505 Clear flag-5 (user programmable - but not associated with input conditions).

G506	Clear flag-6 (user programmable).	
G507	Clear flag-7 (user programmable).	
G508	Clear flag-8 (user programmable).	
G511	Set flag-1.	
G512	Set flag-2.	
G513	Set flag-3.	
G514	Set flag-4.	
G515	Set flag-5 (user programmable).	
G516	Set flag-6 (user programmable).	
G517	Set flag-7 (user programmable).	
G518	Set flag-8 (user programmable).	
G521	Test flag-1 and continue if set. clear, skip to end-of-block.	If
G522	Same as above, but for flag-2.	
G523	Same as above, but for flag-3.	
G524	Same as above, but for flag-4.	
G525	Same as above, but for flag-5.	

G526 Same as above, but for flag-6.

G527 Same as above, but for flag-7.

G528 Same as above, but for flag-8.

G531 Test flag-1 and continue if clear. If
 set, skip to the end-of-block.

G532 Same as above, but for flag-2.

G533 Same as above, but for flag-3.

G534 Same as above, but for flag-4.

G535 Same as above, but for flag-5.

G536 Same as above, but for flag-6.

G537 Same as above, but for flag-7.

G538 Same as above, but for flag-8.

G60 X axis only go home.

G60=nnnnnn Load X axis go home feedrate
 (1 <= nnnnnn <= 150000).

G61 Y axis only go home.

G61=nnnnnn Load Y axis go home feedrate.

G661=nnnnn Load repeat counter 1. This command loads the repeat count "nnnnn" (nnnnn <= 65535).

G662=nnnnn Same as above, but for counter 2.

G663=nnnnn Same as above, but for counter 3.

G664=nnnnn Same as above, but for counter 4.

G665=nnnnn Same as above, but for counter 5.

G666=nnnnn Same as above, but for counter 6.

G667=nnnnn Same as above, but for counter 7.

G668=nnnnn Same as above, but for counter 8.

G671 Decrement/skip-if-zero repeat counter 1, (if zero, skip to the end-of-block).

G672 Same as above, but for counter 2.

G673 Same as above, but for counter 3.

G674 Same as above, but for counter 4.

G675 Same as above, but for counter 5.

G676 Same as above, but for counter 6.

G677 Same as above, but for counter 7.

-
- G678 Same as above, but for counter 8.
- G90 Absolute mode.
- G91 Incremental mode.
- G92 Preset X position register (XPSN) and
 Y position register (YPSN).

CHAPTER 6

SERVICE AND REPAIR

In the event that there is a problem with your Unidex III, DO NOT ATTEMPT TO REPLACE ANY COMPONENTS. There are no customer servicable parts and TO DO SO WOULD NULL YOUR WARRANTY.

If there is a problem, contact:

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

Customer Service Department

NOTE: Located on the rear panel of Unidex III is the Serial Number. WHENEVER YOU CONTACT AEROTECH, INC. IN REFERENCE TO YOUR UNIDEX III, HAVE THE SERIAL NUMBER ON HAND. All special information concerning your system is referenced through the serial number.

APPENDIX 1

CHARACTER	MEANING	CHARACTER	MEANING	CHARACTER	MEANING
0	DIGIT 0	DEL (RUBOUT)	DEVICE RESET #	T	TRIGGER MODE #
1	DIGIT 1	A	AUTO MODE	U	U CODES
2	DIGIT 2	B	CONTINUE PROGRAM AFTER PROGRAM INITIATED SRQ (G25)	V	V CODES
3	DIGIT 3	C	DEVICE CLEAR #	W	CONFIGURE SRQ USING PROGRAMMABLE CHARACTER #
4	DIGIT 4	D	DWELL	X	X CODES
5	DIGIT 5	E	EDIT	Y	Y CODES
6	DIGIT 6	F	FEED	Z	CANCELS W COMMAND #
7	DIGIT 7	G	G CODES		
8	DIGIT 8	H	HOLD FOR TRIGGER		
9	DIGIT 9	I	IMMEDIATE MODE		
-	MINUS	J	CONFIGURE SERIES RTS #		
=	EQUAL TO	K	CONFIGURE PARALLEL RTS #		
>	GREATER THAN	L	GO TO LOCAL #		
<	LESS THAN	M	MISCELLANEOUS CODE		
!	START OF COMMENT FIELD	N	LINE NUMBER		
\$	ERASE (EDIT MODE)	O	CANCEL HOLD #		
*	END-OF-BLOCK	P	PRINT: Pnn - Print program nn PD - " directory POO - " all programs PX - " X position PY - " Y position PS - " statuses		
CR LF	CARRIAGE RETURN/ LINE FEED (EXECUTE)	Q	QUERY (SERIAL POLL) #		
EOI	END OR IDENTIFY (EXECUTE)	R	REMOTE ENABLE (QUIT CURRENT MODE AND DISPLAY "RMT ENAB")		
ESC	BEGINS BUS INTER-FACE COMMANDS #	S	SINGLE MODE		

(APPLIES TO RS-232/422)

TABLE 4-3.1 ASCII CHARACTER SET

SWITCH					DECIMAL ADDRESS
5	4	3	2	1	
OFF	OFF	OFF	OFF	OFF	31*
OFF	OFF	OFF	OFF	ON	30
OFF	OFF	OFF	ON	OFF	29
OFF	OFF	OFF	ON	ON	28
OFF	OFF	ON	OFF	OFF	27
OFF	OFF	ON	OFF	ON	26
OFF	OFF	ON	ON	OFF	25
OFF	OFF	ON	ON	ON	24
OFF	ON	OFF	OFF	OFF	23
OFF	ON	OFF	OFF	ON	22
OFF	ON	OFF	ON	OFF	21
OFF	ON	OFF	ON	ON	20
OFF	ON	ON	OFF	OFF	19
OFF	ON	ON	OFF	ON	18
OFF	ON	ON	ON	OFF	17
OFF	ON	ON	ON	ON	16

SWITCH					DECIMAL ADDRESS
5	4	3	2	1	
ON	OFF	OFF	OFF	OFF	15
ON	OFF	OFF	OFF	ON	14
ON	OFF	OFF	ON	OFF	13
ON	OFF	OFF	ON	ON	12
ON	OFF	ON	OFF	OFF	11
ON	OFF	ON	OFF	ON	10
ON	OFF	ON	ON	OFF	9
ON	OFF	ON	ON	ON	8
ON	ON	OFF	OFF	OFF	7
ON	ON	OFF	OFF	ON	6
ON	ON	OFF	ON	OFF	5
ON	ON	OFF	ON	ON	4
ON	ON	ON	OFF	OFF	3
ON	ON	ON	OFF	ON	2
ON	ON	ON	ON	OFF	1
ON	ON	ON	ON	ON	0

* NOT USED

TABLE 4-4.1

IEEE-488 AND RS-232/422 DAISY CHAIN

IEEE-488 STANDARD INTERFACE BUS SIGNAL LINE

MNEMONIC	NAME	FUNCTION
<u>BUS MANAGEMENT LINES (UNILINE MESSAGES)</u>		
IFC	INTERFACE CLEAR	System controller alone can assert this line, to place all devices in the unaddressed state. Devices go into Talker idle/Listener idle state. If control has been passed to another device, system controller again becomes active by asserting IFC.
ATN	ATTENTION	Asserted true by active controller to send bus interface messages on the bus. When ATN is asserted, signals on the data lines are interpreted as messages. ATN asserted with EOI to do a parallel poll. When ATN is false, data may be sent over the bus by a designated talker.
REN	REMOTE ENABLE	Asserted to program devices on the bus remotely. Any device addressed to listen while REN is true, is placed in Remote mode of operation.
SRQ	SERVICE REQUEST	Asserted by a device to indicate its need for interaction with the controller.
EOI	END OR IDENTIFY	When asserted, indicates the termination of flow of data. Asserted when the last data byte is placed on the bus.
<u>DATA HANDSHAKE LINES</u>		
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 the 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.
<u>DATA LINES</u>		
DIO-1 - DIO-8	DATA LINES	Used for sending data (ATN line false) or Bus interface messages (ATN line true).

IEEE-488 CABLE MANUFACTURERS

Hewlett-Packard

Palo Alto, California 94304

<u>PN</u>	<u>Length</u>
HP 10833D	.5 Meter
HP 10833A	1 Meter
HP 10833B	2 Meters
HP 10833C	4 Meters
HP 10834A	Adapter

Belden Corporation

Richmond, Indiana 47374

<u>PN</u>	<u>Length</u>
9642	1 Meter
9643	2 Meters
9644	4 Meters
9645	8 Meters
9646	16 Meters

UNIDEX III STATUS BYTES

1. Serial Poll Status Byte

When serial polled, Unidex III sends one byte of data to the host computer. When using the IEEE-488 bus, the status byte is sent as a part of the normal serial polling sequence. When using the RS-232/422 serial line, the status byte is followed by <CR><LF><ETX> characters.

The 8 bits of the status byte are explained below:

	7	6	5	4	3	2	1	0	
0	No Error	SRQ not active	REMOTE state	REMOTE enabled	Not busy	Not in HOLD mode	Not G25	Not MO	0
1	Error State	SRQ active	LOCAL state	REMOTE disabled	Busy running program	In HOLD mode	In G25 stop	In MO stop	1

2. Detailed Status Bytes (Sent on command "PS")

In remote mode, when the command "PS" is sent to Unidex III, 10 bytes of data are returned, each in an 8-bit binary format. When using the IEEE-488 bus, the 10 bytes are followed with <CR><LF>. When using the RS-232/422 serial line, they are followed with <CR><LF><ETX>. Please note that an 8-bit character format is required to read the status bytes completely.

a) SYSTEM CONFIGURATION STATUS

	7	6	5	4	3	2	1	0	
0	DC system	LO spd. stepper system	Hard-ware Home	X-axis CW home	Y-axis CW home	LOCAL on LIMIT	No key-board	Not used	0
1	STEPPER system	HI spd. stepper system	Soft-ware Home	X-axis CCW home	Y-axis CCW home	REMOTE on LIMIT	key-board		1

b) SYSTEM_STATUS_1 (Same as serial poll status)c) SYSTEM_STATUS_2

	7	6	5	4	3	2	1	0	
0	Not used	Not used	Not used	Not used	Not used	Not used	Non-Corner Rounding	Absolute Mode	0
1							Corner Rounding Mode	Incremental Mode	1

d) COMMUNICATION_STATUS

	7	6	5	4	3	2	1	0	
0	Keybd. not active	GPIB not active	RS-232/ 422 not active	Not listener addrsd.	Not talker addrsd.	Not used	SRQ via RTS	Not used	0
1	Keybd. active	GPIB active	RS-232/ 422 active	Listener addrsd.	Talker addrsd.		SRQ via char		1

e) COMMUNICATION_ENABLED_STATUS

	7	6	5	4	3	2	1	0	
0	Keybd. dis- abled	GPIB dis- abled	RS-232/ 422 disabld.	Trsmtr. enabld. by DC1	DC1 sent on line	Not used	Not used	Power up or keybd. Reset	0
1	Keybd. en- abled	GPIB en- abled	RS-232/ 422 enabld.	Trsmtr. disabld. by DC3	DC3 sent on line			Remote Device Clear	1

f) SYNTAX_ERROR_STATUS

	7	6	5	4	3	2	1	0	
0	No Syntax Errors								0
1	Error	Illegal Char.	N error	G error	F error	M error	D error	No command	1

g) RUN_ERROR_STATUS

	7	6	5	4	3	2	1	0	
0	No Run Errors								0
1	Error	X Limit	Y Limit	Not used	Stack ovflw	Not used	EOB srch.	Missing program	1

h) EDITOR_ERROR_STATUS

	7	6	5	4	3	2	1	0	
0	No Editor Errors								0
1	Error	Not used	Not used	Not used	Not used	Not used	Not used	Compile error	1

i) M-FUNCTION_OUTPUT_STATUS

	7	6	5	4	3	2	1	0	
0	M8 off	M7 off	M6 off	M5 off	M4 off	M3 off	M2 off	M1 off	0
1	M8 on	M7 on	M6 on	M5 on	M4 on	M3 on	M2 on	M1 on	1

j) CONDITION_INPUT_STATUS

	7	6	5	4	3	2	1	0
0	C4	C3	C2	C1	No Fault	Intrnl clock	Last device	SRQ "ANDED" in daisy chain
1					Fault light on	Joystick clock	Not last device	SRQ "ORED" in daisy chain

INDEX

NOTE: Major topic reference is underscored

-A-

ABORT ON INTERRUPT - 1-3; 4-26; 5-13, 17, 19_to_21
ABSOLUTE MODE - 5-4_to_6, 13_to_15
ABSOLUTE POSITION REGISTER - 4-26; 5-7
ADDRESSING UNIDEX III - 2-12; 4-14, 19, 23; 5-36,
37; AP-1
A.C. POWER - 3-1
ALPHANUMERIC DISPLAY - 1-2; 3-3, 4_TO_6; 4-1
ARROW KEYS - 2-10, 11; 4-10, 11
ASCII - 2-12 TO 14; 4-13, 18, 24; AP-1
ATTENTION LINE - 4-20
AUTO MODE - 2-2, 8, 9, 13; 3-5; 4-7, 8, 16, 17; 5-
29, 30
AXIS COMMANDS (X AND Y CODES) - 5-1, 3_TO_7, 47
 Xnnnnnnnnnn - 5-2_TO_5, 47
 Ynnnnnnnnnn - 5-2_TO_5, 47
AXIS CONTROL - 1-2; 2-10 TO 12; 5-1, 3_TO_7, 47
AXIS SELECT SWITCH - 3-3

-B-

BACKSTEP/STEP - 2-2, 4, 5; 4-3; 5-28
BATTERY BACK-UP - 3-3; 4-29
BAUD RATE - 4-13, 24
BLOCKS - 2-2, 9; 4-8, 17, 27; 5-12, 13, 16, 17, 25
BUFFER - 4-9, 10
BUS - 4-18_TO_21, 23; 5-36; AP-1

-C-

CABLE - 4-20; AP-1
CAPABILITIES - 4-25_TO_29
CARRIAGE RETURN/LINE FEED - 2-12, 13; 4-13, 15, 18;
5-37
CHARACTER LENGTH - 4-13, 24
CHASSIS - 1-1; 3-2, 3; 5-25
CLEAR COMMAND - 2-2, 7, 8; 4-5_TO_7; 5-27
CLEAR DEVICE - 5-44
CLEAR ENTRY - 2-2, 7, 8; 4-5_TO_7; 5-27
CLEAR PROGRAMS - 4-16; 5-35, 36
CLOCK PULSES - 5-14, 15, 18
CODE SUMMARIES - 1-2; 2-1; 5-47_TO_55

COMMANDS (MODE AND MOTION) - 2-1 TO 4; 4-1, 14, 18, 25, 28
 AXIS - 5-2 TO 7, 47
 DWELL - 5-2, 10, 49
 END-OF-BLOCK - 2-13; 4-2, 8, 26 TO 28; 5-5, 12, 13, 16 TO 18, 50
 FEEDRATE - 5-1 TO 5, 7, 8, 49
 MISCELLANEOUS - 5-1, 2, 8, 9, 48
 PREPARATORY - 5-1, 2, 13 TO 23, 49 TO 55
 SEQUENCE (LINE) NUMBERS - 5-1, 10, 11, 47, 48
 COMPONENTS - 3-2
 COMPOUND COMMANDS - 5-6, 7
 CONDITIONAL SKIP - 1-3; 4-26; 5-12
 CONNECTORS - 3-3, 4; 4-13, 20, 24, 25; 5-25
 CONTROL LINES - 4-22, 23; 5-24
 CONTROL SIGNALS - 4-18 TO 22; 5-24, 25
 CONTROLLER - 4-18 TO 25; 5-24, 25
 CORNER-ROUNDING/NON-CORNER-ROUNDING - 3-2; 4-1, 27; 5-6, 14, 15, 18
 CNC - 5-1
 CTS (CLEAR TO SEND) - 4-22; 5-24
 CUSTOMER SERVICE - 3-4; 6-1

-D-

D CODES - SEE DWELL
 DAISY CHAIN - 4-24, 25; AP-1
 DATA LINES - 4-20 TO 22; 5-24; AP-1
 DATA SIGNALS - 4-18 TO 22; 5-24
 DC DRIVE SYSTEM - 3-2; 5-6, 14, 15, 18; AD-1
 DC POWER - 3-2
 DCD (DATA CARRIER DETECT) - 4-22; 5-24
 DECREMENT REPEAT COUNTER - 4-28
 DEFAULT OPTIONS - 3-2; 4-1, 2, 7, 12; 5-6, 8
 DELETE PROGRAMS - 4-16
 DISPLAY MESSAGE - 3-2, 4-1, 4
 DISPLAY PANEL - 2-3; 3-3 TO 5; 4-1, 4
 DOWN COUNTER - 4-27, 28
 DOWNLOADING - 1-3; 2-13; 4-15; 5-25
 DRIVES - 2-3, 10, 12; 3-1, 2; 4-1, 10 TO 12; 5-14, 15
 DSR (DATA SET READY) - 4-22; 5-24
 DTR (DATA TERMINAL READY) - 4-22; 5-24

DUPLEX OPERATION - 4-24
 DWELL (D CODE) - 1-3; 5-1, 10, 19, 49
 Dnnnnnnn - 5-2, 10, 49
 D=nnnnn - 5-10, 49

-E-

EDIT FUNCTIONS - 2-2, 4-10-8, 13; 4-3-10-7, 14; 5-26-10-29
 EDIT MODE - 2-1, 2-10-8, 12, 13; 4-1-10-7, 14-10-16; 5-37-10-39
 EDITING A PROGRAM - 2-4-10-8; 4-3-10-7, 14-10-16; 5-26-10-29, 37-10-39
 ENCODERS - 3-2, 4
 END-OF-BLOCK COMMANDS (EOB) - 2-13; 4-2, 8, 26-10-28; 5-5, 12, 13, 16-10-18, 50
 END OF MEMORY - 4-29
 END OF PROGRAM - 4-9, 14, 15, 27; 5-37
 ENTERING A PROGRAM - 2-3, 4, 12, 13; 4-2, 14, 15; 5-37-10-39
 EQUIPMENT DESCRIPTION - 3-1-10-7
 ERASE - 2-2, 13; 4-5-10-7, 16; 5-43
 ERROR MESSAGE - 5-4, 5; AD-1
 EXECUTE - 2-3 TO 8; 3-6; 4-7, 8, 13; 5-27, 29
 EXECUTION OF COMMANDS - 5-2-10-4; 5-10-23

-F-

F CODES - SEE FEEDRATE COMMANDS
 FEATURES OF UNIDEX III - 1-1, 2; 3-3; 4-13, 25-10-29
 FEEDRATE COMMANDS (F CODES) - 1-1, 2; 2-10, 11; 4-10-10-12, 27; 5-1, 3-10-5, 7, 8, 49
 Fnnnnnn - 5-2, 3, 7, 8, 49
 F=nnnnnn - 5-7, 8, 49
 F-nnnnn - 5-8, 49
 FLAGS - 5-19-10-22, 51-10-53
 FRONT PANEL - 1-3; 2-1; 3-2-10-6; 4-1, 15
 FUNCTIONS OF UNIDEX III - 1-3

-G-

G CODES (PREPARATORY CODES) - SEE PREPARATORY COMMANDS

-H-

HANDSHAKE LINES - 4-20; AP-1
 HOLD - 4-13, 14
 HOME - AD-1
 HOST - 1-1; 2-1, 12, 14; 4-13 TO 16, 18 TO 25; 5-23
 TO 25, 36, 37

-I-

IEEE-488 - 1-1; 3-3; 4-13, 18 TO 21; 5-24, 36,
 38 TO 43
 IMMEDIATE MODE - 2-9 TO 12, 14; 3-5; 4-1, 9 TO 12,
 17, 18; 5-30 TO 34, 41, 42
 INCREMENTAL MODE - 3-2; 4-1, 27; 5-4 TO 6, 13 TO
 15
 INDEXING LOGIC - 3-3
 INPUTS - 1-1, 2; 3-2 TO 4, 7; 4-26; 5-16 TO 20,
 49 TO 51; AD-1
 INSTALLATION - 3-1, 2
 INTERFACING - 1-1; 3-3; 4-1, 18 TO 25; 5-23 TO 46
 LOCAL - 1-1; 2-3 TO 12; 4-1 TO 12; 5-23 TO 36
 IEEE-488 - 1-1, 3-3; 4-1, 13, 18 TO 21; 5-24, 25,
 38 TO 43
 RS-232 - 1-1; 3-3; 4-1, 13, 18, 21 TO 25; 5-24,
 25, 36 TO 46
 RS-422 - 1-1; 3-3, 4; 4-1, 13, 18, 21 TO 25;
 5-25, 36 TO 46
 INTERRUPT - 4-26; 5-16 TO 21

-J-

J COMMAND - 4-22, 23; 5-45
 JOYSTICK - 1-1, 2; 2-11; 3-3, 4, 7; 4-11, 12
 JOYSTICK FEEDRATE - 2-11; 4-11, 12
 JOYSTICK FREQUENCY - 2-11; 4-11, 12
 JUMP - 5-17 TO 19

-K-

K COMMAND - 4-22, 23; 5-46
 KEYBOARD - 1-1, 2-3; 3-3 TO 6; 4-13, 15, 29
 KEYS - 1-1; 3-4 TO 6

-L-

LAG - 5-6, 14, 23
 "LAST DEVICE" SWITCH - 4-25
 LEDS - 2-3, 12; 3-4, 5
 LINE NUMBERING (N CODES) - 4-25, 27, 28; 5-1,
 10 TO 12, 47, 48
 Nnnnnnn - 5-10, 11, 47
 N-nnnnnnn - 5-11, 47
 N=nnnnnn - 5-11, 48
 N>nnnnnn - 5-11, 48
 LISTENER - 4-18, 19
 LOCAL/IMMEDIATE MODE - 4-10 TO 12
 LOCAL PROGRAMMING - 1-2; 2-1, 3 TO 12; 3-2; 4-1
 10 12, 15; 5-23, 24, 26 TO 36, 45
 LOCAL WITH REMOTE ENABLED - 2-3, 12; 3-2; 4-1, 13
 LOGIC - 3-3; 4-26; 5-16

-M-

M CODES - SEE MISCELLANEOUS COMMANDS
 MANUAL CONTROL - 2-10 TO 12; 4-10 TO 12, 13;
 5-32 TO 34
 MEMORY - 1-1, 3; 2-10; 3-3; 4-15, 16, 29; 5-35
 CAPABILITY - 4-29
 EXPANDING - 4-29
 END OF MEMORY - 4-29
 MISCELLANEOUS COMMANDS (M CODES) - 5-1, 2, 8, 9, 48
 M0 - 5-9, 49
 M2 - 5-9, 49
 M30 - 5-9, 49
 M47 - 5-9, 49
 M99 - 5-9, 49
 M-99 - 5-9, 49
 M=nnnn - 5-9, 49
 MODAL - 5-13
 MODE COMMANDS - 2-1

MODE SELECT SWITCHES - 3-3, 4
 MODES OF OPERATION -
 AUTO - 2-1, 8, 9, 13; 4-1, 7, 8, 16, 17; 5-29,
 30
 EDIT - 2-1 TO 8, 13; 4-1 TO 7, 14 TO 16; 5-26
 TO 29, 37 TO 39
 IMMEDIATE - 2-1, 2, 9, 10, 14; 4-1, 9 TO 12,
 17, 18; 5-30 TO 34, 41, 42
 SINGLE - 2-1, 2, 9, 13; 4-1, 8, 9, 17; 5-30
 MOTION COMMANDS - 2-1, 4
 MOTORS CONNECTIONS - 3-1, 4
 MPU BOARD - 3-2, 3
 MPX BOARD - 3-2
 MULTI-AXIS - 1-3, 3-3; 4-24

-N-

N CODES (LINE NUMBERS) - SEE LINE NUMBERING
 NON-CORNER-ROUNDING - 3-2; 4-1, 27; 5-6, 14 TO 18
 NON-MODAL - 5-13, 14 TO 16

-O-

O COMMAND - 5-45
 OPERATING MODES -
 LOCAL - 2-3 TO 12; 4-1 TO 12; 5-23, 24,
 26 TO 36
 AUTO - 2-8, 9; 4-1, 7, 8; 5-29, 30
 EDIT - 2-2 TO 8; 4-1 TO 12; 5-26 TO 29
 IMMEDIATE - 2-9 TO 11, 14; 4-1, 9 TO
 13; 5-30 TO 34
 SINGLE - 2-9; 4-1, 8, 9, 13, 17; 5-30, 31
 REMOTE - 2-12 TO 14; 4-13 TO 18; 5-36 TO 46
 AUTO - 2-13; 4-13, 16, 17; 5-39, 40
 EDIT - 2-12, 13; 4-13 TO 16; 5-37 TO 39
 IMMEDIATE - 2-14; 4-13, 17, 18; 5-41, 42
 SINGLE - 2-13; 4-13, 17; 5-40, 41
 OPERATOR'S MANUAL - 1-1
 OUTPUTS - 1-1, 2; 3-2 TO 4; 5-2, 8, 9; AD-1

-P-

PARALLEL POLLING - 4-20, 21
 PARALLEL RTS - 4-22, 23
 PARITY/NON-PARITY - 4-13, 24
 PAUSE - SEE DWELL
 POLLING - 4-20, 21
 POSITION STATUS - 4-28; AD-1
 POWER REQUIREMENTS - 3-1
 POWER SUPPLY - 3-2
 POWERING UP - 2-3, 12; 3-1, 2; 4-1, 13
 PPR DIPSWITCH - 4-21
 PREPARATORY COMMANDS (G CODES) - 5-1, 13 TO 23,
 49 TO 55

 G7 - 5-2, 49
 G10 - 5-49
 G11 - 5-49
 G12 - 5-49
 G23 - 5-6, 15, 49
 G24 - 3-2; 5-6, 15, 50
 G25 - 5-50
 G271 - G274 - 4-26; 5-16, 50
 G281 - G284 - 5-50, 51
 G301 - G304 - 4-26; 5-17 TO 20, 51
 G311 - G314 - 5-20, 21, 51
 G501 - G508 - 5-20, 21, 51, 52
 G511 - G518 - 5-20, 21, 52
 G521 - G528 - 5-20, 21, 52, 53
 G531 - G538 - 5-20, 21, 53
 G60 - 5-21, 53
 G60=nnnnnn - 5-21, 53
 G61 - 5-21, 53
 G61=nnnnnn - 5-21, 53
 G661=nnnnn TO G668=nnnnn - 5-21, 22, 53, 54
 G671 TO G678 - 4-28; 5-22, 54, 55
 G90 - 5-4, 5, 13, 22, 55
 G91 - 3-2; 5-4 TO 6, 13, 22, 55
 G92 - 5-7, 23, 55
 PRINT - 1-3; 3-6; 4-15, 16, 28; 5-42, 43
 PRINT POSITIONS - 4-28
 PRINT STATUSES - 4-28; AD-1
 PRINT-OUTS - 1-3; 3-5; 5-34, 35
 PROGRAM FLOW - 4-26; 5-19
 PROGRAMMABLE ARRAY LOGIC DEVICE (PAL) - 4-29
 PROGRAMMED STOP - 5-17, 18
 PROGRAMMING - 1-1; 2-1; 4-1
 FEATURES - 1-2, 3; 3-3; 4-13, 25 TO 29

RS-274-D LANGUAGE - 4-25; 5-1 TO 23
 LOCAL - 1-1; 2-1, 3 TO 12; 4-1 TO 12; 5-26
 TO 36
 REMOTE - 1-1; 2-1, 12 TO 14; 4-1, 13 TO 18;
 5-36 TO 46
 PROTOCOL - 4-18, 19, 24; AD-1

-Q-

QUERY (Q COMMAND) - 5-46

-R-

RAM - 4-1
 REAR PANEL (INTERFACE BOARD) - 3-2 TO 4; 4-13, 24,
 25; 5-25
 REMOTE PROGRAMMING - 1-2, 3; 2-1, 12 TO 14; 3-2;
 4-1, 13 TO 18; 5-36 TO 46
 REMOTE KEY - 3-5; 4-1; 5-26
 REPEAT COUNTER - 4-27, 28
 REPEAT CYCLES - 1-3; 4-27, 28
 RESET - 3-4, 5
 RETURN FROM SUBROUTINE - 4-27
 RS-232C - 1-1; 3-3; 4-1, 13, 18, 21 TO 25; 5-24, 25,
 36 TO 46
 RS-274-D COMMANDS - 5-1 TO 23, 47 TO 55
 COMPLETE LISTING - 5-47 TO 55
 EXPLANATION - 5-1 TO 23
 SEQUENCING - 5-3, 4
 RS-422A - 1-1; 3-4; 4-1, 13, 18, 21 TO 25; 5-25,
 36 TO 46
 RTS (REQUEST TO SEND) - 4-22, 23; 5-24, 45, 46
 SERIES RTS - 4-22, 23
 PARALLEL RTS - 4-22, 23
 PROGRAMMABLE SYMBOL FOR RTS - 4-22, 23

-S-

SAMPLE PROGRAMS - 2-3 TO 14; 5-26 TO 46
 LOCAL - 2-3 TO 11; 5-26 TO 36
 REMOTE - 2-12 TO 14; 5-37 TO 46
 SEARCH - 2-2, 5, 6; 4-4 TO 6; 5-26, 28, 29

SELECTOR SWITCHES - 4-13
 SELF-TEST - 3-1, 2; 4-1
 SEND INFORMATION ON COMMAND - 4-28
 SEQUENCE NUMBER COMMANDS (N CODES) - SEE LINE NUMBERING

 SERIAL INTERFACE - 4-13, 24, 25; 5-25
 SERIAL NUMBER - 3-4; 6-1
 SERIAL POLLING - 4-17, 20, 21, 28; 5-40 TO 42, 46;
 AD-1
 SERIAL RTS - 4-22, 23
 SERVICE - 1-2; 6-1
 SERVICE REQUEST - 4-17, 18, 20 TO 23, 28
 SHIFT KEY - 3-4
 SHIPPING - 3-1 TO 3
 SIGNAL LINES - 4-20, 22; AD-1
 SINGLE MODE - 2-2, 9, 13; 3-6; 4-1, 8, 9, 17;
 5-30, 31
 SKIP TO EOB - 4-26 TO 28
 SLEW FUNCTION - 2-10 TO 12; 4-10 TO 12; 5-33, 34
 SLEW WITH JOYSTICK - 2-11, 12; 4-11, 12; 5-33, 34
 SRQ - (SEE SERVICE REQUEST)
 STATUSES - 1-3; 4-13, 24, 28; 5-42; AD-1
 STEP FUNCTION - 2-10; 4-10, 13; 5-32, 33
 STEP/BACKSTEP - 2-2, 5; 4-3; 5-26, 28, 29
 STEPPER SYSTEM - 3-2; 5-6, 14; AD-1
 STOP BITS - 4-13
 STORE - 1-3; 4-15, 16
 STORED PROGRAM CAPABILITIES - 1-2, 3; 4-15, 29
 STROBE - 1-1, 2; 3-3; 5-8, 9
 SUBROUTINE CALL - 1-3; 4-27
 SYSTEM CONFIGURATION - 3-2; AD-1

-T-

TALKER - 4-18, 19
 TEST - 3-1, 2; 4-1
 RAM CHECK - 4-1
 ROM CHECK - 4-1
 TRACKING DISPLAYS - 1-2; 3-3, 6
 TRIGGER - 1-3; 4-13, 14; 5-44

-U-

UNCONDITIONAL JUMP - 1-3; 4-25
UNIVERSAL COMMANDS - 5-44 TO 46
UNPACKING UNIDEX III - 1-1; 3-1
USER ACCESSIBLE REGISTERS - 5-47

-W-

W COMMAND - 4-23, 24

-X-

X CODES - SEE AXIS COMMANDS
X AND Y DISPLAYS CLEAR - 3-6
X AND Y TRACKING DISPLAYS - 3-6
X ON, X OFF - 4-24

-Y-

Y CODES - SEE AXIS COMMANDS

-Z-

Z COMMAND - 5-45



Warranty and Field Service Policy

Aerotech, Inc., warrants its products to be free from defects caused by faulty materials or poor workmanship for a period of one year from date of shipment from Aerotech. Seller's liability is limited to replacing, repairing or issuing credit, as its option, for any products which are returned by the original purchaser during the one-year period. Seller 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 seller in specifications or drawings previously or subsequently provided seller, and whether or not seller's products are specifically designed and/or manufactured by seller for buyer's use or purpose. Aerotech's liability in 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.

RETURNING GOODS PROCEDURE

Claims for incorrect or defective materials must be filed within thirty (30) days from delivery at buyer's place of business. No units or systems may be returned, in or out of warranty, without first obtaining approval from the seller, and no claim will be allowed nor credit given for units or systems returned without such approval.

RETURNED GOODS WARRANTY DETERMINATION

If possible, after approval from Aerotech, the defective unit or system is to be returned to the factory with statement of problem and transportation prepaid (no c.o.d. or collect freight shipments will be accepted). After Aerotech's in-plant examination, warranty or out-of-warranty status will be determined. If upon Aerotech's examination of such unit or system, warranted defects exist, then the unit or system will be repaired

at no charge and shipped, prepaid, back to the buyer. If an out-of-warranty situation exists, the buyer shall be notified of the repair cost immediately. At such time, the buyer must issue a purchase order to cover the cost of the repair or authorize the unit or system to be shipped back as is, at the buyer's expense.

ON-SITE WARRANTY REPAIR

If the system or unit cannot be made functional by telephone assistance or by sending and having customer install replacement parts, and cannot be returned to the Aerotech factory for repair, and if it is determined that 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 bona-fide purchase order to Aerotech covering all transportation and subsistence costs, regardless of warranty determination. If the Aerotech field service representative determines during his on-site repair that the system or unit's problem is not warranty-related, then the prevailing service charge per hour (eight-hour minimum) shall be assessed against the issued purchase order.

NON-WARRANTY FIELD SERVICE

If system or unit cannot be made functional by no-charge telephone assistance or purchased replacement parts, and cannot be returned to the Aerotech factory 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 bona-fide purchase order to Aerotech covering all transportation and subsistence costs and the prevailing cost per hour (eight-hour minimum) including travel time necessary to complete the repair.

SALES OFFICESAEROTECH NORTHEAST (629)

Executive Suite 202
270 Farmington Avenue
Farmington, CT 06032
(203) 673-3330
John S. Mucha
MA, CT, VT, ME, RI, NH,
NJ, DE, E.PA, Upst.NY

AEROTECH MIDWEST (632)

102 N. Main Street
Columbiana, OH 44408
(216) 482-9957
(216) 482-2775
Frank W. Armstrong
OH, IA, IN, IL, MI, MN, MO, KY, WV
KS, WI, W.NY (140,141,142,143,147)

CW RADIATION, INC. (637)

140 Kifer Court
Sunnyvale, CA 94086
(408) 738-0690

AEROTECH WEST (630)

1328 Monterey Boulevard
Hermosa Beach, CA 90254
(213) 376-9698
(408) 738-1220
Russell McKissick
AZ, CA (ZIP 935<)

7155 East Thomas Road (641)
Suite 7
Scottsdale, AZ 85251
(602) 941-5341

AEROTECH NORTHWEST

140 Kifer Court
Sunnyvale, CA 94086
(408) 738-1220
Marty Bergman
CA (ZIP 936>), OR, WA

REPRESENTATIVESGRADY MOORE ASSOCIATES (639)

5901 Gibson Blvd., S.E.
Albuquerque, NM 87018
(505) 262-1416
TWX 910-989-1691
Grady Moore
CO, ID, MT, NM, NV, UT

Art Himle (634)
PO Box 4619
Boulder, CO 80306
8850 Elgin Drive
Lafayette, CO 80026
(303) 494-3351
TWX 910-940-0398
CO, UT

MCMAHAN ASSOCIATES (635)

2160 Park Avenue North
Winter Park, FL 32789
(305) 645-0463
Robert McMahan
AL, FL, GA, SC

MCMAHAN/ZELLMAN ASSOCIATES (610)

8804 Lomas Court
Raleigh, NC 27609
(919) 848-1965
Charles E. Zellman
NC, TN, S.VA

MOORE ASSOCIATES (631)

319 Arborcrest
Richardson, TX 75080
(214) 690-4929
John Moore
TX, OK, LA, AR

SPECTRO ASSOCIATES, INC. (633)

11445 Beechgrove Lane
Rockville, MD 20854
(301) 279-2010
Steve Cades
MD, DC, N.VA (220,221,222,223,226,
227,228)

INTERNATIONAL SALES OFFICESAEROTECH LTD.

19A Livingstone Road
Newbury, Berkshire
England RG14 7PD
(0635) 46313
TLX 847228
Keith Reynolds

AEROTECH GMBH

Neumeyerstrass 90
8500 Nurnberg 10
West Germany
(0911) 521031
TLX 622474
Hans-Werner Bleckwendt

INTERNATIONAL REPRESENTATIVES

Laser Products

LASER OPTRONIC S.A.
Avenue de Broqueville 222
1200 Brussels
Belgium
P. Scheyvaerts
(02) 7712166
TLX 61439

DITTA D.E.S. SRL
Via Pratese 164
50100 Florence
Italy
Dr. Benini
(055) 370331
TLX 570298

MEDILAS AG
Lerzenstr. 11
8953 Dietikon
Switzerland
J. Goessi
(01) 7411111
TLX 59567

MEDILAS HGmbH
Vorstadt 2
6800 Feldkirch
Austria
H. Proem
(5522) 23340
TLX 52504

Y. BEN MOSHE
8a Sheinkin Street
PO Box 18125
Tel Aviv 61181
Israel
(03) 286547
TLX 342436

OPTIKON CORPORATION LTD.
410 Conestogo Road
Waterloo, Ontario
Canada N2L 4E2
Minas Vassiliadis
(519) 885-2551
TLX 069 55373

QUENTRON ELECTRONICS PTY. LTD.
Laser Court, 75A Angas Street
GPO Box 2212
Adelaide, South Australia 5001
Alex Stanco
(08) 223 6533, 223 6224
TLX 82809

ENGINEERING EQUIPMENT CO. LTD.
No. 2, Oda Bldg. 3-1
Higashi-Shimbashi, 1-Chome
Minato-Ku, Tokyo 105, Japan
T. Okada
(03) 572-7071
TLX 2524433

PAUL ENTERPRISES
232 A, Okhla Industrial Estate
Phase 3
New Delhi 110020, India
S. D. Paul

INTERNATIONAL REPRESENTATIVES

Industrial Products

P.V.B.A. SERVOTRONIC S.P.R.L.
Hans Memlingstraat 31
2120 Schoten
Belgium
E. Verbist
(03) 6467047
TLX 72294

MS MECATRONIC SYSTEM A.G.
Elfenastr. 3
2500 Biel
Switzerland
P. Manta
(32) 235023
TLX 34322

ELMEQ B.V.
Nobeldreef 25
2871 KZ Schoonhoven
Holland
Mr. Van der Meer
(1823) 5327
TLX 26401