TABLE OF CONTENTS

I. INTRODUCTION AND SPECIFICATIONS

A. STEPPING MOTORS AND STEPPING MOTOR CONTROLLERS

Stepping motors are a special type of precision motor constructed so that they take discrete steps rather than running continuously as do ordinary motors. Typical stepping motors have from 8 to 400 full steps per revolution, with 200 steps being the most common. Because they can be stepped to a position very precisely, stepping motors are used in many instrument control, factory automation and robotics applications that call for accurate positioning.

To move a stepping motor from one position to another requires that the currents in the motor windings be varied in a specific sequence by a stepping motor controller.

With a half stepping motor controller the motor can be positioned to any of its full step positions or to any position midway between the full steps which are called the half step positions. The Oriel Stepper Mike™ moves 1 micron for each half step.

Inputs to a controller can come from a front panel rotary adjustment knob, front panel switches, a joystick controller, a trackball, a computer or a terminal.

B. DESCRIPTION OF THE 18705/18706 STEPPING MOTOR CONTROLLER

The 18705/18706 is a low-cost intelligent controller that provides for manual or computer control of an Oriel Stepper Mike™ actuator, stepper driven translator or rotator. This unit is provided in a slope front cabinet with all power supplies and motor drivers built-in.

The front-panel controls feature a rotary adjust knob which can be used for manually moving of the motor or for adjusting parameters stored in the controller. A 2 line x 20 character alphanumeric LCD shows the current position and parameter values. Four front panel RUN push buttons provide multiple functions including: jogging the motor individual half steps, indexing the motor a fixed number of steps, running the motor between two fixed target positions or executing four stored programs. Remote manual control can be provided by use of an optional joystick or trackball.

An Intel 8097 16 bit micro-controller provides intelligent motor control, including acceleration and deceleration ramping, stored program control, automatic backlash compensation and relative or absolute positioning. Both the 18705 and 18706 have an RS-232C Serial Interface. In addition to this the 18706 has a GPIB/IEEE-488 interface. The interfaces allow a computer such as a PC to send ASCII characters to the controller and read back position and other parameters.

C. SPECIFICATIONS FOR THE 18705/18706

GENERAL

Input Power:	120 or 240 volts
Fuse Rating:	120V requires one 6/10 amp SLO-BLO Fuse (3AG) 240V requires two 315 mA SLO-BLO Fuses (5 X 20mm)
Operating Temperature Range:	0 to 40°C
Weight:	6 lbs.
Motor Driver	
Resolution:	1/2 step, corresponding to 1 micron on Oriel Stepper Mike™
Driver Type:	Constant Voltage with holding torque reduction

FRONT PANEL CONTROLS

Rotary Adjust Knob:	1 3/4 inch diameter with spinner to adjust position or parameters
Run Switches:	Four square push button switches: +X/P2 and -X/P4 used to control motor direction and for selecting internally stored programs. +Y/P1 and -Y/P3 are used ONLY for selecting internally stored programs.
Select Switch:	Used with Rotary Adjust Knob to select parameters.
Enter Switch:	Used to Enter selected parameters
FRONT PANEL DISPLAY	2 row by 20 column LCD alphanumeric

(Specifications continued on next page)

REAR PANEL CONNECTORS AND SWITCHES

Motor Output Connector:	Oriel Stepper Mike™, 10 pin male, T&B Ansley #622-1016 (Mates with Ansley #609-1030)
Motor Fuse:	1 Amp, 250V, Fast Acting (5 x 20mm)
Configuration Switch:	9 segment mini-DIP Switch for Computer Interface settings
Digital Input/Output	
Connector:	25 pin female DB-25F
Number of Signals:	2 Digital input, 2 Digital output, 2 inputs for + and - LIMITS and HOME input
Signal Levels:	LS TTL compatible
RS-232C Serial Port	
Connector:	25 pin male DB-25P
Baud Rates:	300, 600, 1200, 2400, 4800, 9600, 19200
Echo mode:	Full duplex (with echo) or half duplex (without echo), switch selectable
End of Message Termination:	Carriage Return or Carriage Return plus Line-Feed
GPIB/IEEE-488 Interface	(Model 18706 only)
Contrast Adjust	Adjustment for LCD Front Panel Display
Joystick Connection:	Rear panel 15 pin female connector mates with IBM-PC type single speed joystick. NOTE: This option not included at this time.
Trackball Connection:	Rear-panel 9 pin female connector, mates with IBM-PC type trackball. NOTE: This option is not included at this time.

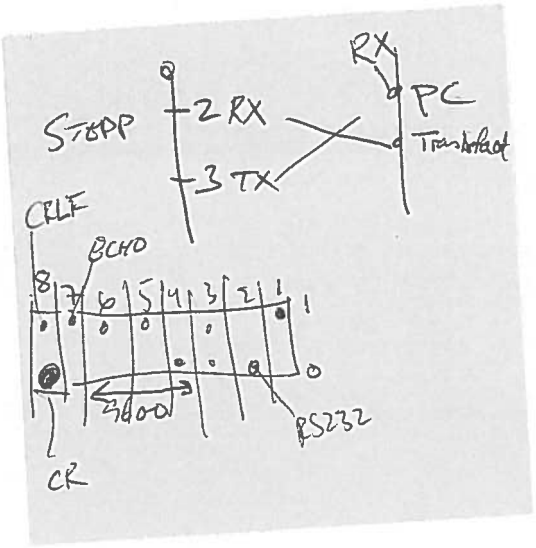
D. START UP

1. Line Voltage

Line voltage is set at the factory. In the event it is necessary to change the line voltage, ***UNPLUG LINE CORD FROM UNIT***, pry the cover off the entry module on the rear and pull out the small circuit board with the white handle/indicator. Rotate the small circuit board so that the desired voltage faces into the module and reinsert the board. Remove the Phillips screw, turn the fuse carrier over, then install screw. Replace cover in the entry module. Note that only 120V and 240V can be chosen. The 100V and 220V selections are not internally connected.

2. Power Up

After selecting the line voltage, plug in the line cord and switch the rear panel power switch on. The front panel display will show the initial parameters. If no display is visible, turn the rear panel contrast adjustment control clockwise to darken the characters on the LCD Display.



II. FRONT PANEL CONTROLS

The front panel controls consist of a **rotary adjust knob** for moving the motor manually or for entering parameters; an **alphanumeric display** that shows the current position plus other information; and four square push buttons which can be used to **JOG** the motor, to **INDEX** the motor a fixed number of steps, to move the motor between **2 TARGET** positions, to run the motor **CONTINUOUSLY**, to run 4 **STORED PROGRAMS**, or to **SCAN UP** or **DOWN**; a pushbutton **SELECT SWITCH** used with Rotary Adjust Knob to select parameters and as an **ABORT RUN** switch; and a pushbutton **ENTER SWITCH** used to **enter selected parameters.**

A. FRONT PANEL ROTARY ADJUST KNOB

The front panel **ADJUST** knob is a rotary optical encoder which inputs pulses to the controller. It can be used to move the motor manually or to adjust values of stored parameters. The parameter to be adjusted is selected by pressing the **SELECT** push button while rotating the **ADJUST** knob until the desired parameter is displayed in the **PARAMETERS** field in the upper left section of the display. The value of the parameter may be changed by releasing the **SELECT** button and turning the **ADJUST** knob until the desired value is obtained. The **ENTER** push button is then pressed to enter the new value.

B. FRONT PANEL ALPHANUMERIC DISPLAY

The front panel display is a 20 character by 2 row liquid crystal display (LCD). Current X position is shown in the lower right field of the display. When a parameter is being set via the rotary adjust knob, the upper left field displays the selected parameter. The present value of any parameter may be viewed in the upper right field by rotating the **ADJUST** knob with the **SELECT** push button held down until the desired parameter is displayed.

C. FRONT PANEL ENTER SWITCH

The front panel ENTER SWITCH is used to ENTER selected settings.

D. USE OF FRONT PANEL ROTARY ADJUST KNOB AND DISPLAY

In the power-up condition, rotating the adjust knob will move the motor manually. To use the adjust knob to change the motor speed, for example, press and hold the **SELECT** button and rotate the adjust know until **"SPEED = 500"** is displayed in the parameter field. Release the **SELECT** button and rotate the adjust knob until the desired speed is displayed and then press the **ENTER** button. You will hear an audible tone and briefly the new value will be displayed in the parameter field. The controller will then switch back to the manual mode as indicated by **"MAN"** being displayed in the upper left parameter field.

E. FRONT PANEL RUN BUTTONS

The front panel **RUN** buttons can be operated in the **JOG** mode, the **CONTINUOUS** mode, the **INDEX** mode, the **TARGET** mode or the **STORED PROGRAM** mode. The mode is switched using the rotary adjust knob with the **SELECT** push buttons held in until the parameter field shows.

"PB > JOG" to switch the RUN buttons to the JOG mode
"PB > CON" to switch the RUN buttons to the CONTINUOUS mode
"PB > NDX" to switch the RUN buttons to the INDEX mode
"PB > TRG" to switch the RUN buttons to the TARGET mode
"PB > PRG" to switch the RUN buttons to the STORED PROGRAM mode
"PB > SCN" to switch the RUN buttons to the SCAN mode

When the desired mode is selected, press **ENTER**. You will hear an audible tone and the lower left corner of the display will show "JOG", "CON", "NDX", "TRG", "PRG" or "SCN" to indicate the current mode of the **RUN** push buttons. The controller is initially in the **JOG** mode at power up.

Note: The 18705/18706 are **SINGLE CHANNEL** controllers (MOTOR X) and only the **+X/P2** and **-X/P4 RUN** buttons control the movement of the Stepper Motor.

1. JOG Mode

In the **JOG** mode, pressing the +X or -X **RUN** push buttons causes the stepper motor to take a 1 micron step in the + or - direction. If a RUN button is held down, the motor will accelerate slightly and continue stepping until the button is released.

2. CONTINUOUS Mode

In the CONTINUOUS MODE, pressing the +X or -X **RUN** push buttons will cause the stepper motor to run **continuously** in the + or - direction.

3. INDEX Mode

In the **INDEX** mode, pressing the +X or -X **RUN** push buttons will cause the stepper motor to take a fixed number of steps in the + or - direction. Upon power up the controller initially sets the INDEX VALUE to 200, but the number of steps, determined by the parameters **N STEPS**, can be re-set using the rotary adjust knob or the computer interface.

4. TARGET Mode

In the **TARGET** mode, pressing the +X push button will cause the stepper motor to move to the TARGET B position and pressing -X push button will cause the stepper motor to move to the TARGET A position. Upon Power Up, the controller initially sets TARGET B (TARG B) to 200 and TARGET A (TARG A) to 0, but the TARGET A and TARGET B positions can be re-set using the rotary adjust knob or the computer interface.

5. STORED PROGRAM Mode

In the **STORED PROGRAM** mode, pressing the **P1 RUN** push button will cause the controller to execute STORED PROGRAM 1, pressing the **P2 RUN** push button will cause the controller to execute STORED PROGRAM 2, pressing the **P3 RUN** push button will cause the controller to execute STORED PROGRAM 3 and pressing the **P4 RUN** push button will cause the controller to execute STORED PROGRAM 4. See Section IV for more information.

6. SCAN Mode

In the **SCAN** mode, pressing the **-X RUN** push button will cause the stepper to SCAN DOWN. Pressing the **+X RUN** push button will cause the stepper to SCAN UP.

7. Abort of Motor Motion

In the CONTINUOUS, INDEX, TARGET or SCAN modes the motor can be stopped by pressing the **SELECT** button or by sending the **ABORT** command through the computer interface.

8. Use of the Front Panel Run Buttons

In the power up condition the controller is in the JOG mode. To change to the INDEX mode, for example, press and hold the **SELECT** push button and rotate the **ADJUST** knob until **"PB>NDX"** is displayed in the parameter field. Release the SELECT push button. Press the **ENTER** push button; you will hear an audible tone and "NDX" will be displayed in the lower left corner of the display to indicate that the RUN push buttons are in the INDEX mode. To set the number of steps to be indexed for X, press and hold the **SELECT** button and rotate the **ADJUST** knob until **"N STEP = 200.0"** is displayed in the parameter field. **NDX X = 0.0** will be displayed in the line below. Release the **SELECT** button and rotate the **ADJUST** knob until the desired number of steps (50, for example) is displayed and then press **ENTER**. You will hear an audible tone. You have now set the stepper to INDEX 50 steps. Pressing **+X RUN** will cause the stepper motor to take +50 steps. Pressing the **-X RUN** will cause the X motor to take -50 steps.

III. INPUT PARAMETERS AND FUNCTIONS

The input parameters are listed in Table III-1 on page 9. All functions can be executed using the front panel controls, the RS-232C Serial interface or the optional GPIB/IEEE-488 bus interface. Refer to the Section VI. for details and examples of using the RS-232C interface. Refer to Section VII. for details and examples of using the GPIB/IEEE-488 interface (Model 18706 only).

In Table III-1 the first column shows the parameter spelled out in full.

The second column shows how the parameters or function is abbreviated to 6 characters for display in the parameter field on the LCD Display. The dashes in this column indicate functions that are executed via the computer interface and do not appear on the display

The third column shows the two-letter ASCII code for use over the computer interface.

The fourth column gives a brief description of the parameters or function.

The last column lists the allowed range of each parameter. Entries outside the allowed range will be changed by the controller to fall within the allowed range.

The individual parameters and functions are described in more detail starting on page 10.

TABLE III-1 INPUT PARAMETERS AND FUNCTIONS

PARAMETER NAME	DISPLAYED AS	ASCII CODE	DESCRIPTION	RANGE*
SPEED	SPEED	SP	MOTOR SPEED IN STEPS/SEC	0.1 TO 1000
ACCELERATION	ACCEL	AC	ACCELERATION IN STEPS/SEC/SEC	200 TO 100 000
BACKLASH	BKLSH	BL	BACKLASH COMPENSATION	-1 999 999 TO 1 999 999
NUMBER OF STEPS	N STEP	NS	NUMBER OF STEPS TO INDEX	0 TO 1 999 999
TARGET A	TARG A	TA	TARGET POSITION A	-1 999 999 TO 1 999 999
TARGET B	TARG B	TB	TARGET POSITION B	-1 999 999 TO 1 999 999
CALIBRATION	CALIB	CA	REDEFINES CURRENT POSITION	-1 999 999 TO 1 999 999
INCREMENT	INCR	IN	INCREMENT OF POSITION PARAMETERS	1.0 TO 1 999 999
SCAN BEGIN	SCNBGN	SB	SCAN BEGIN POSITION	-1 999 999 TO 1 999 999
SCAN END	SCNEND	SE	SCAN END POSITION	-1 999 999 TO 1 999 999
SCAN STEP	SCNSTP	SS	SCAN STEP SIZE	0 TO 1 999 999
SCAN RATE	SCNRT	SR	SCAN RATE FOR CONTINUOUS SCAN	0.1 TO 1000
SCAN WAIT	SCNWT	SW	SCAN WAIT USING STEP AND REPEAT	0.1 TO 9999.9
SCAN NUMBER	SCNNUM	SN	NUMBER OF CONTINUOUS SCANS	0 TO 999 999
SCALE FACTOR	SCLFAC	SF	SCALE FACTOR FOR DISTANCE	0.01 TO 19 999 999
JOG MODE	PB>JOG	PJ	RUN PUSH BUTTONS IN JOG MODE	--
CONTINUOUS MODE	PB>CON	PC	RUN PUSH BUTTONS IN CONTINUOUS MODE	--
INDEX MODE	PB>NDX	PN	RUN PUSH BUTTONS IN INDEX MODE	--
TARGET MODE	PB>TRG	PT	RUN PUSH BUTTONS IN TARGET MODE	--
PROGRAM MODE	PB>PRG	PP	RUN PUSH BUTTONS IN RUN PROGRAM MODE	--
SCAN MODE	PB>SCN	PS	RUN PUSH BUTTONS IN SCAN MODE	--
ABORT	-	AB	ABORT MOTION	--
JOG DOWN	-	JD	MOVE MOTOR -0.5 STEPS (.5 STEP = 1 MICRON)	--
JOG UP	-	JU	MOVE MOTOR +0.5 STEPS (.5 STEP = 1 MICRON)	--
INDEX DOWN	-	ID	INDEX MOTOR -N STEPS	--
INDEX UP	-	IU	INDEX MOTOR +N STEPS	--
GO TO TARG A	-	GA	RUN MOTOR TO TARGET A	--
GO TO TARG B	-	GB	RUN MOTOR TO TARGET B	--
CONTINUOUS DOWN	-	CD	RUN MOTOR CONTINUOUSLY DOWN	--
CONTINUOUS UP	-	CU	RUN MOTOR CONTINUOUSLY UP	--
SCAN DOWN	-	SD	SCAN DOWN	--
SCAN UP	-	SU	SCAN UP	--
INPUT	-	IP	INPUT DIGITAL SIGNALS	(See Table V-1)
OUTPUT	-	OP	OUTPUT DIGITAL SIGNALS	(See Table V-2)
ENTER PROGRAM	-	EP	ENTER PROGRAM TO BE STORED	(See Section IV)
RUN PROGRAM	-	RP	RUN STORED PROGRAM	(See Section IV)
LOOP	-	LP	LOOP (REPEAT) IN STORED PROGRAM	(See Section IV)
WAIT TIME	-	WT	WAIT TIME IN STORED PROGRAM	0.1 TO 9999
WAIT	-	WA	WAIT COMMAND IN STORED PROGRAM	(See Section IV)

* NOTE: RANGES GIVEN ARE FOR SCALE FACTOR AT THE DEFAULT VALUE OF 1.00

A. MOTION CONTROL PARAMETERS AND FUNCTIONS

1. SPEED

The **SPEED** parameter determines the motor's speed in steps per second when operating in the INDEX, TARGET, CONTINUOUS and SCAN modes. The SPEED has no effect in the JOG mode or when operating in the MANUAL mode with the adjust knob. Using the computer interface, the SPEED is set using the ASCII code **SP**. The speed is set to 500 upon Power Up.

2. ACCELERATION

The **ACCELERATION** parameter determines the rate at which the motor's speed is ramped up at the start of a motion and ramped down at the end of motion. The parameter is displayed in steps per second per second. ACCELERATION has no effect in the JOG or MANUAL modes. Using the computer interface, the ACCELERATION is set using the ASCII code **AC**. ACCELERATION is set to 1000 upon Power Up.

3. BACKLASH

The **BACKLASH** parameter can be used to compensate for backlash in mechanical systems. Setting the BACKLASH parameter to a non-zero value causes the controller to finish its motion with a fixed number of steps in a specific direction. For example, if the NUMBER of STEPS is 100.0 and the BACKLASH is -20.0, indexing the motor will cause it to first take 120 steps take -20 steps to end up at 100 steps from where it started. The BACKLASH compensation is not operational when using the rotary knob in the MANUAL mode or when using JOG mode. Using the computer interface, the BACKLASH is set using the ASCII code **BL**.

4. NUMBER OF STEPS

The **NUMBER OF STEPS** parameter determines the number of steps the motor will move when operating in the INDEX mode. Using the computer interface the NUMBER OF STEPS (N STEP) is set using the ASCII code **NS**. N STEP is set to 200 upon Power Up.

5. TARGET A and TARGET B

The **TARGET A** and **TARGET B** parameters determine where the motor will move to in the TARGET mode when the RUN- or RUN+ push buttons, respectively, are pressed. Using the computer interface, the TARGET A and TARGET B parameters are set using the ASCII codes **TA** and **TB** respectively. Upon Power Up, TARGET A is set to 0 and TARGET B is set to 200.

6. CALIBRATION

The **CALIBRATION** parameter redefines the current position but does not move the motor. For example, the motor could be moved until aligned at some reference mark and then the CALIBRATION parameter could be used to redefine this as the 0.0 position. Using the computer interface, the CALIBRATION is set using the ASCII code **CA**. The motor's current position may be read back via the computer interface by issuing the command **CA*** as described in the RS-232C and GPIB/IEEE-488 Computer Interface sections.

7. INCREMENT

The **INCREMENT** parameter determines the amount that distance parameters will be incremented for each pulse from the front panel rotary adjust knob. For example, if the INCREMENT has been set to 10, the NUMBER of STEPS will change in the sequence, 0, 10, 20, 30... etc...as the adjust knob is turned. The parameters affected by INCREMENT are SPEED, ACCELERATION, NUMBER of STEPS, TARGET A, TARGET B, CALIBRATION, BACKLASH, SCAN BEGIN, SCAN END, SCAN STEP AND SCAN RATE. Using the computer interface, the INCREMENT is set using the ASCII code **IN**. Upon Power Up the increment is set to 1.0.

8. SCALE FACTOR

The SCALE FACTOR (SF) can be used to scale the motor steps to more convenient units for a particular application. For example, with a spectrometer or monochromator, the motor steps could be scaled so that the position would read out in nanometers. The SCALE FACTOR is defined as the number of motor half steps per unit of readout. For example, in a system using a motor with 800 half steps per revolution and 5 nm per revolution, the appropriate scale factor would be 800/5 = 160. In this case each half step of the motor would correspond to a .006 nm step on the display. The default value of the SCALE FACTOR is 1, so, that with a stepper mike, the distances are in 1 micron units.

The parameters ACCELERATION, BACKLASH, NUMBER OF STEPS, TARGET A, TARGET B, CALIBRATION, INCREMENT, SCAN BEGIN, SCAN END, SCAN STEP, AND SCAN RATE are all affected by the scale factor. These parameters are stored internally in units of motor half steps and are divided by the scale factor before being output. When being input, these parameters are multiplied by the scale factor to convert them into motor half steps before being stored.

B. MOTION EXECUTION COMMANDS

The JOG MODE, CONTINUOUS MODE, INDEX MODE, TARGET MODE, PROGRAM MODE and SCAN MODE functions switch the mode of operation of the front panel RUN push buttons. These functions can also be executed through the computer interface using the ASCII codes listed in the Table III-1 on page 9.

1. JOG DOWN or JOG UP

From the front panel the JOG DOWN or JOG UP functions are executed by switching to the **JOG MODE** and pressing the **-X RUN** or **+X RUN** push buttons. Holding the buttons down causes the selected motor to run continuously. When executed using the computer interface the JOG- and JOG+ functions are executed by sending the ASCII codes **JD** and **JU** which cause the motor to take one half step in the - or + direction.

2. INDEX DOWN and INDEX UP

From the front panel the INDEX DOWN and INDEX UP functions are executed by switching to the **INDEX MODE** and then pressing the **-X RUN** and **+X RUN** push buttons. Using the computer interface these functions can be executed by sending the ASCII commands **ID** and **IU**. The motor can be stopped before reaching the end of motion by pressing the **SELECT** button or by sending the **ABORT** (AB) command through the computer interface.

3. GO TO TARG A and GO TO TARG B

From the front panel the GO TO TARG A and GO TO TARG B functions are executed by switching to the **TARGET MODE** and then pressing the **-X RUN** and **+X RUN** push buttons. Using the computer interface, these functions can be executed by sending the ASCII commands **GA** and **GB**. The motor can be stopped before reaching the end of motion by pressing the **SELECT** button or by sending the **ABORT** command through the computer interface.

4. RUN CONTINUOUSLY

From the front panel the CONTINUOUS DOWN and CONTINUOUS UP functions are executed by switching to the **CONTINUOUS MODE** and then pressing the **-X RUN** and **+X RUN** push buttons. Using the computer interface, these functions can be executed by sending the ASCII commands **CD** and **CU**. The motor can be stopped by pressing the **SELECT** button or by sending the **ABORT** command through the computer interface.

5. SCAN FUNCTIONS

From the front panel the SCAN DOWN or SCAN UP functions are executed by switching to the **SCAN MODE** and then pressing the **-X RUN** or **+X RUN** push buttons. Using the computer interface these functions can be executed by sending the ASCII commands **SD** or **SU**. The upper left field of the Display will indicate which SCAN direction the STEPPER MOTOR is moving by displaying SCAN DOWN or SCAN UP.

The controller can scan up or down repeatedly between the SCAN BEGIN (SB) and SCAN END (SE) positions. If the value of SCAN STEP (SS) is 0, the position will be scanned continuously at a speed per second determined by the SCAN RATE (SR) parameter. If the value of SCAN STEP is non-zero, the position will be scanned in increments of size SCAN STEP and will wait at each step a time determined by SCAN WAIT (SW) in 1/10 second units. Upon power up the SCAN BEGIN (SB) is 0, SCAN END (SE) is 200, SCAN STEP (SS) is 0, SCAN RATE (SR) is 10 and SCAN WAIT (SW) is 0. When SCAN WAIT is non-zero, the UPPER FIELD of the display will read out "SCAN WAIT TM =" and will show the time decreasing.

The scan will be repeated the number of times specified by the parameter SCAN NUMBER (SN) or until aborted by pressing the front panel SELECT button or sending the ABORT (AB) command through the computer interface. If the SCAN NUMBER is set to 0 the scan will repeat until aborted. Upon power up, the SCAN NUMBER (SN) is 1. The upper right field of the Display will indicate the Number of Scans.

6. RUN STORED PROGRAM

From the front panel the RUN STORED PROGRAM functions are executed by switching to the **STORED PROGRAM MODE** and then pressing the P1, P2, P3 and P4 push buttons. Using the computer interface these functions can be executed by sending the ASCII commands **RP1, RP2, RP3 and RP4**. See Section IV for more details.

IV. STORED PROGRAMS

Four completely separate sequences of remote control commands called **PROG 1, PROG 2, PROG 3** and **PROG 4** can be sent through the computer interface and stored inside the controller. The internally stored programs can then be executed from the front panel by switching to the **STORED PROGRAM MODE** and pressing the **P1, P2, P3** or **P4** push buttons. Using the computer interface these functions can be executed by sending the ASCII commands **RP1, RP2, RP3** or **RP4.**

A. ENTER PROGRAM COMMAND FOR STORING PROGRAMS

The ENTER PROGRAM command **EP1, EP2, EP3** and **EP4** are used to select **PROG 1, PROG2, PROG 3** or **PROG 4** for storing. For example, the command

EP1:NS200:IU

stores in the PROG 1 a program in which the NUMBER OF STEPS (NS) is set to 200 followed by an INDEX UP (IU) command to make the motor take the 200 steps. The commands in the stored program must be preceded by colons as shown. When the ENTER PROGRAM command is used the string of commands will be stored but not executed, i.e., no motion will take place.

B. RUN PROGRAM COMMAND

The RUN command codes **RP1, RP2, RP3** and **RP4** are used to execute **PROG 1, PROG 2, PROG 3** or **PROG 4.** For example, if the program :NS200:IU from the previous example was stored as PROG 1 inside the controller, the command

RP1

would cause the motor to take 200 steps. The front panel P1, P2 P3 and P4 buttons in the STORED PROGRAM MODE perform exactly the same function as the RP1, RP2, RP3 and RP4 commands.

C. REPEATING IN A LOOP

A command or a string of commands in a stored program can be looped or repeated by enclosing them in parentheses, ( ) preceded by the LOOP PROGRAM command. For example, the ASCII commands

EP1:LP10(:IU:ID)

stores a program in which the commands inside the parentheses (INDEX UP, INDEX DOWN) are to be repeated in a LOOP ten times. Positive values of **LP** will cause the corresponding number of LOOPs. A ZERO value of **LP** will cause the program to LOOP INDEFINITELY until the **SELECT** push button is pressed or the **ABORT (AB)** command is sent through the computer interface.

D. WAIT COMMANDS

While entering program commands through the computer interface, a PROGRAMMED DELAY called **WAIT** can be inserted within a string of commands in a stored program by using the command **WA**.

The DURATION of **WAIT TIME** must be ENTERED **PRIOR** to ENTERING THE PROGRAM using the command **WT=*x***. ***x*** can be ANY POSITIVE NUMBER from 0.1 to 9999 and represents **"seconds"** of time, therefore delays from 1/10 SECOND to approximately 2 hours, 46 minutes can be "inserted".

E. READING BACK THE STORED PROGRAM

The stored programs can be read back through the computer interface by sending the commands **P1***, **P2***, **P3*** or **P4***. For example, if the program was entered as:

EP1:NS200:IU

sending **P1*** would cause the controller to respond with:

P1=:NS200:IU

followed by the termination characters.

F. INITIAL STORED PROGRAMS

At power up the initial stored programs are:

PROGRAM 1 :LP0(:IU:ID)
PROGRAM 2 :LP0(:GA:GB)
PROGRAM 3 :IU:ID
PROGRAM 4 :GA:GB

PROGRAMS 1 and **2** move the motor back and forth in an infinite loop. PROG 1 indexes the motor up and down relative to the current position, a number of steps defined by N STEP. PROG 2 moves the motor between two absolute positions defined by TARG A and TARG B. The value of N STEP is initialized (upon Power Up) to 200 and TARG A and TARG B are initialized to 0 and 200 respectively. The initial value of SPEED is 500 steps per second.

PROGRAM 3 is similar to Program 1 because it indexes the motor UP and DOWN relative to the current position, a number of steps defined by N STEP, **except** this is executed only **ONCE.**

PROGRAM 4 is similar to Program 2, **except** this operation is executed only **ONCE.**

NOTE: Stored programs can be **over written** with New Programs, but the New Programs will be **"ERASED"** on Power Down and **DEFAULT** to the stored Programs on Power Up.

V. INPUT, OUTPUT, LIMITS & HOMING

The rear panel DIGITAL I/O connector has 2 TTL digital inputs, 2 TTL digital outputs, 2 inputs for detecting + and - LIMITS and HOME input. These signals are summarized in Table V-1. All inputs are normally pulled high by internal 4.7k Ω resistors shunted with 0.1 μf capacitors and are asserted by pulling them to ground (pins 1 and 14 of the rear panel I/O connector). Table VIII-2 on Page 30 lists the complete pin-out for the 25 pin Rear Panel I/O Connector.

TABLE V-1

INPUT SIGNALS

		HEX OR DECIMAL VALUE		
SIGNAL	PIN NUM	(HEX)	(DECIMAL)	BIT
INPUT 0	17	$1	1	0
INPUT 1	5	$2	2	1
HOME X	20	$4	4	2
**	7	$8	8	3
LIMIT X+	22	$10	16	4
**	9	$20	32	5
**	23	$40	64	6
LIMIT X-	10	$80	128	7

** NOT USED IN MODEL 18705/18706

A. INPUT SIGNALS

The condition of the **INPUT, LIMIT** and **HOME** inputs can be determined using the INPUT readback command **IP***. The binary values table will be appended to IP for each condition that is asserted. For example, if INPUT 0 and INPUT 1 were **both** asserted **low** the command **IP*** would produce the response IP = 3 since 1 + 2 = 3.

In another example, if MOTOR X tripped its - limit switch, then IP* would produce the response IP = 128.

The rear panel LIMIT switch inputs can be used to stop motion past a fixed position. This can be implemented by connecting a normally open switch between ground and the appropriate input pin. The LIMIT inputs are internally connected in parallel with the limit signals on the Rear Panel MOTOR connector. Oriel Stepper Mikes™ have internal end of travel switches.

B. OUTPUT SIGNALS AND COMMANDS

The output signal lines are controlled only by the computer interface using the OUTPUT COMMAND (OP) as listed in Table V-2 below. For example, sending the command **OP=1** would set **OUTPUT 0 HIGH** and pull **OUTPUT 1 LOW.**

TABLE V-2

OUTPUT SIGNALS & COMMANDS

OUTPUT COMMAND (FROM COMPUTER I/F)	OUTPUT 1 (PIN 16 I/O CONN.)	OUTPUT 0 (PIN 3 I/O CONN.)	HEX VALUE	DECIMAL VALUE
(NO COMMAND - POWER UP CONDITION)	H	H	$3	3
OP = 0	L	L	$0	0
OP = 1	L	H	$1	1
OP = 2	H	L	$2	2
OP = 3	H	H	$3	3

H = LOGIC LEVEL 1 (+5V)
L = LOGIC LEVEL 0 (GROUND)

VI. RS-232C SERIAL PORT

A. RS-232C GENERAL

The RS-232C serial port can be used to input remote programming command codes to the stepping motor controller and to readback stored parameter values.

To use the RS-232C port it is necessary to properly connect the data terminal (or other device) as described below in Section B. In addition, it is necessary to set the BAUD rate correctly and to set the CONFIGURATION switch to obtain the echo or non-echo mode and message termination character as discussed in Sections C and D.

B. RS-232C CONNECTIONS

The functions of the pins on the 25 pin RS-232C connector are listed below in Table VI-1. When connecting the stepping motor controller to a data terminal, pin 2 on the stepping motor controller for RECEIVED DATA INPUT (RX) must be connected to the pin on the terminal that corresponds to TRANSMITTED DATA OUTPUT. Similarly pin 3 on the stepping motor controller for TRANSMITTED DATA OUTPUT (TX) must be connected to the pin on the terminal that corresponds to RECEIVED DATA INPUT. If the pins on the terminal do not correspond, it will be necessary to either insert a NULL MODEM adapter between the stepping motor controller and the terminal, or to custom wire a cable so that the signals do correspond.

TABLE VI-1

Pin Number	Stepping Motor Controller Function
2	RECEIVED DATA INPUT (RX)
3	TRANSMITTED DATA OUTPUT (TX)
5	CLEAR TO SEND OUTPUT (CTS)
6	REQUEST TO SEND INPUT (RTS)
7	SIGNAL GROUND

The CLEAR TO SEND OUTPUT signal from the stepping motor controller may be used to signal the terminal that the stepping motor controller is ready to accept data from the terminal.

The REQUEST TO SEND INPUT may be used to indicate to the stepping motor controller that the terminal is ready to accept data from the stepping motor controller. If this pin is unconnected, the stepping motor controller will assume that the terminal is always ready to accept data.

C. RS-232C CONFIGURATION SWITCH SETTINGS

The BAUD rate and other RS-232C parameters are determined by the rear panel CONFIGURATION switch setting at power up.

IMPORTANT: The switch settings are read in by the stepping motor controller only immediately following a power up. To change the mode it is necessary to first change the switch position and subsequently turn the power **off** then **on**.

1. INTERFACE SELECT SWITCH SETTINGS

The active interface is selected by Segment 2 of the rear panel CONFIGURATION SWITCH. If Segment 2 is OFF (OPEN), the RS-232C serial port will be the active interface. If Segment 1 is ON (CLOSED), the GPIB/IEEE-488 bus will be active. NOTE: The GPIB/IEEE-488 interface is only available in the Model 18706.

2. BAUD RATE SWITCH SETTINGS

The BAUD rate is determined by segments, 4,5, and 6 of the rear panel CONFIGURATION switch. Table VI-2 lists the possible settings and the resulting BAUD rates and the approximate character/second rates. The BAUD rate of the stepping motor controller should be set to exactly match the BAUD rate of the device being used to communicate with the stepping motor controller. When more than one BAUD rate is available on the device being interfaced, the highest possible mutually available BAUD rate should be used to maximize data throughput.

TABLE VI-2

BAUD Rate Switch Settings			
Segment 6 5 4	Hex Value	BAUD Rate Exact	Characters/sec (Approximate)
0 0 0	0	RS-232C OFF	-
0 0 1	1	300	30
0 1 0	2	600	60
0 1 1	3	1200	120
1 0 0	4	2400	240
1 0 1	5	4800	480
1 1 0	6	9600	960
1 1 1	7	19200	1920

1 indicates switch in ON (CLOSED) position
0 indicates switch in OFF (OPEN) position

3. RS-232C ECHO SWITCH SETTINGS

When segment 7 of the CONFIGURATION SWITCH is **ON (CLOSED)**, the stepping motor controller is in the echo mode and will re-transmit all characters received through the RS-232C port. This corresponds to the full duplex mode of transmission and is often used with terminals to give feedback that the RS-232C interface is working. When segment 7 is switched **OFF (OPEN)**, the stepping motor controller is switched out of the echo mode and will not re-transmit the characters received.

If the stepping motor controller is in the non-echo mode and the sending terminal is in the full duplex mode, no characters will appear at the terminal. In this case, either switch the stepping motor controller into the echo mode or switch the terminal into the half duplex mode.

If the stepping motor controller is in the echo mode (full duplex mode) but the sending terminal is in the half duplex mode, all transmitted characters will appear double at the terminal. In this case, either switch the stepping motor controller out of the echo mode or switch the terminal into the full duplex mode.

GA
GB

4. RS-232C MESSAGE TERMINATION SWITCH SETTING

When segment 8 is switched **ON (CLOSED)**, received and transmitted messages are terminated with a carriage return (CR) and a line feed (LF). When segment 8 is **OFF (OPEN)**, received and transmitted messages are terminated with only a carriage return (CR).

If the CR + LF position is selected, the stepping motor controller expects messages it receives through the RS-232C port to be terminated with a CR plus a LF. If only a CR is received, the stepping motor controller will wait until a LF is received before processing the message. Messages transmitted by the stepping motor controller will be terminated with a CR plus a LF.

If the CR only position is selected, the stepping motor controller expects messages it receives through the RS-232C post to be terminated with only a CR and will begin processing the message without waiting for a LF. Messages transmitted by the stepping motor controller will be terminated with only a CR.

D. RS-232C PROGRAMMING EXAMPLES

1. SENDING PROGRAMMING COMMANDS THROUGH THE RS-232C SERIAL PORT

The commands to the stepping motor controller are in the form of strings of alphanumeric characters. For example, the command to make the SPEED 200 steps per second is **SP = 200.**

To implement this command using a terminal connected to the RS-232C port, the user would type **SP = 200 (RETURN)** where (RETURN) indicates that the return key is to be pressed to terminate the message.

To implement the same command using a BASIC language computer, a statement such as **10 PRINT "SP = 200"** could be used. Since print statements vary from computer to computer, the detailed formatting might have to be changed slightly.

All the other command codes can be implemented using the RS-232C port in a similar way.

2. READING BACK STORED PARAMETERS THROUGH THE RS-232C SERIAL PORT

Values of parameters stored in the stepping motor controller can be read back through the RS-232C port using the command code *****. For example, if the value of the SPEED stored in the stepping motor controller is 200 steps/sec, typing the command **SP*<RETURN>** will cause the stepping motor controller to send out through the RS-232C port the characters SP+ 200.0 (TERMINATION) indicating that the value of SP is 200. Here (TERMINATION) indicates that the string of characters will be followed by a message termination of either a carriage return (CR) or carriage return plus line feed (LF) as selected by the CONFIGURATION switch.

VII. GPIB/IEEE-488 BUS INTERFACE

A. GPIB/IEEE-488 GENERAL

The GPIB/IEEE-488 bus interface can be used to input remote programming command codes to the stepping motor controller and to read back stored parameter values.

To use the GPIB/IEEE-488 bus interface on the stepping controller it is necessary to connect the stepping motor controller to a computer using a GPIB/IEEE-488 cable and to set the GPIB/IEEE-488 parameters using the rear panel CONFIGURATION switch. Segment 2 of the CONFIGURATION switch selects, either RS-232 or GPIB/IEEE-488, as the active computer interface. If segment 2 is OFF, the RS-232C interface is active and segments 3 to 8 will set the RS-232C parameters as discussed in the section on the RS-232C Interface. If segment 2 is ON, segments 3 to 8 will set the GPIB/IEEE-488 address and termination codes.

B. GPIB/IEEE-488 SETUP & SETTINGS

1. GPIB/IEEE-488 INTERFACE CONNECTOR

The GPIB/IEEE-488 interface connector on the rear of the stepping motor controller accepts the standard GPIB cable with metric threads (black hardware). If the cable being used has English threads (shiny metal hardware) the cable should be replaced or a modification kit should be obtained from the cable supplier. To install the cable:

- Switch off the AC power to the stepping motor controller.
- Orient the cable so that the wider part of the opening is up.
- Insert the GPIB cable in the rear panel receptacle of the stepping motor controller.
- Tighten the locking screws finger tight.

TABLE VII-1
GPIB ADDRESS SETTINGS

DECIMAL	VALUE HEX	OCTAL
0 = GPIB/IEEE-488-INTERFACE-OFF		
01	01	01
02	02	02
03	03	03
04	04	04
05	05	05
06	06	06
07	07	07
08	08	10
09	09	11
10	0A	12
11	0B	13
12	0C	14
13	0D	15
14	0E	16
15	0F	17
16	10	20
17	11	21
18	12	22
19	13	23
21	15	25
22	16	26
23	17	27
24	18	30
25	19	31
26	1A	32
27	1B	33
28	1C	34
29	10	35
30	1E	36

ADDRESS SETTING 31 (DECIMAL) = 1F(HEX)=37(OCTAL) IS RESERVED FOR THE UNLISTEN COMMAND AND WILL NOT FUNCTION AS AN ADDRESS

2. GPIB/IEEE-488 ADDRESS & TERMINATION SETTINGS

The ADDRESS and message termination settings for the GPIB/IEEE-488 interface are set by the rear panel CONFIGURATION switch at Power Up. These settings are read by the Stepping Motor Controller immediately following power on. To make new address or termination settings active the power must be turned OFF and then back ON.

a. GPIB/IEEE-488 ADDRESS

The GPIB/IEEE-488 Address is set by positions 3 to 7 of the CONFIGURATION switch as discussed in the previous section. Table VIII-1 on the previous page shows the possible settings and corresponding address in HEX and OCTAL. Allowed address codes are from 1 to 30 (DECIMAL) with code 31 (DECIMAL) reserved for the UNLISTEN command and not available for an address.

The LISTEN and TALK addresses of the stepping motor controller are the same. For example; if the motor controller's address has been set to 0.1, a statement such as

OUTPUT 701,...

would send data to the motor controller and

INPUT 701,...

would read data from the motor controller. Note that the **OUTPUT** and **INPUT** statements are given here as an example. Other computers may use statements such as **PRINT** and **READ** to perform this function.

When assigning addresses and setting the address switches, care should be taken that the addresses for two devices do not conflict. This can be a particular problem when a given device implicitly has two addresses. For example, some devices have a LISTEN address which matches the rear panel switch and a TALK address which is 1 unit higher. If problems are encountered with addressing the stepping motor controller, or other devices in a system, it is often useful to connect one device at a time and test whether each can be addressed individually. Following this, the devices can be added into the system one at a time while testing the addressability of each. When adding a device interferes with the addressing of another device in the system, the two conflicting devices have been identified and the address setting of one of them should be changed until they no longer conflict.

b. GPIB/IEEE-488 MESSAGE TERMINATION SETTING

The message termination used by the stepping motor controller for the GPIB/IEEE-488 interface is set by position 8 of the CONFIGURATION switch. OFF selects CR for carriage return only and ON selects CR+LF for carriage return followed by a line feed. The message termination setting applies to both received and transmitted messages.

NOTE ON MESSAGE TERMINATORS USED BY GPIB/IEEE-488 COMPUTERS: Most computers terminate their messages with only a CR. Other computers can be configured to terminate with only a CR or with a CR+LF. Be sure that both the computer and the stepping motor controller are set so that their message terminations match.

c. GPIB/IEEE-488 PROGRAMMING - GENERAL

This section gives examples of how to exercise a few of the remote programming features of the stepping motor controller using the GPIB/IEEE-488 interface bus. To make the examples using the GPIB/IEEE-488 bus as explicit as possible some specific assumptions have been made about the system being used. These assumptions are:

- A BASIC language computer such as a Hewlett Packard HP-85 is being used. Thus, in the examples, the OUTPUT command is used to send data over the GPIB/IEEE-488 bus. Many computers use the PRINT command to send data over the GPIB/IEEE-488 bus. If such a computer is being used, PRINT should be substituted for OUTPUT whenever it appears followed by the same string of text.

- The GPIB/IEEE-488 bus interface on the computer is device code 7 and the stepping motor controller has address 01 so that the complete address for the stepping motor controller is 701. If conflicts in the system require another address be used, the ADDRESS and the programming examples should be changed accordingly.

d. SENDING COMMANDS OVER THE GPIB/IEEE-488 BUS

As described in the section on Remote Control Programming, the commands to the stepping motor controller are in the form of strings of alphanumeric characters. For example, the command to set the ACCELERATION to 200 STEPS/SEC/SEC is

```
AC = 200
```

To implement this command using the GPIB/IEEE-488 bus, the user would type

```
OUTPUT 701; "ac = 200"
```

and then execute the line. Alternatively, the line could be embedded in a BASIC language program, e.g.

```
10 OUTPUT 701; "ac = 200"
20 END
```

and then the program could be RUN by the computer to cause the command codes to be sent to the stepping motor controller.

All the other command codes, including the command codes for stored program control can be implemented using the GPIB/IEEE-488 bus in the same way. An example with stored program control is:

```
10 OUTPUT 701; "EP1:NS100:LP10(:IU)"
20 OUTPUT 701; "RP1"
20 END
```

Line 10 stores PROG 1 inside the stepping motor controller.

Line 20 causes the stepping motor controller to RUN PROG 1.

e. READING BACK STORED PARAMETERS ON THE GPIB/IEEE-488 BUS

Values of parameters stored in the stepping motor controller can be read back using the * command. A Program to read back the value of the ACCELERATION is:

```
10 OUTPUT 701; "AC*"
20 ENTER 701; X$
30 DISP X$
40 END
```

Line 10 outputs the command code AC* to the stepping motor controller which instructs it to output the value of the parameter AC over the GPIB/IEEE bus.

Line 20 reads the value of AC from the stepping motor controller into the computer where it is stored as variable X$.

Line 30 displays the value of X$ on the computer screen.

All of the internally stored parameters can be read back this way. The internal stored programs can be read back using Pn* command where n is the program number:

```
10 OUTPUT 701; "P1*"
20 ENTER 701, X$
30 DISP X$
40 END
```

NOTE: When reading back values over the GPIB/IEEE-488 bus, the stepping motor controller functions as a TALKER. However, if the computer fails to read in the value, the stepping motor controller will remain in the TALK mode indefinitely. This condition can be terminated by having the computer read in the variable or by returning the stepping motor controller to the power up conditions.

The following program demonstrates this lock-up condition:

```
10 OUTPUT 701;"AC*"
20 STOP
30 ENTER 701.X
40 DISP X
50 END
```

Line 10 instructs the stepping motor controller to output the value of variable AC.

Line 20 causes the computer to stop. The stepping motor controller will wait until the line 30 is run either by executing a CONTINUE command or by executing a RUN 30 command.

VIII. TECHNICAL REFERENCE

A. REAR PANEL CONFIGURATION SWITCH

The rear panel CONFIGURATION switch is used to set parameters inside the 18705/18706 Stepping Motor Controller at Power Up. The functions of each segment of the switch are listed in Table VIII-1.

TABLE VIII-1
REAR PANEL CONFIGURATION SWITCH

SEG	FUNCTION	OFF = 0	ON = 1
1	HOLDING TORQUE	REDUCED TORQUE	FULL TORQUE
2	ACTIVE INTERFACE	RS-232C	GPIB/IEEE-488
3	INTERFACE FUNCTION	SEE SUB TABLES BELOW	
4	INTERFACE FUNCTION	SEE SUB TABLES BELOW	
5	INTERFACE FUNCTION	SEE SUB TABLES BELOW	
6	INTERFACE FUNCTION	SEE SUB TABLES BELOW	
7	INTERFACE FUNCTION	SEE SUB TABLES BELOW	
8	MESSAGE TERM	CR	CR+LF
SEGMENT 3 - 7 INTERFACE FUNCTIONS IF RS-232C ACTIVE			
SEG	FUNCTION	OFF = 0	ON = 1
3	SPARE	--	--
4	BAUD RATE	SEE SECTION ON RS-232C	
5	BAUD RATE	SEE SECTION ON RS-232C	
6	BAUD RATE	SEE SECTION ON RS-232C	
7	RS-232C ECHO	NO ECHO	ECHO
SEGMENT 3 - 7 INTERFACE FUNCTIONS IF GPIB/IEEE-488 ACTIVE *			
SEG	FUNCTION	OFF = 0	ON = 1
3	ADDR1	SEE SECTION ON GPIB/IEEE-488	
4	ADDR2	SEE SECTION ON GPIB/IEEE-488	
5	ADDR3	SEE SECTION ON GPIB/IEEE-488	
6	ADDR4	SEE SECTION ON GPIB/IEEE-488	
7	ADDR5	SEE SECTION ON GPIB/IEEE-488	

* The GPIB/IEEE-488 interface only on Model 18706

B. REAR PANEL I/O CONNECTOR

The functions of the pins on the rear panel DB-25F DIGITAL I/O connector are listed in Table VIII-2.

TABLE VIII-2 REAR PANEL DIGITAL I/O CONNECTOR

PIN	FUNCTION
1	GROUND OUTPUT
2	Vcc+5 OUTPUT (MAX. CURRENT 100mA)
3	USER OUTPUT 0
4	NC
5	USER INPUT 1
6	NC
7	**
8	NC
9	**
10	LIMIT X - INPUT
11	NC
12	NC
13	NC
14	GROUND OUTPUT
15	Vcc+5 OUTPUT (MAX. CURRENT 100mA)
16	USER OUTPUT 1
17	USER INPUT 0
18	NC
19	NC
20	HOME X INPUT
21	NC
22	LIMIT X + INPUT
23	**
24	NC
25	NC

NOTE: All INPUTS are active low unless otherwise noted.
** = RESERVED (FOR USE ON TWO CHANNEL CONTROLLERS)
NC = NO CONNECTION

C. REAR PANEL MOTOR CONNECTOR

The functions of the pins on the rear panel STEPPER MIKE™ MOTOR CONNECTOR are listed in Table VIII-3.

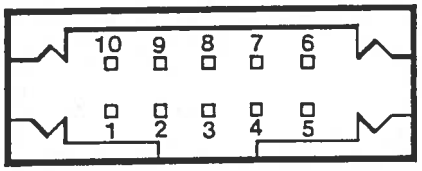


TABLE VIII-3 REAR PANEL STEPPER MIKE™ MOTOR CONNECTOR

PIN NUMBER	FUNCTION
1	+5 VOLTS (100 mA MAXIMUM)
2	REVERSE (-) LIMIT (ACTIVE LOW)
3	FORWARD (+) LIMIT (ACTIVE LOW)
4	+24 VOLTS
5	PHASE A
6	PHASE B
7	PHASE C
8	PHASE D
9	(NO CONNECTION)
10	GROUND (LOGIC COMMON)