Section 8
Appendices

Table of Contents
Section 8 — Appendices

A — Error Messages

The MM4006 controller continually verifies the actions of the motion control system and the operator. When an error is detected, the controller stores it in an error register. To avoid communication and application conflicts, the MM4006 does not automatically report the error. It is the user's responsibility to periodically query the error status, particularly during the development phase of an application.

To better understand error-handling, keep in mind the following points:

- Reading the error with TE or TB clears the error buffer.
- The controller stores only the last error encountered.
- Once an error is detected, it is stored until read or replaced by a new error.
- The error read represents an error that could have happened at any time since the last read.
- For faster communication throughput, use the TE command to read only the error code.
- Use the TB command to read an existing error or to translate an error code.

Error List

The following is a list of all error message codes and their descriptions:

A — Unknown message code.
B — Incorrect axis number.
C — Parameter out of limits.
D — Unauthorized execution.
E — Incorrect I/O channel number.
F — Program number incorrect.
G — Program does not exist.
H — Calculation overflow.
I — Unauthorized command in programming mode.
J — Command authorized only in programming mode.
K — Undefined label.
L — Command not at the beginning of a line.
M — Program is too long.
N — Incorrect label number.
O — Variable number out of range.
P — Number of WE commands does not match the number of open loops.
Q — Unauthorized command.
R — Command cannot be at the beginning of a line.
S — Communication time-out.
T — Error during home search cycle.
U — Failure while accessing the EEPROM.
V — Too long trajectory.
W — Trajectory: to big discontinuity angle.

X — Trajectory: first angle definition error.
Y — Trajectory: Line (x, y) Line expected.
Z — Trajectory: Line (x, y) too big discontinuity.
[— Trajectory: Line (x, θ) or Line (y, θ) impossible.
\ — Trajectory: Arc expected.
] — Trajectory: Arc (r, θ) radius is too small.
^ — Trajectory: Arc (r, θ) radius is too big.
_ — Trajectory: Arc (r, θ) sweep angle is too small.
` — Trajectory: Arc (x, y) circle is too small.
a — Trajectory: Arc (x, y) Circle is impossible.
b — Trajectory: trajectory is empty.
c — Unit not translational or incorrect.
d — Unit not rotationnal or incorrect.
e — Trajectory: Units not translationnal or not identical.
f — sync. pulses generation impossible.
g — mechanical familly name incorrect.
h — Trajectory: execution exceeds physical or logical limits.

Besides the standard screens available on the front panel display, there are a number of error screens that appear only in special error conditions.

Fig. A.1 *— Error screen (English).*

Fig. A.2 *— Error screen (French).*

The screen in Fig. A.1 (English version) or Fig. A.2 (French version) appears if the battery-backed non-volatile memory is corrupted. This will result in a loss of all data in this memory and the controller will request the operator to perform a complete setup procedure on the front panel.

NOTE

Under certain conditions, you may need to erase the non-volatile memory and load the default parameters. This is accomplished simultaneously pressing the minus key "−" and the period key "." on the keypad during the power-up sequence. This will initiate a setup procedure.

The error message shown in Fig. A.3 appears on power-up if the IEEE488 is detected to be malfunctioning. Under this condition, only the RS-232 interface can be used.

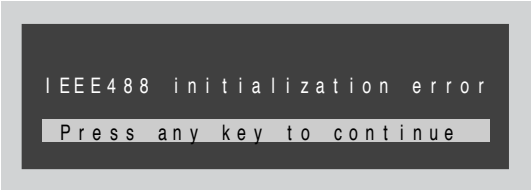


***Fig. A.3** — Error screen, IEEE488.*

The error message in Fig. A.4 appears if one of the function keys or keypad keys are detected being pressed (or stuck) during power-up. The X indicates which key is detected, function keys being labeled from A to D, from left to right.

***Fig. A.4** — Error screen, depressed key during start-up.*

During program creation or modification, the screen shown in Fig. A.5 could appear if the command line being edited exceeds the 110 character limit. The last command entered will be lost but the rest of the line is retained and can be saved. (The XXXX… represents the actual command line being edited).

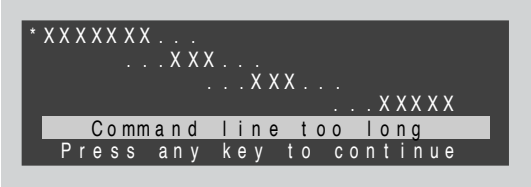


***Fig. A.5** — Error screen, command line too long.*

The second type of error message that is available during program creation or modification is shown in Fig. A.6. It will appear when the non-volatile memory allocated to program storage becomes full. The last line entered (XXXX…) will be lost but the rest of the program is saved.

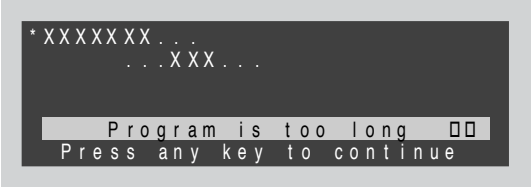


***Fig. A.6** — Error screen, program memory full.*

B — IEEE-488 Link Characteristics

NOTE

In order to meet FCC emission limits for a Class B device, you must use a double shielded IEEE-488 cable. Operating this equipment with a single shielded cable may cause interference to radio and television reception in residential areas.

NOTE

Comply to IEEE Standard Digital Interface for Programmable Instrumentation.

ANSI/IEEE Std. 488 - 1978. This norm is commonly called IEEE-488.

IEEE-488 Functions Supported by MM4006 Controller

Mnemonic	Definition	Support
ATN	Attention	Yes
DCL	Device Clear	Yes
EOI	End or Identify	Yes
EOL	End of Line	Yes
GET	Group Execute Trigger	No
GTL	Go to Local	No
IFC	Interface Clear	Yes
LAD	Listen Address	Yes
LLO	Local Lockout	No
OSA	Other Secondary Address	No
PPC	Parallel Pol Configure	No
PPD	Parallel Poll Disable	No
PPE	Parallel Poll Enable	No
PPU	Parallel Poll Unconfigure	No
REN	Remote Enable	No
SDC	Selected Device Clear	Yes
SPD	Serial Poll Disable	No
SPE	Serial Poll Enable	Yes
SRQ	Service Request	Yes
TAD	Talk Address	Yes
TCT	Take Control	No
UNL	Unlisten	Yes
UNT	Untalk	Yes

IEEE-488 Function Subsets

This controller support the many GPIB function subsets, as listed bellow. Some of the listings described subsets that the controller does not support.

C0 (Controller). The MM4006 can not control other devices.

T5 (Talker). The MM4006 becomes a Talker when the CIC (Controller In Charge) sends its TAD (Talker Address) with the ATN (Attention) line asserted. It ceases to be a talker when the CIC (Controller In Charge) sends another device's TAD (Talker Address) with ATN (Attention) asserted.

L4 (Listener). The MM4006 becomes Listener when the CIC (Controller In Charge) sends its LAD (Listener Address) with the ATN line asserted. The MM4006 does not have Listen Only capability.

SH1 (Source Handshake). The MM4006 can transmit multiline messages accros the GPIB.

AH1 (Acceptor Handshake). The MM4006 can receive multiline messages accros the GPIB.

SR1 (Service Request). The MM4006 asserts SRQ (Serial Request) line to notify the CIC (controller In Charge) when it requires service.

RL0 (Remote/Local). The MM4006 does not support the GTL (Go To Local) and LLO (Local Lock Out) functions.

PP0 (Parralel Poll). The MM4006 has no Parallel Poll capability. It does not respond to the following interface messages: PPC, PPD, PPE and PPU. The MM4006 does not send out a message when the ATN (Attention) and EOI (End or Identify) line are asserted.

DC1 (Device Clear). The MM4006 responds to the DCL (Device Clear) and, when made Listener, the SDC (Selected Device Clear) interface message.

DT0 (Device Trigger). The MM4006 does not support GET (Group Execute Trigger) interface message.

E2 (Electrical). The MM4006 uses tristate buffers to provide optimal high-speed data transfer.

SRQ Using

The NI488.2 User Manual for Windows from National Instruments, in the GPIB Programming Techniques chapter describes the use of Serial Polling as follow (page 7-5):

Serial Polling

You can use serial polling to obtain specific information from GPIB devices when they request service. When the GPIB SRQ line is asserted, it signals the Controller that a service request is pending. The controller must then determine which device asserted the SRQ line and respond accordingly. The most common method for SRQ detection and servicing is serial poll. This section describes how you can set up your application to detect and respond to service requests from GPIB devices.

Service Requests from IEEE-488 Devices

IEEE-488 devices request service from the GPIB Controller by asserting the GPIB SRQ line. When the Controller acknowledge the SRQ, it serial polls each open device on the bus to determine which device requested service. Any device requesting service returns a status byte with bit 6 set and then unasserts the SRQ line. Devices not requesting service return a status byte with bit 6 cleared. Manufacturers of IEEE-488 devices use lower order bits to communicate the reason for the service request or to summarize the state of the device.

Service Requests from IEEE-488.2 Devices

The IEEE-488.2 standard redefined the bit assignments in the status byte. In addition to setting bit 6 when requesting service, IEEE-488.2 devices also use two other bits to specify their status. Bit 4, the Message Availiable Bit (MAV), is set when the device is ready to send previously queried data. Bit 5, the Event Status Bit (ESB), is set if one or more of the enabled IEEE-488.2 events occurs. These events include power-on, user request, command error, execution error, device-dependant error, querry error, request control and operation complete. The device can assert SRQ when ESB or MAV is set, or when a manufacturer-defined condition occurs.

Also on page 7-7, National instruments give an example on how to conduct a serial poll:

SRQ and Serial Polling with NI-488 Device Functions…

The following example illustrates the use of the ibwait and ibrsp functions in a typical SRQ servicing situation when automatic serial polling is enabled.

```
#include "decl.h"

char GetSerialPollResponse (int DeviceHandle)
{
char SerialPollResponse = 0;

ibwait (DeviceHandle, TIMO | RQS);
if (ibsta & RQS)
{
printf ("Device asserted SRQ.\n");
/* Use ibrsp to retrieve the serial poll response. */
ibrsp (DeviceHandle, &SerialPollResponse);
}
return (SerialPollResponse);
}"
```

The MM4006 Controller is an IEEE-488 device in which the SRQ is always enable. It will respond accordingly to the National Instruments example. When the queried data will be ready, the MM4006 will assert the SRQ line and, in the serial poll response bit 6 will be set (Requesting service) and bit 7 (manufacturer-defined) will be set (Message Availiable). After that you can use the ibrd command to retreive the data from the MM4006.

C — Connector Pinouts

Labeling Conventions

All pinout diagrams in this section use the following labeling convention:

AGND ⇒ Analog ground.
DGND ⇒ Digital ground.
N.C. ⇒ Not connected.
UTIL ⇒ Test/utility signal. **DO NOT USE; MAY BE ENERGIZED.**
I ⇒ Input.
O ⇒ Output.

WARNING

The company assumes no responsability for the use of any UTIL labelled pin.

Power Inhibition Connector (9-Pin D-Sub)

This connector is provided for the wiring of one or more remote Emergency Stop switches or Start switches. They will have the same effect as the front panel MOTOR OFF or MOTOR ON buttons.

The minimum rating for the switches should be 50 mA at 24 V and the maximum contact resistance should be less than 100 Ω.

Pin #		Description	
1	—	**N.C.**	
2	—	**UTIL**	Start, switches must be self release push buttons. Wire the switch contacts normally opened. The other side of the switch should be connectd to DGND. If more than one switch is installed, they should be connected in parallela.
3	—	**I**	Emergency Stop, must always be connected to DGND during normal controller operation. An open circuit is equivalent to pressing MOTOR OFF on the front panel. Wire the switch contacts normally closed. If more than one switch is installed, they should be connected in series
4	—	**N.C.**	
5	—	**N.C.**	
6	—	**DGND**	
7	—	**DGND**	
8	—	**DGND**	
9	—	**N.C.**	

Remote Control Connector (15-pin D-Sub)

This connector should only be used with the NEWPORT RC4000 remote Controller.

The connector also provides an Emergency Stop switch input with identical operation to the one in the Power Inhibition connector. If no remote controller are used, the pins must be shorted.

Pin #		Description	
1	—	DGND	
2	—	I	For normal operation connect pins 2 and 3 together.An open circuit is equivalent to pressing the MOTOR OFF on the front panel.
3	—	O	
4	—	UTIL	
5	—	UTIL	
6	—	UTIL	
7	—	UTIL	
8	—	UTIL	
9	—	DGND	
10	—	DGND	
11	—	UTIL	
12	—	UTIL	
13	—	UTIL	
14	—	UTIL	
15	—	UTIL	

WARNING

NEWPORT assumes no responsability for the use of any other Remote Controller.

Auxiliary Connector (25-Pin D-Sub)

This connector is used for the MOTOR OFF indicator, the frequency generator output, the analog inputs and outputs and the synchronisation pulses.

The analog outputs are only available in option.

The logic outputs are open-collector type and are rated for maximum 30 V and 40 mA (Fig. C.2). To drive logic input, they require a pull-up resistor.

The analog inputs and outputs have 12 bits resolution.

The analog inputs are multi-range, software programmable. The available ranges are ±10V, ±5V, 0-10V, 0-5V. See the RA and AM commands for more programmation details. In all cases, analog inputs must be below ±10 V. The impedance of the converter inputs is typically 10kOhms. The maximum input current is ±300µA. The maximum offset error is ±10 LSB, and the maximum gain error is ±10 LSB. The input characteristics of the analog inputs are in Fig. C.1.

The value of 1 LSB depends of the used range:

- 1 LSB is: 20 V/4096 ≈ 5 mV for the ±10 V range.
- 1 LSB is: 10 V/4096 ≈ 2.5 mV for the ±5 V range and 0-10 V range.
- 1 LSB is: 5 V/4096 ≈ 1.25 mV for the 0-5 V range.

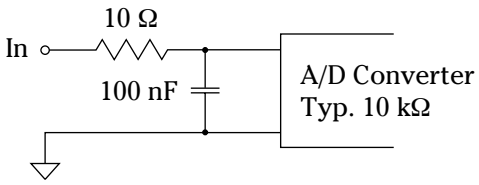


***Fig. C.1** — Equivalent circuit of an analog input.*

The analog outputs range is ±10 V. The maximum offset error is ±200 mV, and the maximum gain error is ±10 LSB. The output setting time is typically 6 µsec. These outputs are voltage outputs (output current less than 1 mA), so to use them properly, they must be connected to an impedance higher than 10 kW. 1 LSB is: 20 V/4096 ≈ 5 mV.

Pin #			Description
1	**—**	**DGND**	
2	**—**	**N.C.**	
3	**—**	**UTIL**	
4	**—**	**UTIL**	
5	**—**	**UTIL**	
6	**—**	**UTIL**	
7	**—**	**UTIL**	
8	**—**	**N.C.**	
9	**—**	**N.C.**	
10	**—**	**O**	A LOW signal indicates that Motor Power is ON.
11	**—**	**O**	Pulse synchronized to one AXIS, see PB, PE, PI and PS commands.
12	**—**	**O**	Pulse synchronized to a trajectory, see NB, NE, NI, NN and NS commands.
13	**—**	**DGND**	
14	**—**	**I**	Analog Input 1.
15	**—**	**I**	Analog Input 2.
16	**—**	**I**	Analog Input 3.

17	**—**	**I**	Analog Input 4.
18	**—**	**DGND**	
19	**—**	**O**	Analog Output 1.
20	**—**	**O**	Analog Output 2.
21	**—**	**O**	Analog Output 3.
22	**—**	**O**	Analog Output 4.
23	**—**	**DGND**	
24	**—**	**O**	Output frequency, defined by the FT command.
25	**—**	**DGND**	

NOTE

Remember that an I/O output bit "set" means that the transistor is conducting, thus appearing to be "low".

GPIO Connector (37-Pin D-Sub)

This connector is dedicated to the digital I/O ports.

All outputs are open-collector type and are rated for maximum 30V and 40mA (Fig. C.2). To drive a logic input, they require a pull-up resistor.

All inputs are optocoupled and are configured as a LED in series with a 1 kΩ resistor connected to the +12 V line (Fig. C.2).

Pin #			Description
1	**—**	**External +12 V/Internal +12 V** [1]	
2	**—**	**+12 V, 25 mA**	
3	**—**	**+5 V,100 mA**	
4	**—**	**I**	Digital port Input 1.
5	**—**	**I**	Digital port Input 2.
6	**—**	**I**	Digital port Input 3.
7	**—**	**I**	Digital port Input 4.
8	**—**	**I**	Digital port Input 5.
9	**—**	**I**	Digital port Input 6.
10	**—**	**I**	Digital port Input 7.
11	**—**	**I**	Digital port Input 8.
12	**—**	**O**	Digital port Output.1.
13	**—**	**O**	Digital port Output.2.
14	**—**	**O**	Digital port Output.3.
15	**—**	**O**	Digital port Output.4.
16	**—**	**O**	Digital port Output.5.
17	**—**	**O**	Digital port Output.6.
18	**—**	**O**	Digital port Output.7.
19	**—**	**O**	Digital port Output.8.

Pin #		Description
20	**—**	**DGND** [2]
21	**—**	**DGND** [2]
22	**—**	**DGND** [2]
23	**—**	**DGND** [2]
24	**—**	**DGND** [2]
25	**—**	**DGND** [2]
26	**—**	**DGND** [2]
27	**—**	**External Ground/Internal Ground** [2]
28	**—**	**DGND** [2]
29	**—**	**DGND** [2]
30	**—**	**DGND**
31	**—**	**DGND**
32	**—**	**DGND**
33	**—**	**DGND**
34	**—**	**DGND**
35	**—**	**DGND**
36	**—**	**DGND**
37	**—**	**DGND**

1) *If optocoupling is not activated, pin #1 outputs +12 VDC.*

If optocoupling is activated, external +12 VDC must be supplied to pin #1. Needs factory service to be activated.

2) *If optocoupling is not activated, pin #20 to pin #29 are tied to the internal DGND.*

If optocoupling is activated, pin 20 to pin 29 are not tied to the internal ground and must be tied to the ground of the external +12 V power supply. Needs factory service to be activated.

Logical Inputs

Parameter	Symbol	Min.	Max.	Units
Low Level Input Voltage	V_{il}	0	5	V
High Level Input Voltage	V_{ih}	11	12	V
Input Current LOW	I_{il}	-5	-10	mA
Pulse Width (1)		1		Servo Cycle
Input low to high	TP_{lh}		10	µsec
Input high to low	Tp_{hl}		10	µsec

1) *Optoisolated logical inputs:*

These inputs works with current driven into the led. If there is no current, input is read as a 1, if there is current through the LED, input is read as a 0.

To drive current through the LED, you can tie the input to ground or drive it by an open collector. This way, the logic level seen at the input, is the same as the one given by the RB command.

To ensure good performances, when current is present its value must be between 5mA and 10 mA.

To be taken into account, one pulse on the input must be larger than one servo-cycle.

Logical Outputs

Parameter	Symbol	Min.	Max.	Units
Low Level Output Voltage	V_{ol}	0	1	V
High Level Output Voltage	V_{oh}		30	V
Output Current LOW	I_{il}		-40	mA
Pulse Width (2)		1		Servo Cycle
Output low to high	TP_{lh}	1		µsec
Output high to low	TP_{hl}	1		µsec

2) *The minimum width on an output pulse cannot be smaller than one servo-cycle.*

To assure good use and performances of the MM4006, respect these maximum ratings.

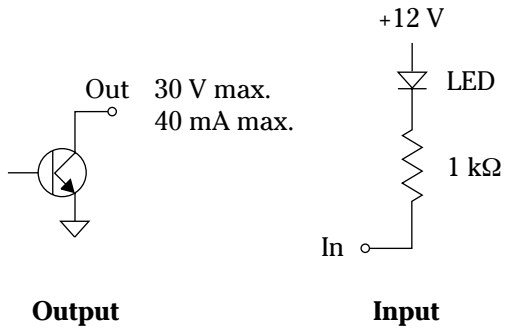


***Fig. C.2 —** Equivalent circuits for the digital input and output ports.*

RS-232C Interface Connector (9-Pin D-Sub)

The RS-232 C interface uses a 9-pin Sub-D connector.

The back panel connector pinout is shown in Fig. C.3.

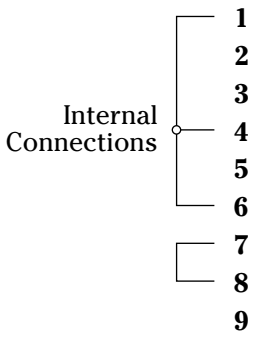


Fig. C.3 — *RS-232C connector pinout.*

RS-232C Interface Cable

The reason some pins are jumpered in the controller as described in Fig. C.3 is to override the hardware handshake when an of-the-shelf cable is used for the RS-232C interface. This guaranties proper communication even when the handshake cannot be controlled from the communication software.

Fig. C.4 shows a simple pin-to-pin cable with 9 conductors.

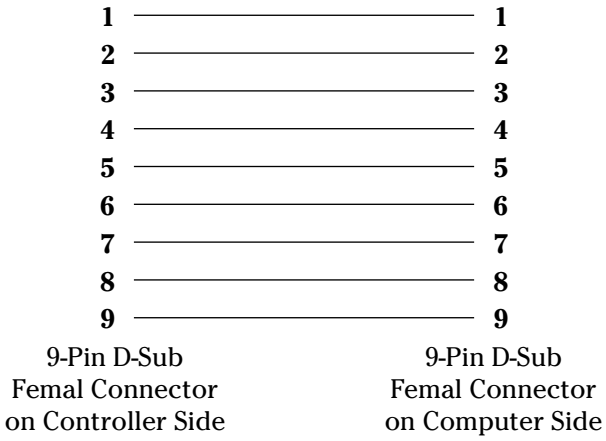


Fig. C.4 — *Conductor, pin-to-pin RS-232C interface cable.*

If you want to use a three conductor cable, you must use a cable configured as in Fig. C.5 to get the same hardware handshake override.

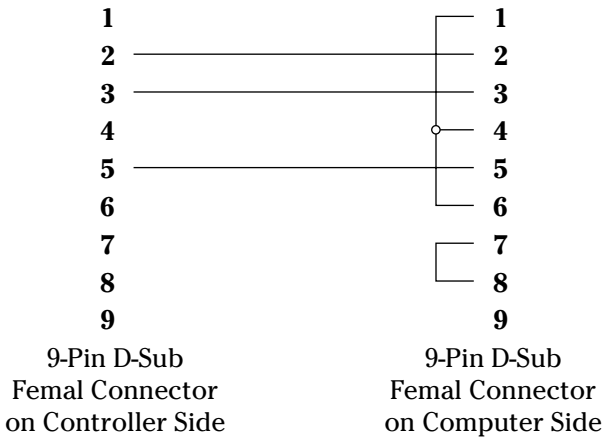


***Fig. C.5** — Conductor RS-232C interface cable.*

If your computer or terminal uses a 25-pin connector for the RS 232C interface, you can use an off-the-shelf 25 to 9-pin adapter and one of the two cables described above.

If you do not wish to add an adapter, you can use an off-the-shelf 9 to 25-pin RS-232C cable or build one like in Fig. C.6.

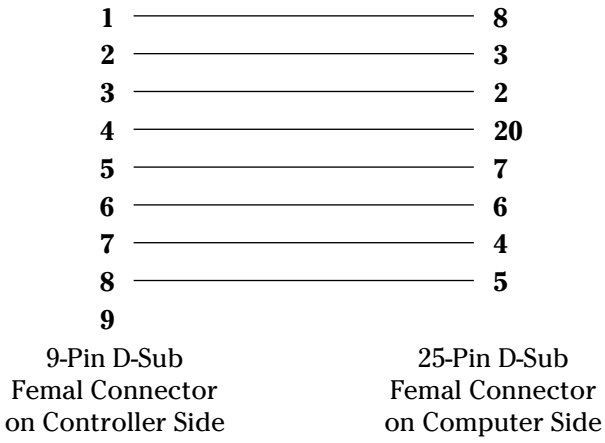


***Fig. C.6** — 9-pin to 25-pin RS-232C interface cable.*

To build a three conductor cable with a 25-pin RS-232C connector, use the wiring diagram in Fig. C.7.

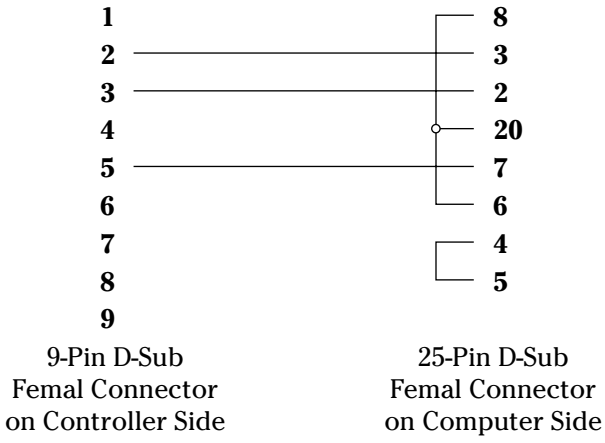


***Fig. C.7** — 3-conductor, 9-pin to 25-pin RS-232C interface cable.*

IEEE488 Interface Connector (24-Pin)

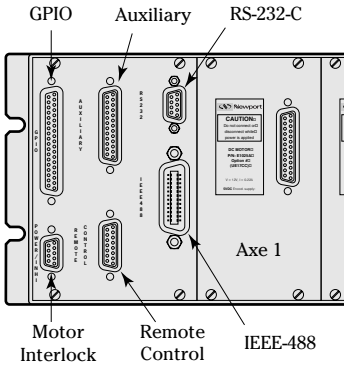
The IEEE488 connector has a standard configuration, shown in Fig. C.8.

	Pin #		
DIO1	1	13	DIO5
DIO2	2	14	DIO6
DIO3	3	15	DIO7
DIO4	4	16	DIO8
EOI	5	17	REN
DAV	6	18	GND
NRFD	7	19	GND
NDAC	8	20	GND
IFC	9	21	GND
SRQ	10	22	GND
ATN	11	23	GND
SHIELD	12	24	SIG. GND

Fig. C.8 — *IEEE488 connector definition.*

RS-485 Interface Connector (5-Pin)

Two identical RS-485 connectors are available. Both are connected in parallel, so you can make the connections on each.



	Pin #	
EARTH	1	1
TX+	2	2
TX-	3	3
RX-	4	4
RX+	5	5

EARTH	1	1
TX+	2	2
TX-	3	3
RX-	4	4
RX+	5	5

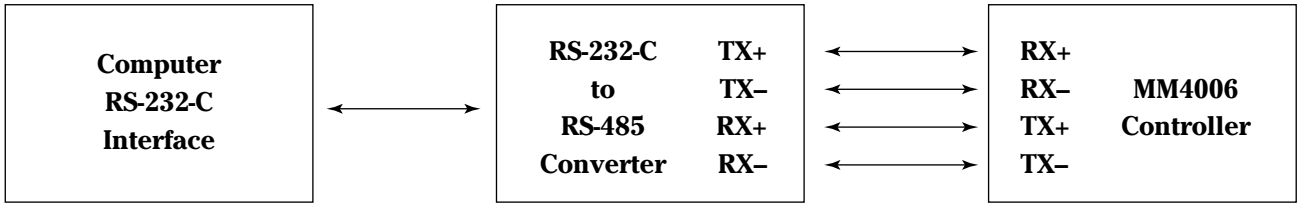
Connecting RS-232-C to a Protocol Converter

To use this communication protocol from a computer equipped with an RS-232-C serial port you must connect a RS-232-C to RS-485 protocol converter. A large choice of those converters can be found from the shelf. The following one are very popular ones and are not a limiting list: ROLINE IC-485S, ROLINE IC-485SI, Burr-Brown LDM485S. Refer to the protocol converter's to properly configure it and check it's connection to a RS-232-C interface. The above figure gives the standard RS-232-C pin-out and interconnection.

Computer RS-232-C Connector				RS-232-C/RS-485 Converter		
25 D-Sub Male	9 D-Sub Male	Pin Name		Pin Name	25 D-Sub Femal	25 D-Sub Femal
3	2	TX	→	RX	2	3
2	3	RX	←	TX	3	2
5	8	RTS	→	CTS	5	4
4	7	CTS	←	RTS	4	5
7	5	GND	—	GND	7	7

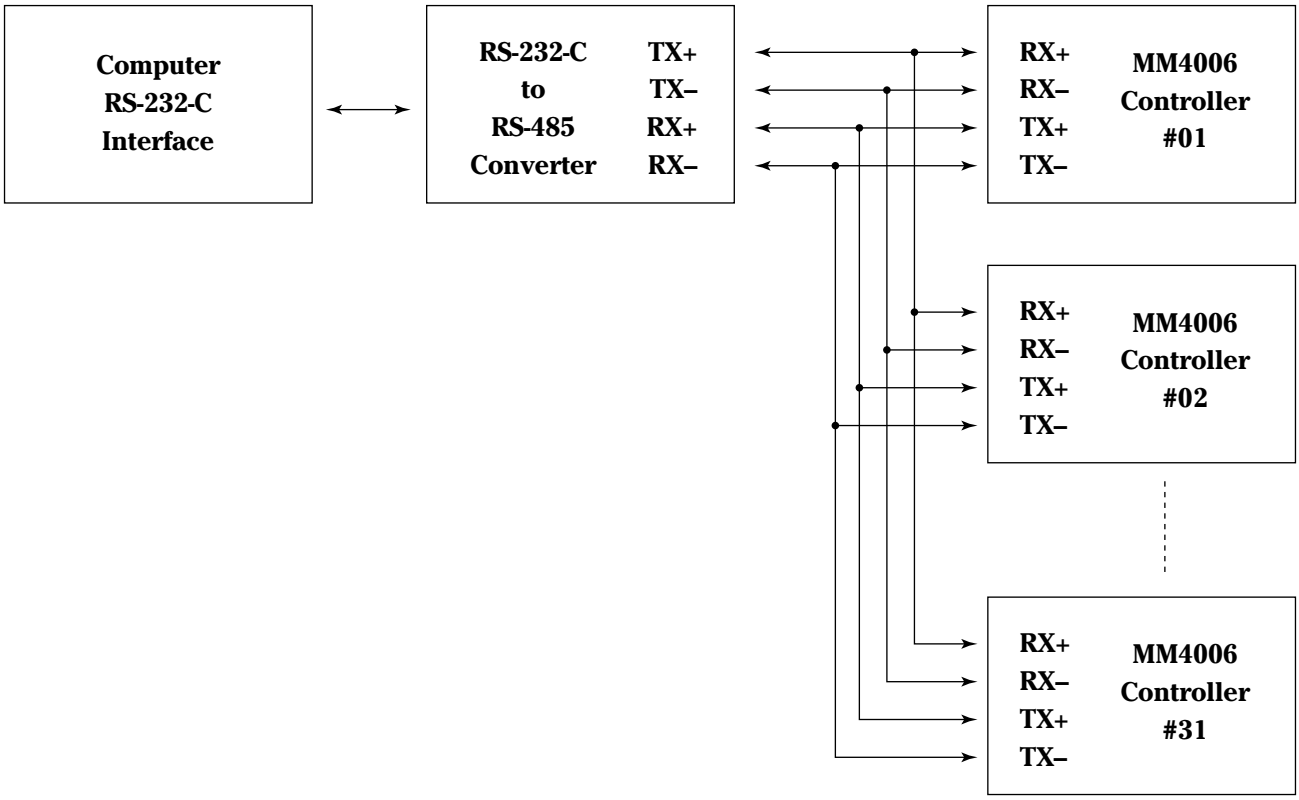
Point-to-Point Four Wires Full Duplex

This is the mode for single computer to a single MM4006 controller in long distance or noisy environment high speed communication. The following figure shows the how to connect your computer, or protocol converter to the MM4006 controller.



Multidrop Four Wires Full Duplex

This feature enables you to connect up to 31 MM4006 controllers to one serial communication port. As a network, each MM4006 controller will have its own address to identify the commands that are sent to it. The following figure shows the how to connect your computer, or protocol converter to several MM4006 controllers.



In this mode of communication each controller must have a single address, from 1 to 31, this address must be different for each controller.

In this network communication, if some controllers are switched off, the others will still continue to work. And, it is no delay in the communication, all controllers receive the commands at the same moment. This is more efficient than daisy chaining, in daisy chaining the computer send command to the first controller who repeat that command to the next one, who repeat to the next one and so on. The daisy chaining puts a lot of traffic on the communication line, introduce repeater delays and will not work if any of the controllers is switched off.

The standard command set of the MM4006 controller is directly usable with the following changes:

- Each command must be initiated with the string address:: to be understood by the right controller. For example for a single computer you send a command like 1OR (home axis 1), this same command will be 1::1OR (controller 1, home axis 1).
- For commands to which the controller has to respond, e.g.: 1TP tell position of axis 1, you should operate with care to avoid any collision on the communication lines. Only one controller should be asked to respond at a time and the computer must wait the reception of the response before interrogating an other controller. So to avoid some of the possible collisions, in this mode commands without axis number to which the controller has to respond will be ignored by the controller. For example commands like TP tell position of all axes will be ignored. To do the same the computer should issue these commands axis per axis and wait the response each time before issuing the next one.

Motor Interface Connector (25-Pin D-Sub)

This connector interfaces to the motion device. Depending on the type of driver and motor, some pins have different meanings. If not otherwise specified, this description is valid for all cases.

	Stepper Motors		DC Motors		
Pin #	UE16PP	UE16PPSC	UE31PP, UE41PP, UE41UP UE62PP, UE63PP	UE16CC, UE17CC, UE31CC, UE33CC, UE35CC, UE404S, UE404S2, UE511S	UE404CC, UE511CC, UE611CC
1	+ Phase 1	+ Phase 1	+ Phase 1	N.C.	+ Tacho Generator
2	N.C.	N.C.	+ Phase 1	N.C.	+ Tacho Generator
3	– Phase 1	– Phase 1	– Phase 1	N.C.	– Tacho Generator
4	N.C.	N.C.	– Phase 1	N.C.	– Tacho Generator
5	+ Phase 2	+ Phase 2	+ Phase 2	+ Motor	+ Motor
6	N.C.	N.C.	+ Phase 2	+ Motor	+ Motor
7	– Phase 2	– Phase 2	– Phase 2	– Motor	– Motor
8	N.C.	N.C.	– Phase 2	– Motor	– Motor
9	N.C.	N.C.	Middle Point (3) Phase 1	N.C.	N.C.
10	N.C.	N.C.	N.C.	N.C.	N.C.
11	N.C.	N.C.	Middle Point (3) Phase 2	N.C.	N.C.
12	N.C.	N.C.	N.C.	N.C.	N.C.
13	Mechanical Zero	Mechanical Zero	Mechanical Zero	Mechanical Zero	Mechanical Zero
14	Shield Ground	Shield Ground	Shield Ground	Shield Ground	Shield Ground
15	Index Pulse I Forcing (Level 1)	Index Pulse I Forcing (Level 1)	Index Pulse I	Index Pulse I (1)	Index Pulse I
16	0 V Logic	0 V Logic	0 V Logic	0 V Logic	0 V Logic
17	+ End-of-Run	N.C.	+ End-of-Run	+ End-of-Run	+ End-of-Run
18	– End-of-Run	N.C.	– End-of-Run	– End-of-Run	– End-of-Run
19	Encoder Phase A	N.C.	Encoder Phase A	Encoder Phase A	Encoder Phase A
20	Encoder Phase B	N.C.	Encoder Phase B	Encoder Phase B	Encoder Phase B
21	+5 V Encoder	N.C.	+5 V Encoder	+5 V Encoder	+5 V Encoder
22	0 V Encoder	N.C.	0 V Encoder	0 V Encoder	0 V Encoder
23	Encoder Phase $\overline{A}$	N.C.	Encoder Phase $\overline{A}$	Encoder Phase $\overline{A}$	Encoder Phase $\overline{A}$
24	Encoder Phase $\overline{B}$	N.C.	Encoder Phase $\overline{B}$	Encoder Phase $\overline{B}$	Encoder Phase $\overline{B}$
25	Index Pulse $\overline{I}$ Forcing (Level 0)	Index Pulse $\overline{I}$ Forcing (Level 0)	Index Pulse $\overline{I}$	Index Pulse $\overline{I}$ (2)	Index Pulse $\overline{I}$

1) For UE16CC and UE17CC motors, the pin #15 is connected: Index Pulse I Forcing (Level 1).

2) For UE16CC and UE17CC motors, the pin #25 is connected: Index Pulse $\overline{I}$ Forcing (Level 0).

3) Except UE41UP motor: N.C.

Pass-Through Board Connector (25-Pin D-Sub)

WARNING

This pass-through board connector takes the place of the motor interface connector only if this axis is connected to an external motor driver.

Pin #		Designation
1	**—**	Ground
2	**—**	5 V Encoder Supply
3	**—**	**I** Mechanical Zero
4	**—**	**I** – End-of-Travel
5	**—**	**I** + End-of-Travel
6	**—**	**I** Driver Fault Signal
7	**—**	**I** Encoder Phase A
8	**—**	**I** Encoder Phase B
9	**—**	**I** Index Pulse I
10	**—**	**O** Pulse Command (1)
11	**—**	**O** Direction Command (1)
12	**—**	**O** ±10 V Analog Output (2)
13	**—**	**N.C.**
14	**—**	0 V Encoder Supply
15	**—**	**O** Driver Inhibition Command
16	**—**	**N.C.**
17	**—**	**N.C.**
18	**—**	**N.C.**
19	**—**	**I** Encoder Phase $\overline{A}$
20	**—**	**I** Encoder Phase $\overline{B}$
21	**—**	**I** Index Pulse $\overline{I}$
22	**—**	0 V logic
23	**—**	0 V logic
24	**—**	**N.C.**
25	**—**	**O** Reference for ±10 V Analog Output

1) Stepper Motor Driver.

2) DC Motor Driver.

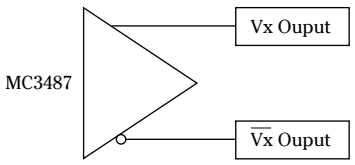


***Fig. C.9** — DiFF. Output Type.*

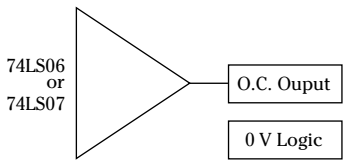


***Fig. C.10** — Open Collector Output Type.*

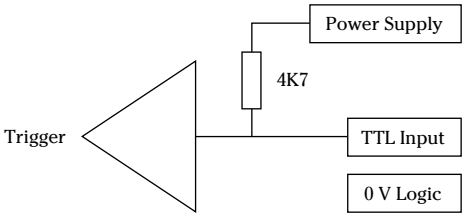


***Fig. C.11** — TTL Input Type.*

D — Motion Program Examples

When learning a new computer language, there is no substitute for actually writing some real programs. The motion controller's command set is a specialized language that needs to be mastered in order to be able to create complex applications. To help you familiarize yourself with MM4006 programming structure and language, this appendix contains a few examples that you can read and copy.

Example 1

The first example is a simple two-axes program that will generate the triangle shown in Fig. D.1.

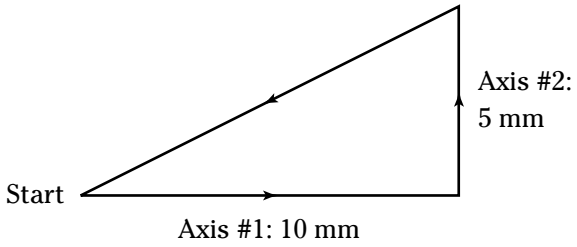
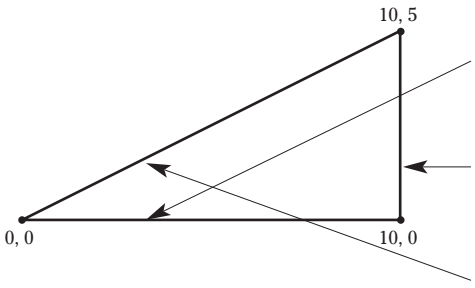


Fig. D.1 *— Triangle Pattern.*

Make sure there is no other program in memory with the same name (number). If you are operating the controller from a remote computer, start by issuing the XX command for that program number. Then, enter the programming mode by using the EP command. If you enter the program from the front panel, ignore these two and the QP commands.



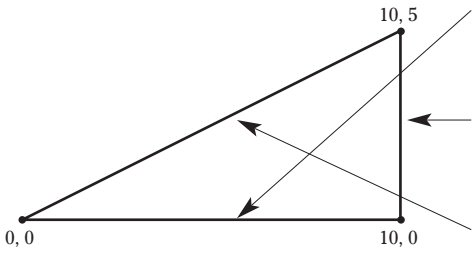
1XX	*Erase program #1, if it exists.*
1EP	*Enter programming mode and store all entries as program #1.*
1VA4	*Set velocity of axis #1 to 4 mm/sec.*
1PA10,1WS	*Move axis #1 to absolute position 10 mm; wait for axis #1 to complete motion.*
2VA4	*Set velocity of axis #2 to 4 mm/sec.*
2PA5,2WS	*Move axis #2 to absolute position 5 mm; wait for axis #2 to complete motion.*
2VA2	*Change velocity of axis #2 to 2 mm/sec.*
1PA0,2PA0	*Move axis #1 to absolute position 0 mm and axis #2 to absolute position 0 mm.*
1QP	*End of program; quit programming mode.*

Example 2

In the previous example, to generate the diagonal line (the third motion segment) both axes must move simultaneously. This is achieved by taking two special precautions: the commands are placed on the same line to insure a good start synchronization and the velocities are modified such that the motions will end in the same time.

But, if you would measure very accurately the precision of this diagonal line, you would notice some errors due to imperfect start synchronization and an incorrect acceleration ratio. In other words, we achieved this dual-axes motion with two independent single-axis motions.

To eliminate these motion errors, we need to use the axes synchronization (linear interpolation) feature. The improved program will have the following listing:



2XX	*Erase program #2, if it exists.*
2EP	*Enter programming mode and store all entries as program #2.*
1VA4	*Set velocity of axis #1 to 4 mm/sec.*
1PA10,1WS	*Move axis #1 to absolute position 10 mm; wait for axis #1 to complete motion.*
2VA4	*Set velocity of axis #1 to 4 mm/sec.*
2PA5,2WS	*Move axis #2 to absolute position 5 mm; wait for axis #2 to complete motion.*
1SY1,2SY1	*Declare axes #1 and #2 synchronized.*
1PA0,2PA0,SE,WS	*Set axis #1 destination to 0 mm and axis #2 destination to 0 mm; start synchronous motion; wait for motion to complete.*
1SY0,2SY0	*Declare axes #1 and #2 non-synchronized.*
2QP	*End of program #2; quit programming mode.*

Notice that there is no need to set the velocities before the synchronized (interpolated) motion. The controller automatically calculates them to get the best accuracy possible, without exceeding the pre-set individual velocities.

Also, when finished with an interpolated motion, always return the axes to the non-synchronized mode.

Example 3

The MM4006 does not offer true circular interpolation but in many cases less demanding applications can be successfully implemented.

Take the example of dispensing glue on the pattern shown in Fig. D.2.

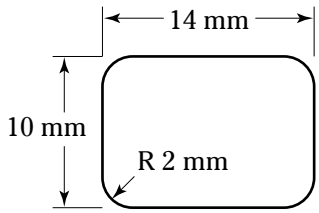


Fig. D.2 — *Glue Dispensing Pattern.*

Notice that there is no need to set the velocities before the synchronized (interpolated) motion. The controller automatically calculates them to get the best accuracy possible, without exceeding the pre-set individual velocities.

Also, when finished with an interpolated motion, always return the axes to the non-synchronized mode.

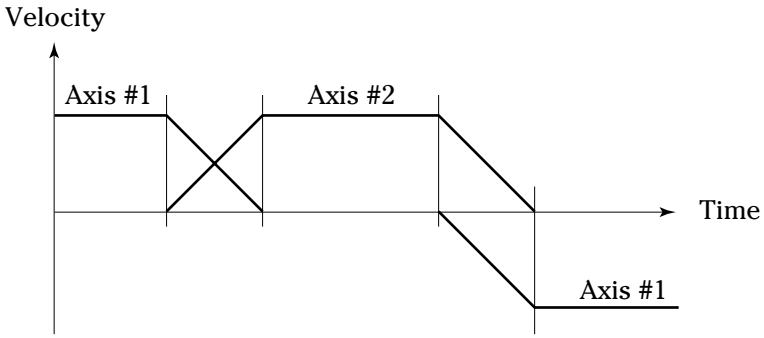


Fig. D.3 — *Overlapping Axis Acceleration/Deceleration.*

Assuming that the desired velocity is 4 mm/sec, we need to calculate the acceleration and the positions where one axis starts decelerating and the other accelerating.

We know that an axis must travel 2 mm before reaching a velocity of 4 mm/sec.

$$\text{Velocity} = \frac{\Delta\ \text{Distance}}{\text{Time}} \Rightarrow \text{Time} = \frac{\Delta\ \text{Distance}}{\text{Velocity}}$$

$$\text{Acceleration} = \frac{\Delta\ \text{Velocity}}{\text{Time}} = \Delta\ \text{Velocity} \bullet \frac{\text{Velocity}}{\Delta\ \text{Distance}}$$

Since the velocity starts from zero, Δ Velocity = Velocity.

$$\text{Acceleration} = \frac{\text{Velocity}^2}{\Delta\ \text{Distance}} = \frac{42}{2} = 8\ \text{mm/sec}^2$$

Before starting to write the actual program, we need to consider one more thing: to assure a good result, the glue must start being dispensed while the motion is in progress. Thus, we have to start the motion first and then turn on the dispenser.

The motion we decide to perform is shown in Fig. D.4.

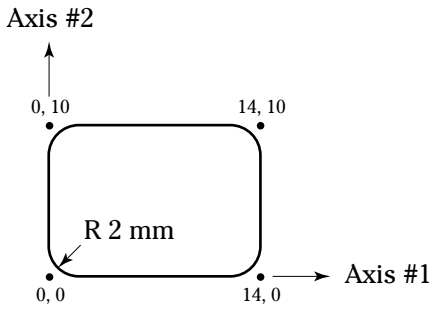
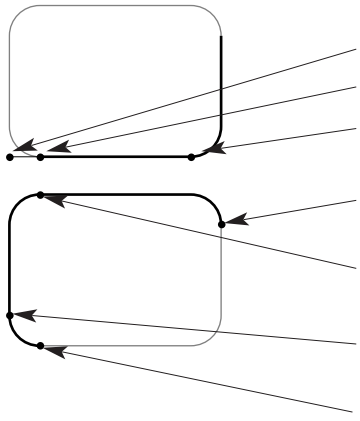


Fig. D.4 *— Desired Motion Result.*

The program will have the following listing:

3XX	*Erase program #3, if it exists.*
3EP	*Enter programming mode and store all entries as program #3.*
CB	*Clear all output I/O bits; set all bits to zero.*
1PA0,2PA0,WS	*Move axes #1 and #2 to absolute position 0 mm; wait for all axes to complete motion.*
1VA4,2VA4	*Set velocity of axes #1 and #2 to 4 mm/sec.*
1AC8,2AC8	*Set acceleration of axes #1 and #2 to 8 mm/s².*
1PA14	*Move axis #1 to absolute position 14 mm.*
1WP2,3SB	*Wait for axis #1 to reach position 2 mm; set bit #3.*
1WP12,2PA10	*Wait for axis #1 to reach position 12 mm; start axis #2 and move to position 10 mm.*
2WP8,1PA0	*Wait for axis #2 to reach position 8 mm; start axis #1 and move to position 0 mm.*
1WP2,2PA0	*Wait for axis #1 to reach position 2 mm; start axis #2 and move to position 0 mm.*
2WP2,1PA4	*Wait for axis #2 to reach position 2 mm; start axis #1 and move to position 4 mm.*
1WP2,3CB	*Wait for axis #1 to reach position 2 mm; clear bit #3.*
3QP	*End of program #2; quit programming mode.*



Example 4

Lets assume we want to write the  from the Newport logo. We have a X-Y table and a 0.5 mm plotter pen (or a laser beam) controlled by a TTL line. One possibility is to scan the symbol with a 0.5 mm spacing and fill it in with 0.5 mm lines. The result will be similar to Fig. D.5.

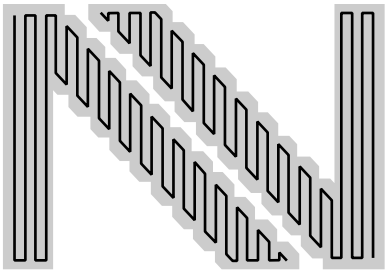


Fig. D.5.

The solid lines show the actual pen trajectory.

Next, we need to select a coordinate system. For simplicity, lets make the lower left corner of the trajectory the origin (zero), as shown in Fig. D.6.

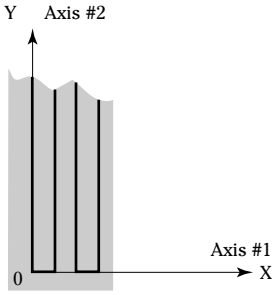
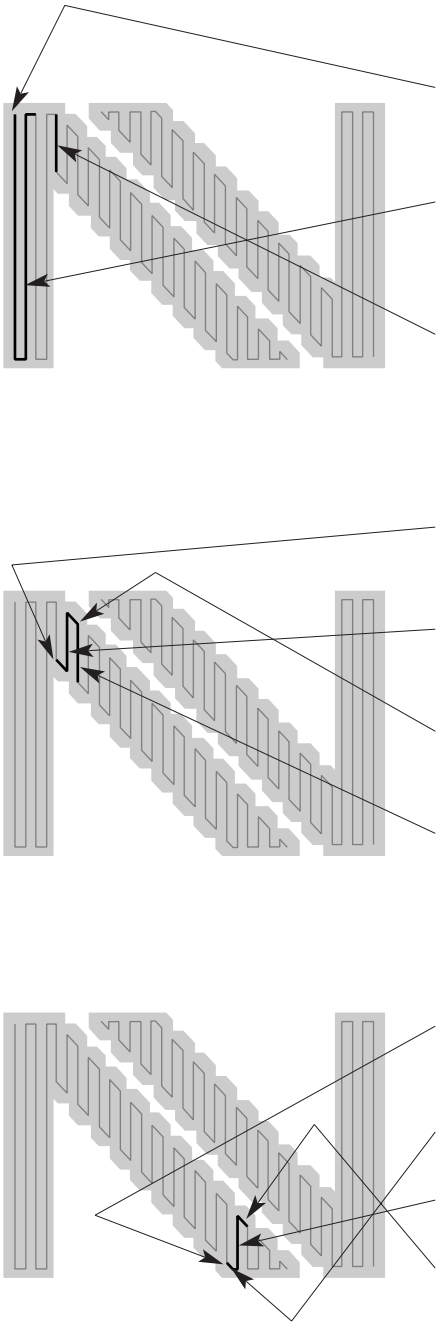


Fig. D.6.

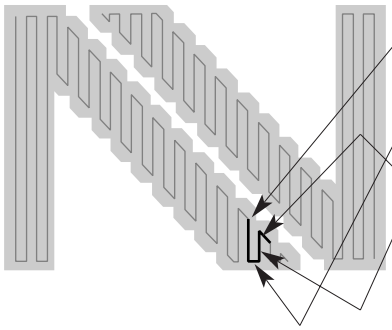
We decide to make the symbol 13 mm high and 17.5 mm wide. But, using a pen with a 0.5 mm wide tip, the actual trajectory must be shrunk to 12.5 ¥ 17 mm. To control the pen up and down we will use bit #8 of the I/O output port, where logic high means pen down.

First, we need to make sure that there is no other program in memory with the same name (number). We do this by listing the program number selected or just by erasing it with the XX command.

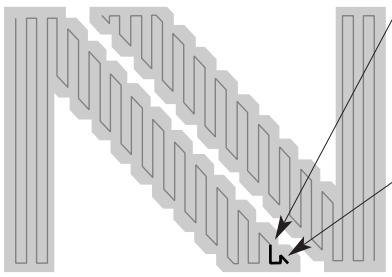
Assuming that this program is being edited on a computer and then downloaded to the controller, we also need to send the commands to enter and terminate the programming mode.



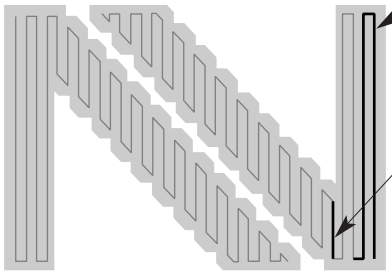
```
4XX                          Erase program #4, if it exists.
4EP                          Store all following entries as program #1.
CB                           Clear all output I/O bits; set all bits to zero.
1PA0,2PA12.5,WS              Move axis #1 to 0 mm and axis #2 to 12.5 mm, wait
                             for all motion to complete.
8SB                          Set I/O bit #8 high; this brings the pen down.
2PR-12.5,WS,1PR0.5,WS,2PR12.5,WS,1PR0.5,WS,RP2 Make four relative
                             motions by sequentially incrementing axis #1 and
                             #2; wait for each motion to stop; repeat the cycle
                             (command line) two times.
2PA10,WS                     Move axis #2 to 10 mm and wait for motion complete.
1YS0                         Initialize variable #1; set its value to zero.
1SY1,2SY1                    Declare axes #1 and #2 synchronized.
1WL8                         Start a while loop; repeat the following commands
                             while variable #1 is less than 8.
1PR0.5,2PR-0.596,SE,WS       Set relative destination of axis #1 at 0.5 mm and of
                             axis #2 at -0.596 mm away from current position; start
                             synchronous motion; wait for motion to complete.
2PR3,SE,WS                   Set relative destination of axis #2 3 mm away from
                             current position; start motion on the synchronized
                             axis; wait for motion to complete.
1PR0.5,2PR-0.596,SE,WS       Set relative destination of axis #1 at 0.5 mm and of
                             axis #2 at -0.596 mm away from current position; start
                             synchronous motion; wait for motion to complete.
2PR-3,SE,WS                  Set relative destination of axis #2 -3 mm away from
                             current position; start motion on the synchronized
                             axis; wait for motion to complete.
1YA1                         Increment variable #1 by 1.
WE                           End while loop.
1PA10.35,2PA0,SE,WE          Set destination of axis #1 to 10.35 mm and of axis
                             #2 to 0 mm; start synchronous motion; wait for
                             motion to complete.
1PA10.5,SE,WS                Set destination of axis #1 to 10.5 mm; start synchro-
                             nized axis; wait for motion to complete.
2PA2.979,SE,WS               Set destination of axis #2 to 2.979 mm; start synchro-
                             nized axis; wait for motion to complete.
1PR0.5,2PR-0.596,SE,WS       set relative destination of axis #1 at 0.5 mm and of
                             axis #2 at -0.596 mm away from current position;
                             start motion; wait for motion to complete.
```



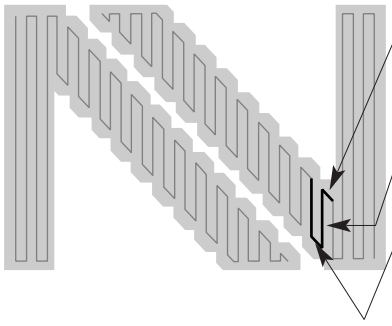
2PA0,SE,WS	*Set destination of axis #2 to 0 mm; start synchronized axis; wait for motion to complete.*
1PA11.5,SE,WS	*Set destination of axis #1 to 11.5 mm; start synchronized axis; wait for motion to complete.*
2PA1.788,SE,WS	*set destination of axis #2 to 1.788 mm; start synchronized axis; wait for motion to complete.*
1PR0.5,2PR-0.596,SE,WS	*set relative destination of axis #1 at 0.5 mm and of axis #2 at -0.596 mm away from current position; start synchronous motion; wait for motion end.*



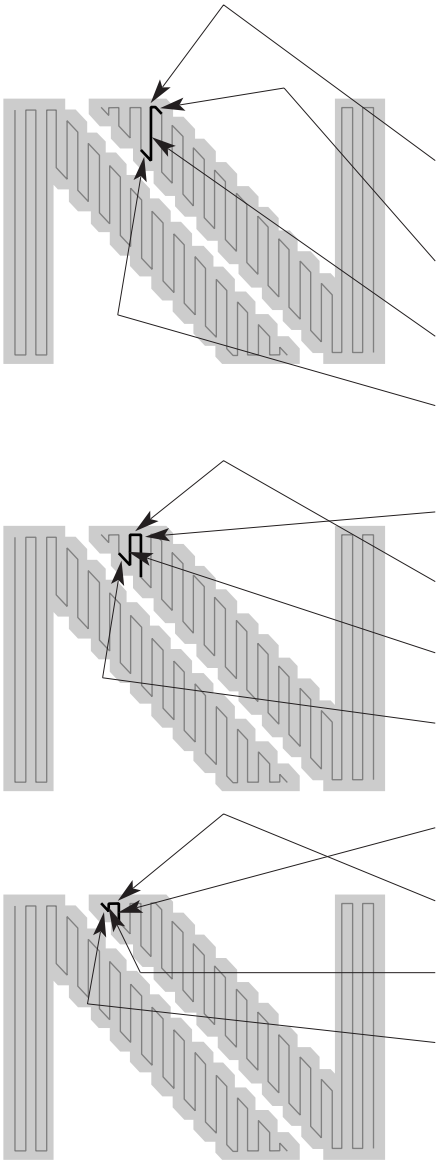
2PA0,SE,WS	*set destination of axis #2 to 0 mm; start synchronized axis; wait for motion to complete.*
1PA12.5,SE,WS	*Set destination of axis #1 to 12.5 mm; start synchronized axis; wait for motion to complete.*
2PA0.596,SE,WS	*Set destination of axis #2 to 0.596 mm; start synchronized axis; wait for motion to complete.*
1PA13,2PA0,SE,WS	*Set destination of axis #1 to 13 mm and of axis #2 to 0 mm; start motion; wait for motion to complete.*
1SY0, 2SY0	*Declare axes #1 and #2 non-synchronized.*
8CB	*Set I/O bit #8 low; this will lift the pen up.*
1PA17,WS	*Move axis #1 to 17 mm; start synchronized axis; wait for motion to complete.*
8SB	*Set I/O bit #8 high; this brings the pen down.*



2PR12.5,WS,1PR-0.5,WS,2PR-12.5,WS,1PR-0.5,WS,RP2	*Make four relative motions by sequentially incrementing axis #1 and #2; wait for each motion to stop; repeat the cycle (command line) two times.*
2PA2.5,WS	*Move axis #2 to 2.5 mm and wait for motion complete.*
1YS0	*Initialize variable #1; set its value to zero.*
1SY1, 2SY1	*Declare axes #1 and #2 synchronized.*
1WL8	*Start a wile loop; repeat the following commands while variable #1 is less than 8.*



1PR-0.5,2PR0.596,SE,WS	*Set relative destination of axis #1 at -0.5 mm and of axis #2 at 0.596 mm away from current position; start motion; wait for motion to complete.*
2PR-3,SE,WS	*Set relative destination of axis #2 -3 mm away from current position; start motion on the synchronized axis; wait for motion to complete.*
1PR-0.5,2PR0.596,SE,WS	*Set relative destination of axis #1 at -0.5 mm and of axis #2 at 0.596 mm away from current position; start synchronous motion; wait for motion to complete.*



2PR3,SE,WS	*Set relative destination of axis #2 3 mm away from current position; start motion on the synchronized axis; wait for motion to complete.*
1YA1	*Increment variable #1 by 1.*
WE	*End while loop.*
1PA6.65,2PA12.5,SE,WS	*Set destination of axis #1 to 6.65 mm and of axis #2 to 12.5 mm; start synchronous motion; wait for motion to complete.*
1PA6.5, SE,WS	*Set destination of axis #1 to 6.5 mm; start synchronized axis; wait for motion to complete.*
2PA9.521,SE,WS	*Set destination of axis #2 to 9.521 mm; start synchronized axis; wait for motion to complete.*
1PR-0.5,2PR0.596,SE,WS	*Set relative destination of axis #1 at -0.5 mm and of axis #2 at 0.596 mm away from current position; start synchronous motion; wait for motion to complete.*
2PA12.5,SE,WS	*Set destination of axis #2 to 12.5 mm; start synchronized axis; wait for motion to complete.*
1PA5.5,SE,WS	*Set destination of axis #1 to 5.5 mm; start synchronized axis; wait for motion to complete.*
2PA10.712,SE,WS	*Set destination of axis #2 to 10.712 mm; start synchronized axis; wait for motion to complete.*
1PR-0,5,2PR0.596,SE,WS	*Set relative destination of axis #1 at -0.5 mm and of axis #2 at 0.596 mm away from current position; start synchronous motion; wait for motion to complete.*
2PA12.5,SE,WS	*Set destination of axis #2 to 12.5 mm; start synchronized axis; wait for motion to complete.*
1PA4.5,SE,WS	*Set destination of axis #1 to 4.5 mm; start synchronized axis; wait for motion to complete.*
2PA11.904,SE,WS	*Set destination of axis #2 to 11.904 mm; start synchronized axis; wait for motion to complete.*
1PA4,2PA12.5,SE,WS	*Set destination of axis #1 to 4 mm and of axis #2 to 12.5 mm; start synchronous motion; wait for motion to complete.*
8CB	*Set I/O bit #8 low; this will lift the pen up.*
1SY0,2SY0	*Declare axes #1 and #2 non-synchronized.*
4QP	*End of program; quit programing mode.*

E — Troubleshooting Guide

Remember that there are no user-serviceable parts or adjustments to be made inside the controller or any other component. Contact Newport for any repair or other hardware corrective action.

Most of the time, a blown fuse or an error reported by the controller is the result of a more serious problem. Fixing the problem should include not only correcting the effect (blown fuse, limit switch. etc.) but also the cause of the failure. Analyze the problem carefully to avoid repeating it in the future. The following is a list of the most probable problems and their corrective actions. Use it as a reference but keep in mind that in most cases a perceived error is usually an operator error or has a simple solution.

Problem	Cause	Corrective Action
Stand-By red LED does not come on	Rear power switch turned off	Turn on the main power switch located on the power entry module in the rear of the unit.
	No electrical power	Verify with an adequate tester or another electrical device (lamp, etc.) that the power is present in the outlet. If not, contact an electrician to correct the problem.
	Unplugged power cord	Plug the power cord in the appropriate outlet. Observe all caution notes and procedures described in the System Setup section.
	Blown fuse	Replace the line fuse as described in the System Setup section. Beware that the fuse blows only when a serious problem arises. If fuse blows again, contact Newport for service.
A physically present axis is declared unconnected	Bad connection	Turn power off and verify the motion device cable connection.
	Bad component	Turn power off and swap motor cable with another axis (if cables are identical) to locate the problem. Contact Newport for cable replacement or motion device service.

Problem	Cause	Corrective Action
The MOTOR ON green LED does not stay on	Limit switch tripped	Execute a home search routine or move the axis in manual mode (jog). Make sure that the limit switch was not tripped by a serious problem.
	Executive following error	Verify that teh motion device installed is connected to the proper driver card.
		Verify that all setup parameters correspond to the actual motion device installed.
		Verify that the load specifications for the motion device are not being exceeded.
The axis does not move	Incorrect connection	Verify that the motion device is connected to the correct driver card, as specified by the labels.
	Incorrect parameters	Verify that all relevant parameters (PID, velocity, etc.) are set properly.
System performance below expectations	Incorrect connection	Verify that the motion device is connected to the correct driver card, as specified by the labels.
	Incorrect parameters	Verify that all relevant parameters (PID, velocity, etc.) are set properly.
Motor excessively hot	Incorrect connection	Verify that the motion device is connected to the correct driver card, as specified by the labels.
Move command not executed	Software travel limit	The software travel limit in the specified direction was reached. If limits are set correctly, do not try to move past them.
	Incorrect parameters	Verify that all relevant parameters (PID, velocity, etc.) are set properly.
Home search not completed	Time-out too short	Verify the home search time-out is set correctly. If the home search velocity was changed, the time-out must be increased.
	Faultry origin or index signals	Carefully observe and record the motion sequence by watching the manual knob rotation, if available. With the information collected, call Newport for assistance.

Problem	Cause	Corrective Action
No remote communication	Wrong line	Make sure that the computer and the controller use the same line terminator.
	Wrong communication port	Verify that the controller is set to communication on the left port RS-232-C or IEEE-488.
	Wrong communication parameters	Verify that all communication parameters match between the computer and the controller.

NOTE

Many other type of problems are detected by the controller and reported on the display and/or in the error register. Consult appendix A for a complete list and description.

F — Decimal/ASCII/Binary Conversion Table

Some of the status reporting commands return an ASCII character that must be converted to binary. To aid with the conversion process, the following table converts all character used and some other common ASCII symbols to decimal and binary. To also help in working with the I/O port related commands, the table is extended to a full byte, all 256 values.

Number (decimal)	ASCII Code	Binary Code	Number (decimal)	ASCII Code	Binary Code
0	*null*	00000000	36	**$**	00100100
1	*soh*	00000001	37	**%**	00100101
2	*stx*	00000010	38	**&**	00100110
3	*etx*	00000011	39	**‘**	00100111
4	*eot*	00000100	40	**(**	00101000
5	*enq*	00000101	41	**)**	00101001
6	*ack*	00000110	42	*****	00101010
7	*bel*	00000111	43	**+**	00101011
8	*bs*	00001000	44	**,**	00101100
9	*tab*	00001001	45	**-**	00101101
10	*lf*	00001010	46	**.**	00101110
11	*vt*	00001011	47	**/**	00101111
12	*ff*	00001100	48	**0**	00110000
13	*cr*	00001101	49	**1**	00110001
14	*so*	00001110	50	**2**	00110010
15	*si*	00001111	51	**3**	00110011
16	*dle*	00010000	52	**4**	00110100
17	*dc1*	00010001	53	**5**	00110101
18	*dc2*	00010010	54	**6**	00110110
19	*dc3*	00010011	55	**7**	00110111
20	*dc4*	00010100	56	**8**	00111000
21	*nak*	00010101	57	**9**	00111001
22	*syn*	00010110	58	**:**	00111010
23	*etb*	00010111	59	**;**	00111011
24	*can*	00011000	60	**<**	00111100
25	*em*	00011001	61	**=**	00111101
26	*eof*	00011010	62	**>**	00111110
27	*esc*	00011011	63	**?**	00111111
28	*fs*	00011100	64	**@**	01000000
29	*gs*	00011101	65	**A**	01000001
30	*rs*	00011110	66	**B**	01000010
31	*us*	00011111	67	**C**	01000011
32	*space*	00100000	68	**D**	01000100
33	**!**	00100001	69	**E**	01000101
34	**“**	00100010	70	**F**	01000110
35	**#**	00100011	71	**G**	01000111

Number (decimal)	ASCII Code	Binary Code	Number (decimal)	ASCII Code	Binary Code
72	**H**	01001000	120	**x**	01111000
73	**I**	01001001	121	**y**	01111001
74	**J**	01001010	122	**z**	01111010
75	**K**	01001011	123	**{**	01111011
76	**L**	01001100	124	**|**	01111100
77	**M**	01001101	125	**}**	01111101
78	**N**	01001110	126	**~**	01111110
79	**O**	01001111	127		01111111
80	**P**	01010000	128		10000000
81	**Q**	01010001	129		10000001
82	**R**	01010010	130		10000010
83	**S**	01010011	131		10000011
84	**T**	01010100	132		10000100
85	**U**	01010101	133		10000101
86	**V**	01010110	134		10000110
87	**W**	01010111	135		10000111
88	**X**	01011000	136		10001000
89	**Y**	01011001	137		10001001
90	**Z**	01011010	138		10001010
91	**[**	01011011	139		10001011
92	\	01011100	140		10001100
93	**]**	01011101	141		10001101
94	^	01011110	142		10001110
95	_	01011111	143		10001111
96	‘	01100000	144		10010000
97	**a**	01100001	145		10010001
98	**b**	01100010	146		10010010
99	**c**	01100011	147		10010011
100	**d**	01100100	148		10010100
101	**e**	01100101	149		10010101
102	**f**	01100110	150		10010110
103	**g**	01100111	151		10010111
104	**h**	01101000	152		10011000
105	**i**	01101001	153		10011001
106	**j**	01101010	154		10011010
107	**k**	01101011	155		10011011
108	**l**	01101100	156		10011100
109	**m**	01101101	157		10011101
110	**n**	01101110	158		10011110
111	**o**	01101111	159		10011111
112	**p**	01110000	160		10100000
113	**q**	01110001	161		10100001
114	**r**	01110010	162		10100010
115	**s**	01110011	163		10100011
116	**t**	01110100	164		10100100
117	**u**	01110101	165		10100101
118	**v**	01110110	166		10100110
119	**w**	01110111	167		10100111

Number (decimal)	ASCII Code	Binary Code	Number (decimal)	ASCII Code	Binary Code
168		10101000	212		11010100
169		10101001	213		11010101
170		10101010	214		11010110
171		10101011	215		11010111
172		10101100	216		11011000
173		10101101	217		11011001
174		10101110	218		11011010
175		10101111	219		11011011
176		10110000	220		11011100
177		10110001	221		11011101
178		10110010	222		11011110
179		10110011	223		11011111
180		10110100	224		11100000
181		10110101	225		11100001
182		10110110	226		11100010
183		10110111	227		11100011
184		10111000	228		11100100
185		10111001	229		11100101
186		10111010	230		11100110
187		10111011	231		11100111
188		10111100	232		11101000
189		10111101	233		11101001
190		10111110	234		11101010
191		10111111	235		11101011
192		11000000	236		11101100
193		11000001	237		11101101
194		11000010	238		11101110
195		11000011	239		11101111
196		11000100	240		11110000
197		11000101	241		11110001
198		11000110	242		11110010
199		11000111	243		11110011
200		11001000	244		11110100
201		11001001	245		11110101
202		11001010	246		11110110
203		11001011	247		11110111
204		11001100	248		11111000
205		11001101	249		11111001
206		11001110	250		11111010
207		11001111	251		11111011
208		11010000	252		11111100
209		11010001	253		11111101
210		11010010	254		11111110
211		11010011	255		11111111

G — Stages Preset in the Controller

Default Stages

DEFAULT-PP-T
DEFAULT-PP-R
DEFAULT-CC-T
DEFAULT-CC-R

Translation Stages

CTS25	ILS100CC	MTL100PP0.1	MTM100PP1/1
CTS25#10.01	ILS100CCHA	MTL100PP1	MTM100CC0.1/-1
	ILS150PP	MTL100PP2.54	MTM100CC0.1/0
GVM500PE10	ILS150CC	MTL100CC0.1HA	MTM100CC0.1/1
GVM500PE100	ILS150CCHA	MTL100CC1	MTM100CC0.1#72/-1
GVM500PP1	ILS200PP	MTL150PP0.1	MTM100CC0.1#72/0
GVM500PP10	ILS200CC	MTL150PP1	MTM100CC0.1#72/1
GVM500CC1	ILS200CCHA	MTL150PP2.54	MTM100CC1/-1
GVM500CC10	ILS250PP	MTL150CC0.1HA	MTM100CC1/0
GVM700PE10	ILS250CC	MTL150CC1	MTM100CC1/1
GVM700PE100	ILS250CCHA	MTL200PP0.1	MTM100CC1#79/-1
GVM700PP1	IMS300PP	MTL200PP1	MTM100CC1#79/0
GVM700PP10	IMS300CC	MTL200PP2.54	MTM100CC1#79/1
GVM700CC1	IMS300CCHA	MTL200CC0.1HA	MTM100CC0.1HA/-1
GVM700CC10	IMS400PP	MTL200CC1	MTM100CC0.1HA/0
GVM1000PE10	IMS400CC	MTL250PP0.1	MTM100CC0.1HA/1
GVM1000PE100	IMS400CCHA	MTL250PP1	MTM100CC0.1HAT/-1
GVM1000PP1	IMS500PP	MTL250PP2.54	MTM100CC0.1HAT/0
GVM1000PP10	IMS500CC	MTL250CC0.1HA	MTM100CC0.1HAT/1
GVM1000CC1	IMS500CCHA	MTL250CC1	
GVM1000CC10	IMS600PP		MTM150PE0.1/-1
GVM1400PE10	IMS600CC	MTM100PE0.1/-1	MTM150PE0.1/0
GVM1400PE100	IMS600CCHA	MTM100PE0.1/0	MTM150PE0.1/1
GVM1400PP1		MTM100PE0.1/1	MTM150PE1/-1
GVM1400PP10	MFN8PP	MTM100PE1/-1	MTM150PE1/0
GVM1400CC1	MFN8PP0.1	MTM100PE1/0	MTM150PE1/1
GVM1400CC10	MFN8CC	MTM100PE1/1	MTM150PP0.1/-1
	MFN8CC0.1	MTM100PP0.1/-1	MTM150PP0.1/0
ILS50PP	MFN25PP	MTM100PP0.1/0	MTM150PP0.1/1
ILS50CC	MFN25PP0.1	MTM100PP0.1/1	MTM150PP1/-1
ILS50CCHA	MFN25CC	MTM100PP1/-1	MTM150PP1/0
ILS100PP	MFN25CC0.1	MTM100PP1/0	MTM150PP1/1

MTM150CC0.1/-1
MTM150CC0.1/0
MTM150CC0.1/1
MTM150CC0.1#72/-1
MTM150CC0.1#72/0
MTM150CC0.1#72/1
MTM150CC1/-1
MTM150CC1/0
MTM150CC1/1
MTM150CC1#79/-1
MTM150CC1#79/0
MTM150CC1#79/1
MTM150CC0.1HA/-1
MTM150CC0.1HA/0
MTM150CC0.1HA/1
MTM150CC0.1HAT/-1
MTM150CC0.1HAT/0
MTM150CC0.1HAT/1

MTM200PE0.1/-1
MTM200PE0.1/0
MTM200PE0.1/1
MTM200PE1/-1
MTM200PE1/0
MTM200PE1/1
MTM200PP0.1/-1
MTM200PP0.1/0
MTM200PP0.1/1
MTM200PP1/-1
MTM200PP1/0
MTM200PP1/1
MTM200CC0.1/-1
MTM200CC0.1/0
MTM200CC0.1/1
MTM200CC0.1#72/-1
MTM200CC0.1#72/0
MTM200CC0.1#72/1
MTM200CC1/-1
MTM200CC1/0
MTM200CC1/1
MTM200CC1#79/-1
MTM200CC1#79/0
MTM200CC1#79/1
MTM200CC0.1HA/-1
MTM200CC0.1HA/0
MTM200CC0.1HA/1
MTM200CC0.1HAT/-1
MTM200CC0.1HAT/0
MTM200CC0.1HAT/1

MTM250PE0.1/-1
MTM250PE0.1/0
MTM250PE0.1/1
MTM250PE1/-1
MTM250PE1/0
MTM250PE1/1
MTM250PP0.1/-1
MTM250PP0.1/0
MTM250PP0.1/1
MTM250PP1/-1
MTM250PP1/0
MTM250PP1/1
MTM250CC0.1/-1
MTM250CC0.1/0
MTM250CC0.1/1
MTM250CC0.1#72/-1
MTM250CC0.1#72/0
MTM250CC0.1#72/1
MTM250CC1/-1
MTM250CC1/0
MTM250CC1/1
MTM250CC1#79/-1
MTM250CC1#79/0
MTM250CC1#79/1
MTM250CC0.1HA/-1
MTM250CC0.1HA/0
MTM250CC0.1HA/1
MTM250CC0.1HAT/-1
MTM250CC0.1HAT/0
MTM250CC0.1HAT/1

TBM400PE
TBM400CC
TBM400CC1HA
TBM600PE
TBM600CC
TBM600CC1HA
TBM800PE
TBM800CC
TBM800CC1HA
TBM1000PE
TBM1000CC
TBM1000CC1HA
TBM1200PE
TBM1200CC
TBM1200CC1HA
TBM1400PE
TBM1400CC
TBM1400CC1HA
TBM1600PE

TBM1600CC
TBM1600CC1HA

TIX200CC0.1
TIX200CC0.5
TIX200PP0.5
TIX200PP1
TIXY200CC0.1
TIXY200CC0.5
TIXY200PP0.5
TIXY200PP1

TS50DC0.5
TS50DC1
TSP50
TS100DC0.5
TS100DC1
TSP100
TS150DC0.5
TS150DC1
TST150DC0.1
TST150DC0.5
TST150DC1
TSW150DC1
TSW150DC0.5
TSV150DC0.5
TSP150
TSPW150
TS200DC0.5
TS200DC1
TST200DC0.1
TST200DC0.5
TST200DC1
TSW200DC1
TSW200DC0.5
TSP200
TSPW200
TS250DC0.5
TS250DC1
TS300DC0.5
TS300DC1
TSW300DC1
TSW300DC0.5
TSP300
TSPW300

UTM25PE0.1/-1
UTM25PE0.1/0
UTM25PE0.1/1
UTM25PE1/-1
UTM25PE1/0
UTM25PE1/1
UTM25PP0.1/-1
UTM25PP0.1/0
UTM25PP0.1/1
UTM25PP1HL/-1
UTM25PP1HL/0
UTM25PP1HL/1
UTM25CC0.1/-1
UTM25CC0.1/0
UTM25CC0.1/1
UTM25CC1HL/-1
UTM25CC1HL/0
UTM25CC1HL/1
UTM25CC1HL#72/-1
UTM25CC1HL#72/0
UTM25CC1HL#72/1
UTM25CC0.1DD/-1
UTM25CC0.1DD/0
UTM25CC0.1DD/1
UTM25CC1DD/-1
UTM25CC1DD/0
UTM25CC1DD/1

UTM50PE0.1/-1
UTM50PE0.1/0
UTM50PE0.1/1
UTM50PE1/-1
UTM50PE1/0
UTM50PE1/1
UTM50PP0.1/-1
UTM50PP0.1/0
UTM50PP0.1/1
UTM50PP1HL/-1
UTM50PP1HL/0
UTM50PP1HL/1
UTM50CC0.1/-1
UTM50CC0.1/0
UTM50CC0.1/1
UTM50CC1HL/-1
UTM50CC1HL/0
UTM50CC1HL/1
UTM50CC1HL#72/-1
UTM50CC1HL#72/0
UTM50CC1HL#72/1
UTM50CC0.5HA/-1
UTM50CC0.5HA/0
UTM50CC0.5HA/1
UTM50CC0.5HA#72/-1
UTM50CC0.5HA#72/0

UTM50CC0.5HA#72/1	UTM100CC1HL/1	UTM150PP0.1/1	UTS20PP0.1
UTM50CC0.1DD/-1	UTM100CC1HL#72/-1	UTM150PP1HL/-1	UTS20PP0.1F
UTM50CC0.1DD/0	UTM100CC1HL#72/0	UTM150PP1HL/0	UTS20PP1
UTM50CC0.1DD/1	UTM100CC1HL#72/1	UTM150PP1HL/1	UTS20PP1F
UTM50CC1DD/-1	UTM100CC0.5HA/-1	UTM150CC0.1/-1	UTS20CC0.1
UTM50CC1DD/0	UTM100CC0.5HA/0	UTM150CC0.1/0	UTS20CC0.1F
UTM50CC1DD/1	UTM100CC0.5HA/1	UTM150CC0.1/1	UTS20CC1
	UTM100CC0.5HA#72/-1	UTM150CC1HL/-1	UTS20CC1F
UTM100PE0.1/-1	UTM100CC0.5HA#72/0	UTM150CC1HL/0	
UTM100PE0.1/0	UTM100CC0.5HA#72/1	UTM150CC1HL/1	UZM80PE0.1
UTM100PE0.1/1	UTM100CC0.1DD/-1	UTM150CC1HL#72/-1	UZM80PP0.1
UTM100PE1/-1	UTM100CC0.1DD/0	UTM150CC1HL#72/0	UZM80CC0.1
UTM100PE1/0	UTM100CC0.1DD/1	UTM150CC1HL#72/1	UZM160PE0.05
UTM100PE1/1	UTM100CC1DD/-1	UTM150CC0.5HA/-1	UZM160PP0.05
UTM100PP0.1/-1	UTM100CC1DD/0	UTM150CC0.5HA/0	UZM160PP0.1
UTM100PP0.1/0	UTM100CC1DD/1	UTM150CC0.5HA/1	UZM160CC0.05
UTM100PP0.1/1		UTM150CC0.5HA#72/-1	UZM160CC0.05#72
UTM100PP1HL/-1	UTM150PE0.1/-1	UTM150CC0.5HA#72/0	UZM160CC0.1
UTM100PP1HL/0	UTM150PE0.1/0	UTM150CC0.5HA#72/1	
UTM100PP1HL/1	UTM150PE0.1/1	UTM150CC0.1DD/-1	UZS80PP0.1
UTM100CC0.1/-1	UTM150PE1/-1	UTM150CC0.1DD/0	UZS80CC0.1
UTM100CC0.1/0	UTM150PE1/0	UTM150CC0.1DD/1	
UTM100CC0.1/1	UTM150PE1/1	UTM150CC1DD/-1	VP-25XA
UTM100CC1HL/-1	UTM150PP0.1/-1	UTM150CC1DD/0	VP-5ZA
UTM100CC1HL/0	UTM150PP0.1/0	UTM150CC1DD/1	

Rotation Stages

495PE	BGM160PP	RTM120CC	RV120PE
495APE	BGM160CC	RTM120CC#79	RV120PEHL
495PP	BGM160CC#79	RTM160PE	RV120PPHL
495APP	BGM200PE	RTM160PP	RV120PP
495CC	BGM200PP	RTM160CC	RV120CC
495ACC	BGM200CC	RTM160CC#79	RV120CCHL
495CCHL	BGM200CC#79	RTM240PE	RV120HA
495ACCHL		RTM240PP	RV120HAHL
	PR50PP	RTM240CC	RV120HAT
BGM50PE	PR50CC	RTM240CC#79	RV120HAHLT
BGM50PP		RTM350PE	RV160PE
BGM50CC	RGV100	RTM350PP	RV160PEHL
BGM80PE		RTM350CC	RV160PPHL
BGM80PP	RTM80PE	RTM350CC#79	RV160PP
BGM80CC	RTM80PP		RV160CC
BGM120PE	RTM80CC	RV80PE	RV160CCHL
BGM120PP	RTM80CCHL	RV80PEHL	RV160HA
BGM120CC	RTM80CCHL#72	RV80PP	RV160HAHL
BGM120CC#79	RTM120PE	RV80CC	RV160HAT
BGM160PE	RTM120PP	RV80CCHL	RV160HAHLT

RV240PE	RV350PP	UBG80PP	URM80ACCHL
RV240PEHL	RV350CC	UBG80CC	URM100PE
RV240PPHL	RV350CCHL	UBG120PP	URM100APE
RV240PP	RV350HA	UBG120CC	URM100PP
RV240CC	RV350HAHL		URM100APP
RV240CCHL	RV350HAT	URM80PE	URM100CC
RV240HA	RV350HAHLT	URM80APE	URM100ACC
RV240HAHL		URM80PP	URM100CCHL
RV240HAT	SR50PP	URM80APP	URM100CCHL#72
RV240HAHLT	SR50CC	URM80CC	URM100ACCHL
RV350PE		URM80ACC	URM150PP
RV350PEHL	UBG50PP	URM80CCHL	URM150CCHL
RV350PPHL	UBG50CC	URM80CCHL#72	

Actuators

850F	CMA12PP	VM4CC	VM25.4PPE
850F-HS	CMA12CCCL	VM4CCE	VM25.4CC
850F-LS	CMA25PP	VM12.7PP	VM25.4CCE
850G	CMA25CCCL	VM12.7PPE	
850G-HS		VM12.7CC	VP-25AA
850G-LS	VM4PP	VM12.7CCE	
	VM4PPE	VM25.4PP	

Drives

EM31CC-T	EM41PP-T
EM31CC-R	EM41PP-R

H — Factory Service

Introduction

This section contains information regarding factory service for the MM4006. The MM4006 contains no user-serviceable parts. The user should not attempt any maintenance or service of this instrument and/or accessories beyond the procedures outlined in the Troubleshooting Guide, Appendix E. Any problem that cannot be resolved should be referred to Newport Corporation or your Newport representative for assistance.

Obtaining Service

To obtain information about factory service, contact Newport Corporation or your Newport representative. Please have the following information available:

1 Instrument model number (MM4006).

2 Instrument serial number.

3 Firmware version number.

4 Description of the problem.

If the instrument is to be returned for repair, you will be given a Return Authorization Number, which you should refer to in your shipping documents. Please fill out the service form on the next page and return the completed form with your system.

Service Form

Your Local Representative
Tel.: _____
Fax: _____

Name: _____ Return Authorization #: _____
(Please obtain prior to return of item)

Compagny: _____

Adress: _____ Date: _____

Country: _____ Phone Number: _____

P.O. Number: _____ Fax Number: _____

Item(s) Being Returned:

Model #: _____ Serial #: _____

Description: _____

Reasons of return of goods (please list any specific problems): _____