

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Newport / Klinger MC4* at our website: ***Click HERE***

MC-4/MD-4

STEPPING MOTOR CONTROLLER DRIVER

40K

OPERATOR AND TECHNICAL MANUAL

NEWPORT CORPORATION

1791 Deere Avenue • Irvine, CA 92714
Tel. No. (949) 863-3144 • Fax No. (949) 253-1800

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Warranty

Newport Corporation warrants this product to be free from defects in material and workmanship for a period of 1 year from the date of shipment. If found to be defective during the warranty period, the product will either be repaired or replaced at Newport's option.

To exercise this warranty, write or call your local Newport representative, or contact Newport headquarters in Irvine, California. You will be given prompt assistance and return instructions. Send the instrument, transportation prepaid, to the indicated service facility. Repairs will be made and the instrument returned, transportation prepaid. Repaired products are warranted for the balance of the original warranty period, or at least 90 days.

Limitation of Warranty

This warranty does not apply to defects resulting from modification or misuse of any product or part. This warranty also does not apply to fuses.

This warranty is in lieu of all other warranties, expressed or implied, including any implied warranty of merchantability or fitness for a particula use. Newport Corporation shall not be liable for any indirect, special, or consequential damages.

©1995 Newport Corporation
1791 Deere Avenue
Irvine, CA 92714
(949) 863-3144

P/N 71174, Rev. A
IN-05951

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

TABLE OF CONTENTS

INTRODUCTION v

CHAPTER 1: MC4/MD4 System Hookup 1-1

1.1 MD4 to Stage(s) Cable Connection
1.2 MC4 to MD4 Cable Connection
1.3 Power Input
1.4 MC4 Motor Type Programming
1.5 Motor Type Programming Theory of Operation

CHAPTER 2: Axis Configuration Switch Settings 2-1

CHAPTER 3: Local Front Panel Operation 3-1

3.1 Front Panel Positioning
3.2 Manual Low Speed and High Speed Programming
3.2.1 Setting Low Speed Rate
3.2.2 Setting High Speed Rate
3.3 Origin Search
3.4 Internal Program Selection and Execution
3.5 Resetting Front Panel Displays
3.6 Emergency Stop
3.7 Local Lockout Keyswitch

CHAPTER 4: IEEE-488 Interface 4-1

4.1 IEEE-488 Connector Pin Definitions
4.2 MC4 and the IEEE-488
4.2.1 IEEE-488 Application Program
4.2.2 Configuring MC4 for IEEE-488
4.2.3 Testing the IEEE-488 Interface
4.3 Local Lockout Keyswitch

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 5: RS-232-C Interface 5-1

5.1 RS-232-C Connector for the MC4
5.2 MC4 and the RS-232-C
5.2.1 Hardware/Software Handshake
5.2.2 RS-232-C Application Program
5.2.3 Configuring MC4 for RS-232-C
5.2.4 Setting up for RS-232-C
5.2.5 Testing the RS-232-C Interface

CHAPTER 6: Automated Control/Command Set Usage 6-1

6.1 Summary of MC4 Commands
6.2 Theory and Description of Commands
6.2.1 Principles of Command Execution
6.2.2 Simultaneous and Interdependent Mode Commands
6.3 Command Programming Modes
6.3.1 Real-Time Programming Mode
6.3.2 Program Entry and Edit Mode
6.3.3 Program Storage Organization
6.3.4 Program Execution Mode
6.3.5 Practice Command Programming Modes
6.4 Status Report/Request
6.4.1 Status Format Command
6.4.2 MC4 Status Response Format
6.4.3 Status Report Practice
6.4.4 Error Checking and Reporting
6.5 Service Request Generation
6.5.1 Service Request Enable Command
6.5.2 MC4 Status Byte Format

CHAPTER 7: Detailed Description of MC4 Commands 7-1

7.1.1 Summary of MC4 Commands
7.1.2 Alphabetical Listing and Description of Commands

CHAPTER 8: MC4 General Purpose Input/Output Connector 8-1

CHAPTER 9: Front Panel Position Display 9-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 10: MC4 Theory of Operation 10-1

10.1 Frequency
10.2 Quantity
10.3 Direction
10.4 Time

CHAPTER 11: Sample Programs 11-1

11.1 IEEE-488 Sample Programs (BASIC)
11.1.1 Real Time Devices (BASIC)
11.1.2 National Instruments (BASIC)
11.1.3 HP-85 (BASIC)
11.2 RS-232-C Sample Programs (BASIC)
11.2.1 IBM-PC: Hardware Handshake (BASIC)
11.2.2 IBM-PC: Software Handshake (BASIC)
11.2.3 IEEE Send and Receive (C)
11.2.4 MC4 Listen with RS-232-C (C)

CHAPTER 12: Ill-Condition Status Messages: Problems and Solutions 12-1

12.1 Operational Problems
12.2 IEEE-488 and RS-232-C Communication Problems
12.3 Other Problems and Possible Causes
12.3.1 IEEE-488
12.3.2 RS-232-C
12.3.3 Operational

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

INTRODUCTION

Klinger Scientific's Series MC4 Programmable Stepping Motor Controller provides up to four axes of intelligent stepping motor pulse generation. When connected to a stepping motor driver and a host computer, it forms a compact, high performance unit capable of both basic independent movements and very elaborate repetitive cycles, similar to a robotic system. Different controls, located on the front panel, give the MC4 total autonomy.

Model MC4 is a 4-axis intelligent stepping motor pulse generator. It does not contain any internal power amplifiers to drive stepping motors. It is designed to be used with our MD4 Stepping Motor Power Driver.

The MC4 Programmable Stepping Motor Controller features the following capabilities:

- IEEE-488 and RS-232-C compatible
- flexible command set for automated operation
- manual control via front panel switches and remote control unit
- immediate command execution or command storage in a program buffer for later execution
- simultaneous, sequential, and interdependent command execution
- status reporting while idle and while executing commands
- service request generation at the completion of any move and at the occurrence of a travel limit
- error checking and reporting over the communication channel
- indication of position via LED display and reporting of position over the communication channel
- programmable step pulse rates in either pulses/second or time delay between pulses
- maximum pulse rate: 40,000 pulses/second per axis
- relative and absolute positioning in both the positive and negative directions
- programmable time delay between execution of commands

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- variable definition of stepping pulse parameters for succinct program coding
- controlled looping for repetitive execution of commands
- setting and clearing of up to 8 output bits for triggering external devices
- 3 external inputs for synchronization of program execution
- easy editing and listing of internally stored programs
- user program retention in battery back up RAM during power-off state
- 8 kilobytes user program buffer
- front panel selection for execution of up to 9 internally stored programs

Model MD4 Stepping Motor Power Driver is a modular unit capable of driving from one to four stepping motors. This can include any combination of variable reluctance and permanent magnet motors.

Variable reluctance motors are driven in full step mode only. Permanent magnet motors are driven in ministep mode only. A ministep is a fraction of a full step. The MD4 will move a permanent magnet stepping motor one-tenth of a full step for every step pulse it receives from the MC4. For a normally specified 200 step/revolution stepping motor, it now becomes a 2000 step/revolution motor.

HOW THIS MANUAL IS ORGANIZED

The chapter structure of this manual is such that it will guide you through the logical sequence of events leading to a quick and successful use of the MC4/MD4 system. The approach is hands-on; as you read each chapter, you should be doing what it asks you to do.

Chapter 1 describes the proper connection configuration for your system. Here you will connect the MC4 to the MD4 with 15 pin cables. You will also connect the MD4 to the motors/positioning stages with 19 pin cables. When you have made all the connections, you will then program the MC4 from the front panel to tell it what type of motor/stage is connected at each of its axes.

Chapter 2 describes the switch settings which tell the MC4 the parameters of your system. Sample parameters are: encoder/no encoder, 200(48)/2000 step encoder, origin switch on the stage itself, top zero pulse available.

Chapter 3 describes the use and functions of the front panel switches. After you have built and configured your system in chapters 1 and 2, you will use the front panel functions to move and position your stages/motors. You will also use the switches and the displays to program low speed and high speed stepping rates.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapters 4 and 5 describe the two communication interfaces used to send commands to the MC4. These commands enable us to duplicate all the functions of the front panel and much more. Chapter 4 describes the IEEE-488 interface in general and, more importantly, as it pertains to the MC4. Chapter 5 describes the RS-232-C serial interface in the same manner. In these chapters you will learn the switch settings on the MC4 necessary to enable your host computer to communicate with the MC4. You will then use the sample programs to send commands to the MC4 and observe the results. Once you have established communication and sent a few rudimentary commands, you will move on to chapter 6.

Chapter 6 describes the theory of operation of MC4 commands. You will learn about the format, the restrictions (how and when they can be used), and the techniques (i.e., grouping of commands) to accomplish some specific tasks. As you learn each new command, you will program it on your host computer and, by observing the MC4 response, you will verify its operation as described in this manual.

Chapter 7 describes all MC4 commands in detail. The commands are listed in alphabetical order for easy location. Each command is described by its function, legal parameters, options and examples of use.

Chapter 8 describes the MC4 General Purpose Input/Output connector. The majority of the input pins of this connector duplicate every switch on the MC4 front panel. By grounding an input pin, the same result is obtained as if you pressed a switch on the front panel of the same designation. Three additional input pins serve as event synchronization, (i.e., the MC4 may be programmed to wait or proceed based on the logic level of any one of these inputs). Eight output pins serve as programmable output bits. The MC4 has two commands that enable you to set and clear any of these bits.

Chapter 9 describes the MC4 front panel display. You will see what the display represents and how it is to be interpreted.

Chapter 10 describes the theory of operation of the MC4.

Chapter 11 presents the sample programs for the IEEE-488 interface and the RS-232-C interface.

Chapter 12 describes the error and status messages available from the MC4. It also describes possible problems you may encounter and their probable cause.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 1: MC4/MD4 SYSTEM HOOKUP

This chapter describes the step by step procedure for properly connecting all the components of your system. It is advised that you physically perform each step as you read it. This will insure proper set-up and no harm to the system. An improper connection may damage the internal boards.

The procedure described assumes a four motor/stage system. If your system will consist of fewer than four stages, go no further than necessary.

1.1 MD4 TO STAGE(S) CABLE CONNECTION

The cable used may be 2 meters, 5 meters, or a custom length. It is terminated with a round 19 pin female connector on one side and a round 19 pin male connector on the other side. We will refer to these connectors and their mates by their brand name: Jaeger.

These cable connections will depend solely on the type of motors connected to your stages. This connection is very sensitive. Please be careful and double check before you make any cable hookup. A wrong connection may destroy the motor driver boards of the MD4.

Determine the type of motors connected to your stages by referring to Figure 1. There you will see the possible motors and their corresponding codes names: UE30, UE31, UE71, UE72 and UE73.

Look on the back of the MD4. Right below each 19 pin female Jaeger connector you will see a label specifying the type of motor that can be connected to this connector.

Connect a cable from the 19 pin male Jaeger connector on the stage/motor to the "motor-type matching" 19 pin female Jaeger connector on the back of the MD4.

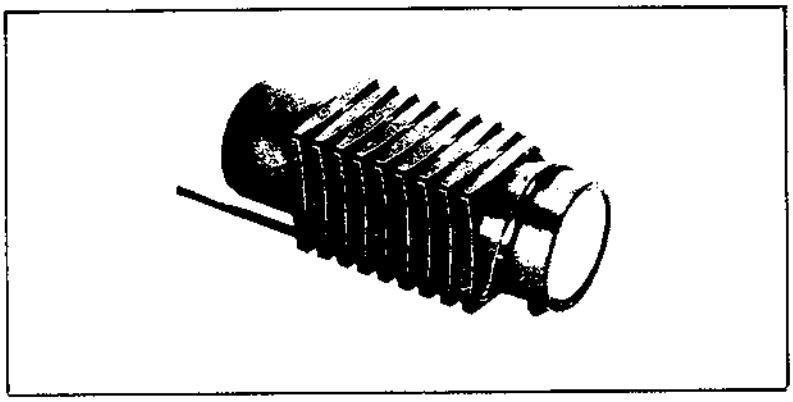
Repeat the above for the rest of your stages.

1.2 MC4 TO MD4 CABLE CONNECTION

The cable used is approximately 3 feet long and is terminated with a 15 pin male D connector on each end.

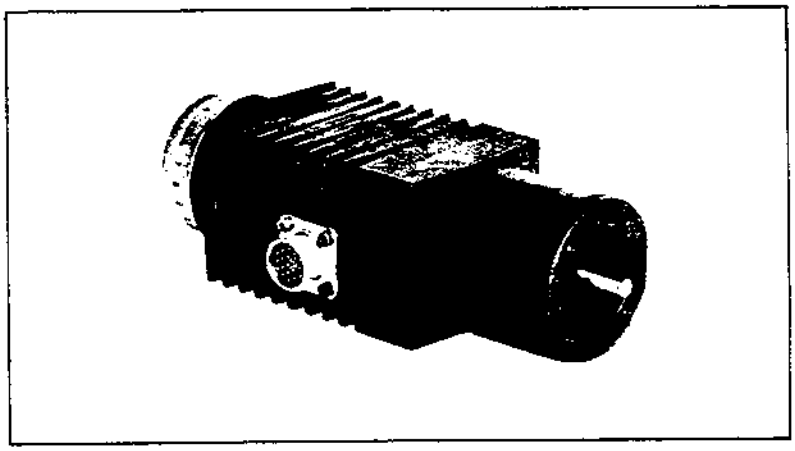
Look on the back panels of the MC4 and the MD4. There you will see 15 pin female D connectors labelled W-AXIS, X-AXIS, Y-AXIS, Z-AXIS on the MC4 and INPUT W, INPUT X, INPUT Y, INPUT Z on the MD4.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



UE30 Stepping Motor

Type: Variable reluctance
Rated voltage: 28 V
Rated current: 200 mA
Resistance/winding: 140-150 Ω
Inductance/winding: 100 mH



UE72 Stepping Motor

Type: Permanent magnet
Rated voltage: 5.3 V
Rated current: 1.6 A
Resistance/winding: 3.3 Ω
Inductance/winding: 9 mH

Figure 1 - UE30 and UE72 stepping motors

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Check the stage/motor connected to the OUTPUT W Jaeger connector on the back of the MD4. If you want to reference this stage/motor as your w-axis, connect a cable from the MD4 connector labelled INPUT W to the MC4 connector labelled W-AXIS. If you do not want to reference this stage/motor as your w-axis, then connect a cable from MD4 connector labelled INPUT W to the MC4 connector with the desired axis label. Repeat this procedure for the rest of your stages/motors.

1.3 POWER INPUT

The MC4 and the MD4 will operate properly with either a 110 or 220 VAC line input. Both units are factory preset to operate on 110 VAC. If you need to operate on 220 VAC, please call Klinger Scientific for directions to convert to 220 VAC.

Using the line cords provided, connect each unit to the power line.

1.4 MC4 MOTOR TYPE PROGRAMMING

Once you have connected your motors/stages to the desired axis input/output connectors, you must tell the MC4 what this set-up is for each axis. You will do this by using a two key combination on the MC4 front panel. The two keys are: 1. the reset switch of the axis you want to program and 2. the SELECT switch inside the block of switches designated PROGRAM.

The MC4 stores the motor type information inside its battery back-up RAM. This enables it to remember the motor types during its power-off time. Thus, once you program a motor type for a particular axis, you do not need to re-program on subsequent power ups. On the back of the MC4 there is a switch labelled WRITE PROTECT. One position of the switch allows storing into the battery back-up RAM, and the other position disables this function. When programming motor types, this switch must be set towards dip switch bank S3 to enable storing into the battery back-up RAM. If you try to program motor types with the write protect feature true, the MC4 will beep to indicate an error.

Turn on the MC4 by pressing the POWER switch until it clicks. The displays will all be zeroes, a red bar in the upper left hand display will be illuminating W-AXIS, and the number next to the SELECT switch in the "select window" will be 1. The red bar illuminator identifies the display that corresponds to that axis. The RESET switch on the right of a display is that display's reset switch.

Press and release the AXIS SELECT switch. The red bar will now be illuminating X-AXIS. Repeat this several times and notice which display corresponds to which axis.

You will now tell the MC4 which type of motor is connected to each axis. The procedure is as follows:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1. Place write protect switch towards dip switch bank S3

2. Press and hold down the w-axis reset switch.

3. Press and hold down the program select switch.

4. Release the w-axis reset switch.

5. Release the program select switch

At this time, the number in the select window will change to 3 or a 7 and will flash. The w-axis illuminator bar will also flash to indicate that your are setting the motor type for the w-axis. The specific number that appears will be the code for the motor type that we connected to the w-axis when we set up your system at the factory. The numbers 3 and 7 are codes for the following motor types:

Code	Motor Type
3	UE30
7	UE71, UE72, UE73

If the code number and the type of motor connected to the w-axis match, then go to step 7. Otherwise,

6. Press and release the program select switch.

At this time, the number in the select window will change to the only other possible code number and will flash.

7. Press and release the w-axis reset switch.

At this time, the MC4 will store the motor type information into its battery back up memory. Once set, you do not have to do this again unless you wish to change motor types.

8. Repeat the above procedure for your X,Y, and Z axes by using the X,Y, and Z reset switches, respectively, and the program select switch.

9. Restore write protect switch to its original position

1.5 MOTOR TYPE PROGRAMMING THEORY OF OPERATION

Whenever the above step sequence 1-3 is performed, the number in the select window will show the code for the present motor type programmed for that axis. The number will flash to indicate that you are in this mode of operation. The axis illuminator of the axis you selected for programming the motor type will also flash to indicate which axis is being set.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

While in this mode, pressing the SELECT switch will alternate the code numbers between 3 and 7.

To exit this mode of operation and at the same time program the MC4 for the motor type, press and release the reset switch of the corresponding axis.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 2: AXIS CONFIGURATION SWITCH SETTINGS

This chapter describes the switch settings that will tell the MC4 the parameters of your system. For each stage/motor in your system, you will have to determine the following parameters:

1. Is there an origin switch on the stage: Yes/No
2. Is there a top zero pulse option: Yes/No
3. Is there an encoder on the motor: Yes/No
4. If there is an encoder, is it a 48, 200 or a 2000 pulse encoder.

Once you have determined these parameters for each stage, you can then set the MC4 axis configuration switches labelled SW, SX, SY, and SZ. The switch definitions are identical for each axis and represent the following:

Switch Location	Function and Switch Position
1	Origin switch yes - DOWN / Origin switch no - UP
2	Top zero yes - DOWN / Top zero no - UP
3	Encoder yes - UP / Encoder no - DOWN
4	48 or 200 pulse encoder - UP / 2000 pulse encoder - DOWN

NOTE: If switch location #3 was set for no encoder, then switch position of switch location #4 is irrelevant.

NOTE: The 48 pulse encoder is used on UE30 motors. The 200 or 2000 step encoders are used on the UE7x series motors.

Knowing the parameters of the stage connected to the w-axis, set the axis configuration switches labelled SW as follows:

1. If there is an origin switch on the stage, set SW.1 down. If there is no origin switch or if you do not wish to use the origin switch if present, then set SW.1 up.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. If there is a top zero pulse option, set SW.2 down. If there is no top zero or if you do not wish to use the top zero pulse if present, then set SW.2 up.

3. If there is an encoder, set SW.3 up. If there is no encoder or you do not wish to use the encoder if present, then set SW.3 down.

4. If you set SW.3 for encoder, then set SW.4 up if it is a 48 or 200 pulse per revolution encoder or set SW.4 down if it is a 2000 pulse per revolution encoder.

If you set SW.3 for no encoder, then the position of SW.4 is irrelevant.

Repeat the above steps for the rest of your axes.

NOTE:	The front panel display will show the decimal point when it is set for non-encoder (Internal Pulses) or when set for encoder with 2000 point setting.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 3: LOCAL FRONT PANEL OPERATION

In this chapter you will learn about the use and functions of the MC4 front panel switches. The following features are available via front panel controls of the MC4:

- positioning and moving by single step, low speed and high speed operation
- programming low speed and high speed stepping rates
- automatic home (origin) seeking operation
- selection and execution of internally stored programs
- resetting of front panel displays
- emergency stop of all movement

We will now describe the above mentioned features of the MC4 in greater detail. It is advised that you perform each function as it is described.

Turn on both the MC4 and the MD4 by pressing the POWER switch until it clicks into position.

3.1 FRONT PANEL POSITIONING

Front panel control of movement and positioning is done with the AXIS SELECT switch and the set of switches outlined by the box labelled MANUAL. The manual switches are defined as follows:

- right arrow switch does low speed motion in the positive direction
- left arrow switch does low speed motion in the negative direction
- HIGH SPEED switch does high speed motion when pressed along with a low speed switch

Since there is only one set of manual switches and we can operate up to four axes, we use the AXIS SELECT switch to select which axis the manual switches will control.

Manual positioning consists of the following steps:

1. Press and release the AXIS SELECT switch. A red bar indicator will light to illuminate the axis selected.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Each time you press and release the AXIS SELECT switch, the red bar indicator will illuminate the next axis selected.

2. To move a single step at a time on the axis selected, press and release one of the arrow switches.

For UE30 motors, each single step corresponds to a full step of the motor. These motors have 48 steps/revolution; thus one revolution would require 48 single step actuations.

For UE7x motors, each single step corresponds to one-tenth of a full step of the motor. These motors have 200 full steps per revolution; thus one revolution would require 2000 single step actuations.

3. To move continuously at low speed on the axis selected, press and hold down one of the arrow switches. The low speed is factory set at 400 full steps/second. This speed is adjustable. See section 3.2.1 below.

4. To move continuously at high speed on the axis selected, press and hold down both an arrow switch and the HIGH SPEED switch. The high speed is factory set at 2000 full steps/second. This speed is adjustable. See section 3.2.2. The acceleration ramp from low speed to high speed is programmable. The acceleration ramp is factory set at 0.4 second. See section 7.1.1.

3.2 MANUAL LOW SPEED AND HIGH SPEED PROGRAMMING

Programming the low speed and the high speed for a particular axis involves a two switch combination. One switch is the reset switch of the axis for which you wish to set the speed. The other switch is either arrow switch if you wish to set the low speed or the HIGH SPEED switch if you wish to set high speed.

The display of the axis for which you are setting the speed is used to show the speed as you adjust it. As a result, the position display information will be lost.

The MC4 stores the programmed speeds inside its battery back-up RAM. This enables it to remember the set speeds during its power off time. On the back of the MC4 there is a switch labelled WRITE PROTECT. One position of the switch allows storing into the battery back-up RAM and the other position disables this function. When programming speeds, this switch must be set towards the dip switch bank S3 to enable storing into the battery back-up RAM. If you try to set a new speed with the write protect feature true, the MC4 will beep to indicate an error.

3.2.1 SETTING LOW SPEED RATE

To set the low speed for the w axis, the steps are as follows:

1. Place write protect switch towards dip switch bank S3

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. Press and hold down the w axis reset switch

3. Press and hold down either arrow switch

4. Release the w axis reset switch

5. Release the arrow switch

At this time, the w axis display will show the present low speed setting. The w axis illuminator will be flashing to indicate that you are setting the w axis speed.

6. To increment the low speed by 10 full steps/second, press and release the right arrow switch. For continuous incrementing, hold down the right arrow switch.

7. To decrement the low speed by 10 full steps/second, perform step 6 with the left arrow switch.

8. To increment/decrement the low speed by 50 full steps/second use the HIGH SPEED switch. First you must establish increment or decrement by pressing the appropriate arrow switch once. Then press the HIGH SPEED switch to change by 50 full steps/second.

9. When you have reached your desired speed, press and release the w axis reset switch to save your low speed setting and to exit this mode.

3.2.2 SETTING HIGH SPEED RATE

To set the high speed for the w axis the steps are as follows:

1. Place write protect switch towards dip switch bank S3

2. Press and hold down the w axis reset switch

3. Press and hold down the HIGH SPEED switch

4. Release the w axis reset switch

5. Release the HIGH SPEED switch

At this time, the w axis display will show the present high speed setting. The w axis illuminator will be flashing to indicate that you are setting the w axis speed.

6. To increment the high speed by 50 full steps/second, press and release the right arrow switch. For continuous incrementing, hold down the right arrow switch.

7. To decrement the high speed by 50 full steps/second, perform step 6 with the left arrow switch.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

8. To increment/decrement the high speed by 200 full steps/second use the HIGH SPEED switch. First you must establish increment or decrement by pressing the appropriate arrow switch once. Then press the HIGH SPEED switch to change by 200 full steps/second.

9. When you have reached your desired speed, press and release the w axis reset switch to save your high speed setting and to exit this mode.

Repeat the above procedure described for the w axis for all the other axes by using the RESET switch for that axis and the manual control switches.

When you finish setting speeds, set the write protect switch to its original position.

> NOTE: The lowest low speed that may be programmed is 50 full steps per second. The highest high speed is 4000 full steps/second. The smallest gap between the low speed and the high speed allowed is 200 full steps/second. For example, if the low speed is set for 350, the lowest high speed that you may program is 550 full steps/second. You do not have to be wary of these limits. The MC4 will not allow you to exceed them.

3.3 ORIGIN SEARCH

The origin search function is an automatic home seeking operation. It returns a stage/motor to a preset position -- a reference point for subsequent positioning. The origin point must be defined and conveyed to the MC4 through the corresponding axis configuration switch on the back of the unit.

The three possible origin points are:

1. The position defined by the location of a switch mounted on a positioning stage

2. The position defined by the location of a switch mounted on a positioning stage in conjunction with a top zero pulse

3. The position that corresponds to the step count of zero on the display for that axis

You will recall from chapter 2 that axis configuration switch locations #1 and #2 (i.e. SW.1, SW.2, SX.1, etc.) are used to set the MC4 for the above origin point definitions. If you set switch location #1 (i.e., SW.1, SX.1, etc.) for no origin switch, the MC4 automatic defaults to the origin point where the display is zero.

Origin point definition number 3 is called a floating origin point. The reason is that you can always define a new position as your origin point by simply moving to a new location and then resetting the display to zero.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

To perform origin search, do the following:

1. Select the axis by using the AXIS SELECT switch

2. Press and release the ORIGIN SEARCH switch

A red indicator light above the ORIGIN SEARCH switch will turn on to signal this mode of operation. When the origin search is complete, the indicator light will turn off.

3.4 INTERNAL PROGRAM SELECTION AND EXECUTION

The MC4 has a battery back-up RAM for the purpose of storing many user programs. These programs are entered and edited with a computer. They are stored in the battery back-up RAM during the power off state. Subsequently, a program may be selected and executed without having a host computer present. The PROGRAM SELECT and the PROGRAM START switches are used for this purpose. The SELECT switch allows selection of the first nine programs stored. However, note that more than nine programs may be stored and called for execution. To access programs beyond program nine, a computer must be used. See chapter 7, GI command description.

NOTE: When your MC4 was prepared at the factory, it was loaded with five very simple programs. Program 1 moves the w axis 1000 steps in the positive direction at a rate of 2000 steps/second. Then it moves the w axis 1000 steps in the negative direction. Then it loops back to repeat the same process indefinitely. Programs 2, 3, and 4 do the same operation for the X, Y, Z axes, respectively. Program 5 moves all axes simultaneously, does origin search on each axis and toggles output bits.

To execute a program, do the following:

1. Select the program by pressing and releasing the SELECT switch

2. Start program execution by pressing and releasing the START switch

A red indicator light above the START switch will turn on to signal this mode of operation. When the program is done, the indicator light will turn off.

If you wish to stop program execution momentarily, press and release the PAUSE switch. This will cause execution to stop when the present command is done. Note that execution will not stop immediately after you activate pause. The MC4 will pause only after the present command is fully executed. A red indicator light will turn on to signal this mode of operation.

To resume execution after a pause, press and release the PAUSE button again. The red indicator will turn off to signal that pause is now inactive.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 RESETTING FRONT PANEL DISPLAYS

Each axis display has its own reset switch. To perform a reset, simply press and release the RESET switch on the right side of the display you wish to reset.

Resetting a particular axis display does three things:

1. Sets the display readout to zero
2. Sets the present position as the floating origin point
3. Sets the present position as the reference point for the absolute positioning command -- More on this in chapters 6 and 7.

A description of the front panel display and what the numbers represent under different system configurations is presented in chapter 9.

3.6 EMERGENCY STOP

The STOP switch is used to stop all stage movement immediately. Pressing the STOP switch causes the MC4 to stop sending step pulses and returns it to its power up state. When the switch is activated, the MC4 will beep once to acknowledge that it received the stop request.

> NOTE: Any RESET in conjunction with Emergency STOP will flush out the battery back-up RAM, i.e., bring back factory default values. See NOTE on page 4-3.

3.7 LOCAL LOCKOUT KEYSWITCH

The purpose and function of this keyswitch will be revealed in the next chapter when we discuss the IEEE-488 interface.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 4: IEEE-488 INTERFACE

In addition to the front panel controls for moving and positioning, the MC4 also has a command set which enables you to program it to perform a variety of precise and automated operations. The necessary sequence of commands to perform a certain move must be sent to the MC4 by a host computer or a similar device. To establish a path for the correct transfer of commands a standard interface is necessary between the MC4 and the host computer.

The MC4 supports two standard interfaces. These are the IEEE-488 (also known as GPIB and HPIB) and the RS-232-C.

The MC4 IEEE-488 interface implements both the listen function and the talk function. The listen function is used to send commands to the MC4. The talk function is used by the MC4 to send status information back to the host computer. Typical status will be:

1. error message if incorrect command sent
2. current display readout
3. command execution in progress or complete
4. end of travel limits true or false
5. listing of stored programs

4.1 IEEE-488 CONNECTOR PIN DEFINITIONS

For connection to the IEEE-488 bus, the MC4 has the standard 24 pin female ribbon connector. Pin assignments for this connector are shown below and are based on the standard IEEE-488 protocol.

IEEE-488 BUS CONNECTOR

E1	DIO1	E 9	IFC	E17	REN
E2	DIO2	E10	SRQ	E18	Gnd. for DAV
E3	DIO3	E11	ATN	E19	Gnd. for NRFD
E4	DIO4	E12	SHIELD	E20	Gnd. for NDAC
E5	EOI	E13	DIO5	E21	Gnd. for IFC
E6	DAV	E14	DIO6	E22	Gnd. for SRQ
E7	NRFD	E15	DIO7	E23	Gnd. for ATN
E8	NDAC	E16	DIO8	E24	LOGIC Gnd.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.2 MC4 AND THE IEEE-488

We will now describe the steps necessary to establish communications between the MC4 and your host computer. You will test for successful communications by using your host computer to send a few commands to the MC4 and observing the MC4's response. When you are satisfied that the interface is functioning properly, you can move on to chapters 6 and 7 for the description and usage techniques of all MC4 commands. Since your communications interface is set up, you can experiment with each command to fully understand its function.

You will now be asked to do the following:

1. load an IEEE-488 application program into your host computer
2. configure the MC4 dip switches to set communication parameters
3. send commands to the MC4 to verify successful implementation

4.2.1 IEEE-488 APPLICATION PROGRAM

In Chapter 11 we show an application program for several different manufacturers' IEEE-488 interface. If you are using one of these interfaces, enter the program for your interface. If you are using a different manufacturer, study the application program and write a similar one for your interface. The application program does the following:

1. initializes the interface
2. prompts you to enter a command at the keyboard
3. sends the command to the MC4
4. if the command sent was one which requires the MC4 to send back information, the program asks for this information and decodes it

4.2.2 CONFIGURING MC4 FOR IEEE-488

1. Connect a standard IEEE-488 cable from your computer to the IEEE-488 connector on the back of the MC4
2. Set the desired IEEE-488 address for the MC4 by programming dip switch bank S1, accessed from the back of the MC4. S1-1 is the most significant bit and S1-5 is the least significant bit. Our application program is written for the MC4 at address number 6. The dip switch setting for address 6 is:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

S1-1	OFF	0×2^4	=	0
S1-2	OFF	0×2^3	=	0
S1-3	ON	1×2^2	=	4
S1-4	ON	1×2^1	=	2
S1-5	OFF	0×2^0	=	0
				6 ← ADDRESS

3. Set the MC4 for the command terminator your computer will use at the end of every command. Switch S3-4 is used for this purpose. The three possible terminators and the S3-4 setting for each is listed below:

Terminator:	S3-4 Setting:
carriage return	ON (UP)
carriage return / line feed	OFF (DOWN)
line feed	OFF (DOWN)

Our application program will output a carriage return/line feed terminator so set S3-4 OFF (or down).

4.2.3 TESTING THE IEEE-488 INTERFACE

The application program is in your computer and the MC4 is configured. You can now send a few commands to the MC4. By observing the response (or lack of one), you can verify if everything is okay or if you will have to spend some time troubleshooting.

1. RUN the application program

2. At the prompt, enter the command MW and press Return or Enter key

> NOTE: Every command string is terminated by a Return or Enter key so we will not keep repeating that statement.

If there is a stage/motor connected to the w axis, (i.e., the + LIMIT and the - LIMIT indicator are not lit), the w axis will move 1000 steps in the positive direction.

> NOTE: On power up, the MC4 sets the following defaults:
>
Number of Steps:	1000
High Speed Stepping Rate:	2000 steps/second
Low Speed Stepping Rate:	400 steps/second
Direction:	Positive
Acceleration:	0.4 second

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

NOTE: If the above command does not execute, it may be due to a problem with the IEEE-488 interface. The MC4 is designed to detect some IEEE-488 errors. If it detects an error, it will indicate the error by flashing a number in the program select window and beeping 3 times. Refer to chapter 12 for an explanation and possible solution of the error.

The following set of commands and the expected results will reference only the w axis. After you have done the w axis commands or if you do not have a stage/motor connected to the w axis, repeat the commands for all the other axes by simply replacing the W in each command by the axis (X,Y, or Z) you want to program.

At the prompt, enter the following commands and observe the listed results:

COMMAND:	RESULT:
AW	W display reset to zero
NW1234	Set number of steps = 1234
RW1500	Set high speed step rate = 1500 steps/sec.
MW	W axis moves and display goes to 1234
DW	Read the w-axis display; display on the screen will be: W=+00001234
PW500	W axis moves to position 500
NW2000	Set number of steps = 2000
-W	Set direction to negative
MW	W axis moves and display goes to -1500
OW	W axis moves to origin position; display goes to zero if floating origin or some number if origin defined by a switch
MV	V is an invalid parameter. The MC4 will beep and store the error message into its output buffer.
WX	W is an invalid command. The MC4 will beep.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 LOCAL LOCKOUT KEYSWITCH

The purpose of this keyswitch is to enable or override the remote enable signal (REN in IEEE-488 terminology) from the controller. The controller will normally be the IEEE-488 interface of your host computer.

When you are using the IEEE-488 interface to operate the MC4, you will probably wish to disable the front panel functions to prevent any accidental switch activations. You can do this by asserting the REN signal of the IEEE-488 true while the MC4 is addressed and turning the MC4 keyswitch to the ON position.

When you wish to enable the front panel functions, turn the keyswitch to the OFF position.

The keyswitch performs its function only after the MC4 IEEE-488 interface senses the REN signal from the controller and it has been addressed. Prior to that time, the front panel functions will be enabled regardless of the position of the keyswitch.

> NOTE: You can override the Local Lockout ON via the G.P.I.O. connector. See Chapter 8.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 5: RS-232-C INTERFACE

In addition to the front panel controls for moving and positioning, the MC4 also has a command set which enables you to program it to perform a variety of precise and automated operations. The necessary sequence of commands to perform a certain move must be sent to the MC4 by a host computer or a similar device. To establish a path for the correct transfer of commands a standard interface is necessary between the MC4 and the host computer.

The MC4 supports two standard interfaces. These are the RS-232-C and the IEEE-488 (also known as GPIB and HPIB).

The MC4 RS-232-C interface implements a half-duplex version of the RS-232-C protocol. This means that information can pass from your computer to the MC4 and from the MC4 to your computer, but not simultaneously. The MC4 can only accept or only output information at a given time.

The MC4 accepts and executes commands that are part of its command set. It outputs status information when it deems it necessary and when you ask for it. Typical status will be:

1. Error message if incorrect command sent
2. Current display readout
3. Command execution in progress or complete
4. End of travel limits true or false
5. Listing of stored programs

5.1 RS-232-C CONNECTOR FOR THE MC4

The MC4 uses the standard 25 pin female D connector for coupling to a host computer or data terminal. The MC4 connector is configured as a DCE. Your host computer will be a DTE so you can use straight through cable. See section 5.2.4. The following table lists the MC4 connector pinouts for the RS-232-C.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RS-232-C CONNECTOR

1	N.C.	9	+5V
2	Receive Data	10-17	N.C.
3	Transmit Data	18	+12V
4	N.C.	19	N.C.
5	Clear to Send	20	Data Terminal Ready
6	DSR	21-24	N.C.
7	Signal Ground	25	-12V
8	N.C.		

5.2 MC4 AND THE RS-232-C

We will now describe the steps necessary to establish communications between the MC4 and your host computer. You will test for successful communications by using your host computer to send a few commands to the MC4 and observing the MC4's response. When you are satisfied that the interface is functioning properly, you can move on to chapters 6 and 7 for the description and usage techniques of all MC4 commands. Since your communications interface is set up, you can experiment with each command to fully understand its function.

You will now be asked to do the following:

1. Decide whether you will use a hardware handshake or a software handshake to synchronize the data transfers between the MC4 and your computer.

2. Load an RS-232-C application program into your host computer

3. Configure the MC4 dip switches to set communication parameters

4. Send commands to the MC4 to verify successful implementation

5.2.1 HARDWARE/SOFTWARE HANDSHAKE

The MC4 needs a handshake sequence to insure proper reception and execution of all commands. The handshake is such that it prevents you from sending commands when the MC4 is not ready to accept them. The MC4 provides both a hardware and a software handshake. The selection of handshake mode is done with switch S2-4, accessed from the back of the unit.

HARDWARE HANDSHAKE

The hardware handshake uses the following pins on the RS-232-C connector to implement the handshake:

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- Pin 5: Clear to Send

  This signal is an output from the MC4 to your computer

  The MC4 asserts clear to send (+12V) when it is ready to accept a new command

- Pin 20: Data Terminal Ready

  This signal is an input to the MC4 from your computer

  The MC4 polls this signal for a true state (+12V) before its sends any data

To send commands to the MC4, simply monitor the state of the clear to send line. Once it is asserted true, send a command. When the MC4 accepts the command, it will assert clear to send false. Continue polling the state of this line for a true signal before sending the next command.

To receive data from the MC4, your computer must set data terminal ready true when it can accept data. If your computer has a communications buffer, you will probably keep this line always true and accept MC4 data when it wants to send it.

SOFTWARE HANDSHAKE

When the MC4 is ready to accept a new command, it outputs an XON character (ASCII DC1, HEX 11). You will receive the XON character immediately when the MC4 is turned on and more importantly, when the MC4 finishes executing the previous command.

Once the MC4 accepts a command and starts to execute it, it will output an XOFF character (ASCII DC3, HEX 13).

The procedure for the software handshake is very simple. Wait to receive an XON character. When received, send a command. Wait to receive an XON character. When received, send another command, etc. As you can see, you do not even have to look for the XOFF character.

When the MC4 is sending data to your computer, it is checking for reception of XON/XOFF characters from your computer. If your communication buffer is large or your program can read data as it is received from the MC4, you will never send XOFF. However, if you cannot accept data at a certain time, you can send XOFF. At that time the MC4 will suspend transmission until it receives a XON character.

> NOTE: The MC4 powers up in the XON mode. If you never send XOFF, then you never have to send XON.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.2.2 RS-232-C APPLICATION PROGRAM

In Chapter 11 we show two application programs for the RS-232-C interface, one using the hardware handshake and one using the software handshake. These programs are written for the IBM-PC and compatibles. You can choose to load one of these programs into your computer or you can use any serial communications program that you may have. The application program does the following:

1. Initializes the interface
2. Prompts you to enter a command at the keyboard
3. Sends the command to the MC4
4. If the command sent was one which requires the MC4 to send back information, the information comes into your computer's communications buffer and is displayed on the screen

> NOTE: Use Klinger's "EZ232" program, catalog number KE071085A, for the easiest MC4 RS-232C operation.

5.2.3 CONFIGURING MC4 FOR RS-232-C

You will now configure the MC4 for serial communication by using dip switch bank S2 and S3 on the back of the unit to set the communication parameters. Switch bank S2 is used to set the baud rate and the choice between a hardware or a software handshake (refer to Figure 1). Switch bank S3 is used to set the data word characteristic and the command terminator (refer to Figure 2).

S2-1	S2-2	S2-3	BAUD RATE
ON	ON	ON	110
ON	ON	OFF	150
ON	OFF	ON	300
ON	OFF	OFF	600
OFF	ON	ON	1200
OFF	ON	OFF	2400
OFF	OFF	ON	4800
OFF	OFF	OFF	9600

S2-4	HANDSHAKE
ON	Software
OFF	Hardware

Figure 1: Switch Settings for Baud Rate and Handshake Mode

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

S3-1	S3-2	S3-3	WORD CHARACTERISTIC
ON	ON	ON	7 bits, even parity, 2 stop bits
ON	ON	OFF	7 bits, odd parity, 2 stop bits
ON	OFF	ON	7 bits, even parity, 1 stop bit
ON	OFF	OFF	7 bits, odd parity, 1 stop bit
OFF	ON	ON	8 bits, 2 stop bits
OFF	ON	OFF	8 bits, 1 stop bit
OFF	OFF	ON	8 bits, even parity, 1 stop bit
OFF	OFF	OFF	8 bits, odd parity, 1 stop bit

S3-4	TERMINATOR
ON	Carriage return
OFF	Carriage return/line feed or line feed only

Figure 2: Switch Settings for Word Characteristics and Terminator

5.2.4 SETTING UP FOR RS-232-C

1. Make an appropriate cable to connect the MC4 and your computer. You will recall that the MC4 is designed to act as a DCE and your computer is most probably configured as a DTE. As a result your cable should look like this:

PIN #:		PIN #:
1	----------	1
2	----------	2
3	----------	3
5	----------	5
7	----------	7
20	----------	20

2. Set the baud rate to 9600 by using the table in Figure 1. Our application program is written to operate at 9600 baud.

3. Set the word characteristic to match the application program by using the table in Figure 2. Our application program is set for 8 bits, NO parity, and 1 stop bit.

4. Set the handshake mode by using S2-4. Set S2-4 ON (up) for the software handshake and set it OFF (down) for the hardware handshake.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5. If you set up for software handshake mode, and your computer does not provide +12V on pin 20 of its RS-232-C, then you must connect pin 18 and pin 20 of the MC4 RS-232-C connector to supply the +12V necessary.

6. Set the MC4 for the command terminator your computer will use at the end of every command. Switch S3-4 is used for this purpose. Our application program will output only a carriage return terminator thus set S3-4 ON (up).

5.2.5 TESTING THE RS-232-C INTERFACE

The application program is in your computer and the MC4 is configured. You can now send a few commands to the MC4. By observing the response (or lack of one), you can verify if everything is okay or if you will have to spend some time troubleshooting.

1. RUN the application program

2. At the prompt, enter the command MW and press Return or Enter key

> NOTE: Every command string is terminated by a Return or Enter key so we will not keep repeating that statement.

If there is a stage/motor connected to the w axis, (i.e. the + LIMIT and the - LIMIT indicator are not lit), the w axis will move 1000 steps in the positive direction.

> NOTE: On power up, the MC4 sets the following defaults:
>
> | | |
> |---|---|
> | Number of Steps: | 1000 |
> | High Speed Stepping Rate: | 2000 steps/second |
> | Low Speed Stepping Rate: | 400 steps/second |
> | Direction: | Positive |
> | Acceleration: | 0.4 second |

> NOTE: If the above command does not execute, it may be due to a problem with the RS-232-C interface. The MC4 is designed to detect some RS-232-C errors. If it detects an error, it will indicate the error by flashing a number in the program select window and beeping 3 times. Refer to chapter 12 for an explanation and possible solution of the error.

The following set of commands and the expected results will reference only the w axis. After you have done the w axis commands or if you do not have a stage/motor connected to the w axis, repeat the commands for all the other axes by simply replacing the W in each command by the axis (X,Y, or Z) you want to program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

At the prompt, enter the following commands and observe the listed results:

COMMAND:	RESULT:
AW	W display reset to zero
NW1234	Set number of steps = 1234
RW1500	Set high speed step rate = 1500 steps/sec.
MW	W axis moves and display goes to 1234
DW	Read w axis display; display on screen will be: W=+00001234
PW500	W axis moves to position 500
NW2000	Set number of steps = 2000
-W	Set direction to negative
MW	W axis moves and display goes to -1500
OW	W axis moves to origin position; display goes to zero if floating origin or some number if origin defined by a switch
MV	V is an invalid parameter. The MC4 will beep and send an error message to your computer which will be displayed on the screen. The message will be: E02 PARAMETER ERROR
WX	W is an invalid command. The MC4 will beep and the error message will be: E01 COMMAND ERROR

If all the above commands execute as expected, you are ready to move on to Chapters 6 and 7 for a detailed description of all MC4 commands.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 6: AUTOMATED CONTROL/COMMAND SET USAGE

The MC4 has a command set which enables automated and precise control of its operation. The commands may be sent to the MC4 for immediate execution or they may be stored in the MC4's internal battery back up program memory for execution at a later time. The commands are sent to the MC4 using the IEEE-488 interface or the RS-232-C interface.

6.1 SUMMARY OF MC4 COMMANDS

1. **Motor Control**

   - Commands for setting stepping rate

     R program stepping rate from 32 to 4000 steps/second

     T program step intervals from 2 milliseconds to 100 seconds

   - Commands for relative positioning

     +/- set direction of movement

     N set number of steps for the next move

     M start move

   - Commands for absolute positioning

     A set present position as reference point; also resets front panel display

     P move to location specified relative to the defined reference point

   - Command for programming acceleration

     AC

   - Origin search

     O

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. **Wait for Event**

   B wait for motor to finish moving a predefined number of steps

   UT wait a predefined number of milliseconds (time delay)

   UE wait for a high or low condition of an input

3. **Status Report/Request**

   FS program status options desired

   ? report status

   D read front panel display (i.e. position of motors)

4. **Service Request**

   FI program conditions which will generate service request

5. **Output Control**

   C set output bit low

   S set output bit high

6. **Stepping and Holding Torque Inhibit/Enable**

   K

7. **Define and Execute Variable Parameters**

   V

8. **Simultaneous and Interdependent Command Execution**

   <command;command; ... ;command;>

9. **Stored Program Editing Commands**

   E enter program entry mode

   QP end of a single program

   QE exit program entry mode

   I insert a program line

   # delete program lines

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

L list program lines

$ comments inside programs

10. **Stored Program Execution Command**

    GI

11. **Jump Commands**

    H define a "jump to" label

    J jump to the specified label

6.2 THEORY AND DESCRIPTION OF COMMANDS

An MC4 command is always composed of a sequence of ASCII characters terminated by a carriage return, a carriage return/line feed combination, or a line feed.

- the first character is the command letter of the function to be performed
- for commands associated with an axis, the second character represents the desired axis (W,X,Y,Z or A for all axes)
- subsequent characters are numerical parameters for that command

Examples:

```
RX1000    ;set the high speed rate for the X axis at 1000 steps/sec.
NY5000    ;set the number of steps for the next Y axis move to 5000
+Y        ;set Y axis move direction to positive
MY        ;start Y axis move
S3        ;set output bit #3 high
OZ        ;perform origin search on Z axis
DA        ;display all axes positions
FS1F      ;program status response desired
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.1 PRINCIPLES OF COMMAND EXECUTION

The MC4 has three different modes of command execution:

1. **Sequential**

   A command is accepted and executed when the previous command is done. If you try to send a command while the MC4 is busy executing the previous command, your interface will go into a looping mode waiting for the MC4 to become ready. At that time, the command will be sent and the MC4 will proceed to execute it.

   Examples:

   ```
   MX      ;move X

   MY      ;move Y when X move complete
   ```

2. **Simultaneous**

   The execution of all commands starts at the same time. To designate this mode, the set of commands must be enclosed in arrowhead brackets and each command must be followed by a semicolon, including the last one.

   > NOTE: A bunch of simultaneous commands will necessarily be sent in the sequential mode of operation.

   Examples:

   ```
   <MW;MX;>        ;start moving W and X at the same time

   <MX;C2;S2;>     ;move X and at the instant X begins to move, pulse output
                    bit #2
   ```

3. **Interdependent**

   The start of execution of a command is dependent on the extent of execution of another command.

   Examples:

   ```
   NX1000              ;set X axis for 1000 steps
   <MX;BX500;MY;>      ;move X; when X outputs 500 steps, move Y
   <MY;UT2000;MW;>     ;move Y; 2 seconds after Y move started, move W
   ```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2.2 SIMULTANEOUS AND INTERDEPENDENT MODE COMMANDS

The following subset of the MC4 command set can be used in these modes of operation. Using a command not specified below will be considered an error. The MC4 will beep and send an appropriate error message.

```
B     ;wait for specified number of steps to be completed
C     ;clear output bit
H     ;declare jump label
J     ;jump to specified label
M     ;start move on designated axis
S     ;set output bit high
UE    ;wait for external input level
UT    ;wait for specified time delay
+     ;set direction for relative move to plus
-     ;set direction for relative move to minus
V     ;define and use variable parameter
```

6.3 COMMAND PROGRAMMING MODES

The MC4 has three programming modes:

1. real-time mode
2. program entry and edit mode
3. program execution mode

6.3.1 REAL-TIME PROGRAMMING MODE

The MC4 goes automatically into this mode when:

- it is turned on
- when you exit program entry and edit mode (i.e. issue a QE command)
- and when a stored program is finished executing

In this mode, commands are executed as they are received. You have to wait until the previous command is done before sending another command. The implementation of this timing requirement is dependent on the communication interface you are using (IEEE-488 or RS-232-C).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

IEEE-488

When the MC4 accepts a command and starts to execute it, it asserts the "not ready for data" signal on the IEEE-488 interface. Your computer interface senses this signal and does not allow you to send any further commands. Once the previous command is done, the MC4 will assert "ready for data" and now your computer interface will allow you to send another command.

RS-232-C

In the hardware handshake mode, the MC4 will assert "clear to send" false to the interface when it accepts a command and starts to execute it. Your computer interface should be set up to check the status of this line before sending a command. In this case, sensing a "clear to send" false signal, it should wait for this line to go to "clear to send" true. When the previous command is done, the MC4 will assert "clear to send" true.

In the software handshake mode, the MC4 sends an XON character (ASCII 11) to your computer when it is ready to accept another command. Therefore, you must wait to receive the XON before sending another command.

6.3.2 PROGRAM ENTRY AND EDIT MODE

In this mode, a sequence of commands is sent and stored in a program buffer for later execution. The program buffer is battery powered during the MC4 power-off state. This enables all entered programs to be saved and available for execution at any time. Once programs have been entered, they can be read from the program buffer and modified easily.

In this mode, as in real-time mode, a handshake is necessary to transfer the commands. Here however, the MC4 will signal "not ready" only momentarily because it only converts the command to a coded, compact form and stores it in memory. It does not execute it.

6.3.3 PROGRAM STORAGE ORGANIZATION

The capacity of the program buffer is 8 kilobytes. The commands are stored in a coded, compact form. This enables more commands to be stored than if they were stored as ASCII characters. For example, a command like NX1234567 is stored in 4 bytes rather than the 9 bytes that would be required if each character's ASCII code were stored.

The stored programs are organized in terms of a program line. The concept of the program line is essential to the correct execution of all program entry and edit commands. A program line can be:

1. a single command, a single simultaneous command, or a single interdependent command

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. a group of commands, a group of simultaneous commands, or a group of interdependent commands

The MC4 keeps track of all program lines, numbering them in ascending order starting from zero, from the next available line, or from the line number specified with the E command. The following commands all require the proper use of program line numbers.

- enter program entry mode: Ennnn
- list program: Lnnnn
- insert a program line: Innnn
- delete program lines: #nnnn
- execute from program line: GInnnn

  nnnn = line number

You enter program entry mode by sending the E command with a starting line number. After this, you do not have to send program line numbers. You just send commands and the MC4 automatically assigns line numbers to your commands.

The MC4 allows you to store many different programs. Each program is delineated by the command QP. When you have entered all the commands for a particular program, you must issue a QP command to tell the MC4 that this is the end of this program.

If you wish to start entering another program, simply continue issuing commands and when done, end the program with the QP command. If you wish to exit program entry mode, send the QE command.

This will return you to real-time mode. Now you may wish to have the MC4 execute one of your stored programs. This can be the program you just entered or any other program previously stored. If you do not wish to execute any programs, the MC4 is ready to execute any other commands in its command set.

6.3.4 PROGRAM EXECUTION MODE

In this mode, the MC4 will execute a program previously stored in its program memory. You can enter this mode in one of two ways. One is with the front panel switches associated with program selection and execution. The other is with your computer, by issuing a GInnnn command. nnnn specifies the line number at which you wish to start executing your program. The MC4 will continue to execute commands until it encounters a QP command. At that time, it will exit program execution mode and re-enter real-time mode.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

NOTE: There is a difference between executing programs with the front panel controls and executing them with the GI command. With the front panel controls, execution starts with the first command of the program selected. However, with the GI command you can specify a line number and therefore start execution anywhere you want.

6.3.5 PRACTICE COMMAND PROGRAMMING MODES

If you followed the instructions in Chapter 4 or Chapter 5 you presently have a functioning interface between the MC4 and your computer, and you have been sending commands to the MC4 in the real-time programming mode. We will now present commands that illustrate the principles of the other two programming modes of the MC4: program entry and program execution.

Even if your interface is not set up, please go through the following section because it is very informative.

First command: L

The MC4 will respond by sending a listing of the programs stored in its memory. These programs were loaded into the MC4 before it was shipped to you. Note the line number of the command QE, which is the last command. You will use this line number to enter program entry mode at the correct location. For the purpose of illustration, this exercise will assume that this line number is 60. However, wherever you see line number 60 below, you must use the line number of command QE observed above.

Second command: E60

NOTE: If you do not have a stage/motor connected to an axis called out below, please substitute the axis designator of the ones you do have.

Subsequent commands:

```
OX
OY
NX1000
+X
MX
NY2000
-Y
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
MY
QP
OX;OY;NX1000;+X;MX;NY2000;-Y;MY
QP
OW
OX
OY
NW300
NX100
NY200
-W
+X
-Y
<MW;MX;>
<MW;MY;>
QP
OW;OX;OY;NW300;NX100;NY200;-W;+X;-Y;<MW;MX;>;<MW;MY;>
QP
QE
```

After you have entered these commands, you may wish to get a listing from the MC4. If so, enter L60. (Remember 60 = line number parameter used with the E command above; the second command issued). The MC4 will send a listing of the command statements in its memory starting from the location specified with the L command.

The listing will look as follows except the line numbers may be different.

```
0060  OX
0061  OY
0062  NX1000
0063  +X
0064  MX
0065  NY2000
0066  -Y
0067  MY
0068  QP
0069  OX;OY;NX1000;+X;MX;NY2000;-Y;MY
0070  QP
0071  OW
0072  OX
0073  OY
0074  NW300
0075  NX100
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
0076  NY200
0077  -W
0078  +X
0079  -Y
0080  <MW;MX;>
0081  <MW;MY;>
0082  QP
0083  OW;OX;OY;NW300;NX100;NY200;-W;+X;-Y;<MW;MX;>;<MW;MY;>
0084  QP
0085  QE
```

Let us now execute these programs and see what happens.

Command: GI60

Response: Origin search on X axis followed by origin search on Y axis. Move X 1000 steps in the positive direction. Move Y 2000 steps in the negative direction

Command: GI69

Response: Same as above.

Execute the last two programs by issuing the commands: GI71 and then GI83. You will notice that these two programs also have identical responses. As you can see, the same task can be accomplished by different program line structuring.

The motivation for placing more than one command on a single line is that you can tie together all logically connected commands which may make readability and understanding quicker. Also, the MC4 allows only 9999 line numbers. If you are going to have many programs, you may run out of line numbers.

The disadvantage of placing too many commands on a single line is that if you must edit a single command in that line, you must re-type the whole line. Also you cannot insert or delete a command inside a program line consisting of several commands.

6.4 STATUS REPORT/REQUEST

The following status information is available from the MC4:

1. Descriptive error message if command sent is invalid
2. Descriptive error message if error occurs during execution
3. Axis end of travel limits true or false
4. Front panel display reading while stationary and "on the fly"
5. Axis moving or stationary
6. Logic level of 3 input bits

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The above statuses are available while the MC4 is idle, and more importantly, statuses 2-5 are available when it is executing positioning commands.

Statuses 1 and 2 above will be available only when the error condition occurs. Status #3 is available when an end of travel limit occurs and is considered an error. Status #3 is also available whenever you query the MC4 for its status.

When an error does occur, the MC4 will beep twice, very quickly, and store the descriptive error message into its output buffer. If your interface is IEEE-488, you must then issue a read statement (the exact command will depend on your IEEE-488 interface manufacturer) to the interface which will place the MC4 in the Talker mode. The MC4 will then send the contents of its output buffer to your computer.

If your interface is RS-232-C, the MC4 will automatically send the error message to your computer without you asking for it. However, this will occur only if:

- for the hardware handshake mode, your data terminal ready line is true
- for the software handshake mode, the MC4 is in the XON state. The MC4 will be in the XON state if you have not sent an XOFF or if you have sent an XOFF, and you subsequently sent an XON.

Statuses 3-6 are available whenever you wish to know them. This can be while the MC4 is idle, waiting for a command, or executing a positioning command or a wait command.

The steps necessary to have the MC4 report its status at any time you want are:

1. When the MC4 is idle, you must tell it which status options you want. You do this with the FS command. This command is explained below.

2. At the instant you want a status report, you do the following based on your interface and the processing state of the MC4:

   IEEE-488

   MC4 Idle:

   - first send the ? command
   - then issue a read statement to the interface to place the MC4 in the Talker mode

   MC4 Processing Command:

   - just issue a read statement to the interface to place the MC4 in the Talker mode. The MC4 assumes you want status information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RS-232-C

MC4 Idle or Processing Command:

- send the ? command and the MC4 will automatically send the status report.

6.4.1 STATUS FORMAT COMMAND

Before you can ask the MC4 its status, you must tell it what particular status you want. You do this by using the FS (format status) command. To the FS command you append a hexadecimal byte which represents the status information you want. If you want a particular status, you set that status' bit location to 1; otherwise you set it to 0. The following table shows the bit locations within the byte assigned to each status.

Hexadecimal Byte: B7 B6 B5 B4 B3 B2 B1 B0

B0 = 1 Report axis moving/stationary status
B0 = 0 Do not report axis moving/stationary status

B1 = 1 Report W and X axis end of travel limits true/false status
B1 = 0 Do not report W and X axis end of travel limits true/false status

B2 = 1 Report Y and Z axis end of travel limits true/false status
B2 = 0 Do not report Y and Z axis end of travel limits true/false status

B3 = 1 Report Logic level of 3 input bits
B3 = 0 Do not report Logic level of 3 input bits

B4 = 1 Report W axis display readout
B4 = 0 Do not report W axis display readout

B5 = 1 Report X axis display readout
B5 = 0 Do not report X axis display readout

B6 = 1 Report Y axis display readout
B6 = 0 Do not report Y axis display readout

B7 = 1 Report Z axis display readout
B7 = 0 Do not report Z axis display readout

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Examples:

1. We want axis moving/stationary status, the W axis display and the Y axis display.

   HEX byte: 0101 0001 ---> 51H

   MC4 Command: FS51

2. We want axis moving/stationary status, W and X end of travel limits, the W axis display, and the X axis display.

   HEX byte: 0011 0011 ---> 33H

   MC4 Command: FS33

3. We want all statuses

   HEX byte: 1111 1111 ---> FFH

   MC4 Command: FSFF

6.4.2 MC4 STATUS RESPONSE FORMAT

The MC4 status response is a string of coded bytes. Your application program will read this string into a string variable. It will then have to manipulate the contents of the string variable to extract the information contained within it.

To help you distinguish which status a particular status byte represents, each status byte is preceded by a header byte to identify it. Following the header byte is the status byte. The process for decoding the status string is to search the string for the header and then read the next byte as the status byte.

The form of the MC4 status report string is:

header1status1header2status2...headerNstatusN carriage return/line feed

As you can see, the header and status bytes are all bunched together and terminated by a carriage return/line feed.

- Moving/Stationary Status Byte

  Header Bit Pattern: 0011 0000 ---> ASCII 0

  Status Bit Pattern: 0100 B3 B2 B1 B0

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Status Bit Interpretation:

B0 = 1 W axis moving	B2 = 1 Y axis moving
B0 = 0 W axis stationary	B2 = 0 Y axis stationary
B1 = 1 X axis moving	B3 = 1 Z axis moving
B1 = 0 X axis stationary	B3 = 0 Z axis stationary

Example: You desire moving/stationary status (FS01) and the W axis is moving when you ask for status. The MC4 response will be the ASCII string 0ACR/LF; CR = carriage return; LF = line feed

- W and X axis End of Travel Limits

  Header Bit Pattern: 0011 0001 ---> ASCII 1

  Status Bit Pattern: 0100 B3 B2 B1 B0

  Status Bit Interpretation:

  B0 = 1 W axis plus end of travel limit true
  B0 = 0 W axis plus end of travel limit false

  B1 = 1 W axis minus end of travel limit true
  B1 = 0 W axis minus end of travel limit false

  B2 = 1 X axis plus end of travel limit true
  B2 = 0 X axis plus end of travel limit false

  B3 = 1 X axis minus end of travel limit true
  B3 = 0 X axis minus end of travel limit false

Example: In addition to moving/stationary status, assume you desire W and X axis end of travel limit (FS03). At the time you ask for status, W axis is moving and the X axis plus end of travel limit is true. The MC4 response will be the ASCII string: 0A1DCR/LF

- Y and Z axis End of Travel Limits

  Header Bit Pattern: 0011 0010 ---> ASCII 2

  Status Bit Pattern: 0100 B3 B2 B1 B0

  Status Bit Interpretation:

  B0 = 1 Y axis plus end of travel limit true
  B0 = 0 Y axis plus end of travel limit false

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B1 = 1 Y axis minus end of travel limit true
B1 = 0 Y axis minus end of travel limit false

B2 = 1 Z axis plus end of travel limit true
B2 = 0 Z axis plus end of travel limit false

B3 = 1 Z axis minus end of travel limit true
B3 = 0 Z axis minus end of travel limit false

Example: Building on the above status response, you also desire the Y and Z axis end of travel limits status (FS07). Assume the two previous statuses are the same and in addition, the Y axis minus end of travel limit is true. The MC4 response will be the ASCII string: 0A1D2BCR/LF

- 3 Bit Input Status

  Header Bit Pattern: 0011 0011 ---> ASCII 3

  Status Bit Pattern: 0100 B3 B2 B1 B0

  Status Bit Interpretation:

  B0 = 1 Input bit 1 high
  B0 = 0 Input bit 1 low

  B1 = 1 Input bit 2 high
  B1 = 0 Input bit 2 low

  B2 = 1 Input bit 3 high
  B2 = 0 Input bit 3 low

  B3 = 1 No information
  B3 = 0 No information

Example: In addition to all the above, you also desire 3 bit status (FS0F). Assume the previous three statuses are the same, and in addition, input bit 2 is low when you ask for status. The MC4 response will be the ASCII string: 0A1D2B3ECR/LF.

- Displays Status

  The display information is sent as ASCII characters, so your program should interpret it as such. Each displays numerical information is preceded by two ASCII characters that identify it. These are: W=, X=, Y=, and Z=.

  Header Bit Pattern:

  W axis display 0101 0111 ---> ASCII W
  X axis display 0101 1000 ---> ASCII X

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Y axis display 0101 1001 ---> ASCII Y
Z axis display 0101 1010 ---> ASCII Z

Status Bit Interpretation:

In this case, the individual bits following the header have no specific meaning. However, the bytes formed by these bits should be interpreted as numerical ASCII characters.

Example: In addition to all the above, you also desire the X and Y display information (FS6F). Assume all previous statuses are the same and, in addition, w-axis display is at +01234567 and the X axis display is at -07654321. The MC4 response will be the ASCII string:

0A1D2B3EW=+01234567 X=-07654321 CR/LF

- Wait Status

This status is enabled all the time. It cannot be selected or de-selected with a bit in the FS command. If the MC4 is in a wait state at the time you request a status report, a byte in the status response string will indicate this state. If the MC4 is not in a wait state, then only the programmed status bytes will be sent.

Header Bit Pattern: 0011 0100 ---> ASCII 4

Status Bit Pattern: 0100 B3 B2 B1 B0

Bit Interpretation:

B0 = 1 Executing B command: wait for number of steps to be completed
B0 = 0 Not executing B command

B1 = 1 Executing UE command: wait for external input level change
B1 = 0 Not executing UE command

B2 = 1 Executing UT command: wait for time out
B2 = 0 Not executing UT command

B3 = 1 No information
B3 = 0 No information

Example: Assume all above statuses are true and the MC4 is executing the UT command when you ask for status. The MC4 response will be the ASCII string:

4D0A1D2B3EW=+01234567 X=-07654321 CR/LF

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

NOTE: If the wait status is true when you ask for status, that information will be in the first two bytes of the status response string.

NOTE: If there is a status you do not desire, you inform the MC4 of this fact with the FS command. The resulting status report string from the MC4 will simply have the undesired status byte missing from the sequence. For example: You only desire moving/stationary, 3 bit input, and X axis and Y axis display status, and the MC4 is not in the wait state. The MC4 status response will be the ASCII string:

0A3EW=+01234567 X=-07654321 CR/LF

6.4.3 STATUS REPORT PRACTICE

If you followed the instructions in Chapter 4 or Chapter 5 and presently have a functioning interface between the MC4 and your computer, you are set to experiment with the status reporting feature of the MC4. If you study the practice application program in Chapter 11 (and currently in your computer) you will notice that starting at line number 900 or line 1000 for RS-232-C hardware handshake, is the software that decodes the status report string of the MC4. This is done by searching the bytes of the string for a header byte. Once found, the program branches to the routine that then decodes the information contained in the next byte. If the header indicates display information, then we simply treat the subsequent bytes as ASCII characters.

- **IEEE-488**

First command: FSF1 ;we want moving/stationary and all display status

Second command: ? ;MC4 response is that all axes are stationary and the
;display readout

Subsequent commands:

```
NW5000
NX3000
RA500
MW              ;start moving W
```

While W is moving, enter Q <RETURN> at your keyboard. (In this case, Q is not an MC4 command. It is used as a flag to cause the application program to branch to the software which places the MC4 in the Talker mode.)

MC4 response: W axis moving, all others stationary
W axis display readout as it is changing, all others no change

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

While W is moving and when it stops, enter Q <RETURN> many times.

MC4 response: Same as above, except the W display readout is changing.

When W stops, the MC4 will report that condition along with the final readout of the W display.

Next command: <MW;MX;> ;start moving W and X simultaneously; X will
;stop before W

Enter Q <RETURN> many times

MC4 response: At first it will indicate W and X moving, Y and Z stationary and the display readouts at the time you entered Q <RETURN>.

Then when X stops, it will indicate W moving, X,Y,Z stationary and the display readouts at the time you entered Q <RETURN>.

Finally when W stops, it will indicate W,X,Y,Z stationary and the final display readouts.

- **RS-232-C**

  The practice commands and sequence are the same as above for the IEEE-488, except, wherever it says to enter Q <RETURN>, enter ? <RETURN> instead.

  For RS-232-C, you must send the status request command (? command) when you want a status report.

  You may wish to experiment further by sending different parameters with the FS command and observing the MC4 status response.

6.4.4 ERROR CHECKING AND REPORTING

The MC4 will only execute valid commands and valid combinations of commands. A valid command is one that conforms to the specifications listed in chapter 7. If you send an invalid command, the MC4 will beep twice, very quickly, and store an error message in its output buffer. If your interface is IEEE-488, you should execute a read statement that will place the MC4 in the Talker mode. If your interface is RS-232-C, the MC4 will automatically send the error message. The format of the error message will be:

Enn ERROR MESSAGE Carriage Return/Line Feed

E is the header byte telling you an error message is coming

nn is the code number for the type of error.

ERROR MESSAGE is a description of the error

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

See chapter 12 for the listing of all possible errors and their code numbers.

Typical errors may include:

- numerical parameter too small or two large
- not specifying axis designator (W,X,Y,Z, or A)
- sending program entry and edit commands while in program entry mode
- sending an invalid command in the simultaneous and interdependent mode of execution. Only a subset of MC4 commands are allowed in these modes. Refer to section 6.2.2 for a listing of commands that may be used in these modes.

Examples:

Command: RW4500

MC4 response: Beep,beep: E02 PARAMETER ERROR

Command: WX

MC4 response: Beep,beep: E01 COMMAND ERROR

Command: <MW;RY2000;MY;>

MC4 response: Beep,beep: E01 COMMAND ERROR
(R is not allowed in simultaneous mode)

6.5 SERVICE REQUEST GENERATION

Service requests are only available for the IEEE-488 interface. It is part of the protocol.

The MC4 may be programmed to generate a service request signal when any of the following five events occur. These events are:

1. when the W axis completes its move
2. when the X axis completes its move
3. when the Y axis completes its move
4. when the Z axis completes its move
5. when any axis reaches a travel limit

MC4 commands M, P, and O are commands that initiate a move. Thus at the completion of any of these commands for any of the four axes, the MC4 can generate a service request.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The steps necessary to enable the MC4 to generate a service request and subsequently send a status byte to your computer for interpretation are:

1. When the MC4 is idle, you must tell it which of the five events listed above, you want to generate a service request. You will do this with the FI command. This command is explained below.

2. When a selected event occurs, the MC4 will generate a service request signal on the IEEE-488 interface. In response to this signal, your computer will output a serial poll command to the MC4.

3. In response to the serial poll command, the MC4 will send a status byte to your computer. The information represented by each bit of the status byte is described below.

6.5.1 SERVICE REQUEST ENABLE COMMAND

When any of the events occur that can generate a service request, the MC4 will check if it should output a service request signal. If this particular event is enabled, the MC4 will output a service request signal, otherwise it will not. You will use the FI command to enable and disable particular events to generate service requests. To the FI command, you will append a hexadecimal byte which represent enabled and disabled events. If you want a particular event to generate a service request, you set that event's bit location to 1, otherwise you set it to 0. The following table shows the bit locations assigned to each event.

Hexadecimal Byte: B7 B6 B5 B4 B3 B2 B1 B0

B0 = 1 Generate service request when W axis move is completed
B0 = 0 Do not generate service request when W axis move is completed

B1 = 1 Generate service request when X axis move is completed
B1 = 0 Do not generate service request when X axis move is completed

B2 = 1 Generate service request when Y axis move is completed
B2 = 0 Do not generate service request when Y axis move is completed

B3 = 1 Generate service request when Z axis move is completed
B3 = 0 Do not generate service request when Z axis move is completed

B4 = 1 Generate service request when any axis reaches a travel limit
B4 = 0 Do not generate service request when any axis reaches a travel limit

B5 - B7 Not presently assigned

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Examples:

1. We want to generate service requests whenever the W axis move is completed, the Y axis move is completed, or an axis reaches a travel limit.

   HEX byte: 0001 0101 ---> 15H
   MC4 command: FI15

6.5.2 MC4 STATUS BYTE FORMAT

After the MC4 generates a service request, your computer will need to poll all the devices on the bus to determine which device requested service. The MC4 will only respond to a serial poll command. It does not respond to a parallel poll. In response to a serial poll, the MC4 will output the following status byte:

Status Byte: B7 B6 B5 B4 B3 B2 B1 B0

B0 = 1 W axis move completed
B0 = 0 W axis idle or moving

B1 = 1 X axis move completed
B1 = 0 X axis idle or moving

B2 = 1 Y axis move completed
B2 = 0 Y axis idle or moving

B3 = 1 Z axis move completed
B3 = 0 Z axis idle or moving

B4 = 1 An axis has reached a travel limit
B4 = 0 No axis has reached a travel limit

B5 = X Don't care - no information

B6 = 1 MC4 generated the present service request signal
B6 = 0 MC4 did not generate the present service request signal

B7 = X Don't care - no information

Example:

In response to a serial poll command, the MC4 returns the following status byte: 0100 0010. This means that the MC4 did generate a service request because the X axis service request was enabled with the FI command and the X axis move was just completed.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 7: DETAILED DESCRIPTION OF MC4 COMMANDS

In this chapter we will first show a summary of MC4 commands grouped according to associated function. Then we present each command in alphabetical order and list everything you should know about it.

7.1.1 SUMMARY OF MC4 COMMANDS

1. **Motor Control**

   - Commands for setting stepping rate

     R program stepping rate from 32 to 4000 steps/second

     T program step intervals from 2 milliseconds to 100 seconds

   - Commands for relative positioning

     +/- set direction of movement

     N set number of steps for the next move

     M start move

   - Commands for absolute positioning

     A set present position as reference point; also resets front panel display

     P move to location specified relative to the defined reference point

   - Command for programming acceleration

     AC

   - Origin search

     O

2. **Wait for Event**

   B wait for motor to finish moving a predefined number of steps

   UT wait a predefined number of milliseconds (time delay)

   UE wait for a high or low condition of an input

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. **Status Report/Request**

   FS program status options desired

   ? report status

   D read front panel display (i.e. position of motors)

4. **Service Request**

   FI program conditions which will generate service requests

5. **Output Control**

   C set output bit low

   S set output bit high

6. **Stepping and Holding Torque Inhibit/Enable**

   K

7. **Define and Execute Variable Parameters**

   V

8. **Simultaneous and Interdependent Command Execution**

   <command;command; ... ;command;>

9. **Stored Program Editing Commands**

   E enter program entry mode

   QP end of a single program

   QE exit program entry mode

   I insert a program line

   # delete program lines

   L list program lines

   $ comments inside programs

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

10. Stored Program Execution Command

GI

11. Jump Commands

H define a "jump to" label

J jump to the specified label

7.1.2 ALPHABETICAL LISTING AND DESCRIPTION OF COMMANDS

COMMAND: A (W, X, Y, Z, A)

FUNCTION:
1. Reset display counter(s) to zero
2. Set present position as the floating origin point
3. Set present position as the reference point for the absolute positioning command (i.e., P command)

EXAMPLE:
AX ; perform functions 1-3 on the X axis
AA ; perform functions 1-3 on all axes

COMMAND: AC (W, X, Y, Z)n.n

FUNCTION: Program acceleration ramp from low speed to high speed.

RANGE: n.n = 0.1 second - 2 seconds

COMMENTS: The acceleration command parameter represents the time the MC4 will accelerate from 50 Hz to 4000 Hz. Thus if you program a stepping rate lower than 4000 Hz that stepping rate will be reached faster than the time specified in the AC command. As you vary the time to ramp to 4000 Hz you also vary the time to ramp to all lower stepping rates.

The acceleration ramp is always exponential. This means that it is steep at the beginning and becomes asymptotic as it approaches 4000 Hz. For example, if you program ACW2, and your desired high speed rate is 2000 Hz, the MC4 W axis will ramp to 2000 Hz in less than 1 second.

Each axis can be programmed for its own acceleration ramp.

The programmed acceleration is valid for all modes of motion: program, command and manual modes of operation.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This command can not be executed in simultaneous execution mode. i.e., <MW;MX;ACY1.5;MY;MZ;>

The AC command parameter is preserved during power-down. Consequently, the desired acceleration does not have to be reissued after power-up. (Default is all axes are 0.4)

EXAMPLE:
```
ACW.1   ;ramp to 4000 Hz in 0.1 seconds on the W axis
ACX1.7  ;ramp to 4000 Hz in 1.7 seconds on X axis
```

COMMAND: B (W, X, Y, Z) nnnnnnn

FUNCTION: Wait for the specified axis to complete nnnnnnn steps and then start executing the next command

RANGE: nnnnnnn = 1 to 7999999

COMMENTS: This command allows you to coordinate the movement of several axes. It also can be used, in conjunction with the C and S command, to signal the execution of n number of steps along the whole movement.

EXAMPLE:
```
<MW;BW500;MX;>
```
- start W, when W reaches 500 full steps, start X

```
<MW;BW100;C1;S1;BW200;C1;S1;BW300;C1;S1;>
```
- start W, when W reaches 100 full steps, pulse output #1, when W reaches 200 full steps, pulse output #1, etc.

COMMAND: C (1, 2, 3, 4, 5, 6, 7, 8)

FUNCTION: Set specified output bit number low.

RANGE: Output bits are numbered from 1 to 8

COMMENTS: The output bits are available on the MC4 General Purpose Input/Output (GPIO) connector. Refer to chapter 8 for the pin-outs. These outputs are open-collector, so you must place a pull-up resistor (value: 1K ohm to 10K ohm) between a power supply and the output pin. For the power supply, you may use the +5 volts or the +12 volts available on the GPIO connector.

EXAMPLE:
```
C1              ; set output number 1 low

<C6;C7;C8;>     ; set outputs number 6,7, and 8 low
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

COMMAND: D (W, X, Y, Z, A)

FUNCTION: Read front panel display and store data into output buffer

COMMENTS: For IEEE-488 interface, you must issue a read statement to the interface to place the MC4 into the Talker mode. At that time the MC4 will send the contents of its output buffer to your computer.

For RS-232-C interface, the MC4 will automatically send the display data to your computer in response to the D command.

The format of the display data sent by the MC4 is:

- in response to DW: W = ±xxxxxxxx
- in response to DY: Y = ±xxxxxxxx
- in response to DA: W = ±xxxxxxxx X = ±xxxxxxxx
  Y = ±xxxxxxxx Z = ±xxxxxxxx

All characters sent are ASCII coded, including the 3 spaces between each display readout.

The response string is always terminated by a carriage return/line feed combination regardless of the setting of switch S3-4, which only tells the MC4 how its commands are terminated.

EXAMPLE:
```
DX      ;read X axis display
DA      ;read all axes displays
```

COMMAND: Ennnn

FUNCTION: Enter program entry mode at line number nnnn

RANGE: nnnn = 0 to 9999

COMMENTS: Line numbers less than nnnn remain unchanged

Line numbers greater than or equal to nnnn are destroyed.

If nnnn is greater that the number of existing program lines, then command storage will begin at the last program line. The last program line will be QE command line number. This will be overwritten with the first command of the new program.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

If nnnn is not specified, nnnn defaults to 0000. Command storage will start at line number 0000. This will erase all previously stored programs.

You should normally request a listing of your existing programs before issuing the E command. By sending the L command, the MC4 will send a listing of existing programs to your computer. The listing consists of line numbers and commands. By seeing the position of your other programs, you will be able to specify the correct line number for the E command.

CAUTION: If you accidentally enter a wrong line number or forget one, you may be in danger of erasing good programs.

DO NOT enter QE, hoping to exit and try again. Instead, press the STOP switch on the MC4 front panel. This will take you out of program entry mode safely and you may then enter the E command with the proper line number.

EXAMPLE:
```
E      ;enter program entry mode at line number 0000
E20    ;enter program entry mode at line number 0020
```

COMMAND: FShh

FUNCTION: Program the MC4 for status options desired

RANGE: hh = 00H to FFH

COMMENTS: The parameter hh is HEX data where the value (1 or 0) of each bit of its binary equivalent represents your desire or non-desire of the particular status that it represents.

hh byte = B7 B6 B5 B4 B3 B2 B1 B0

B0 = 1 want axis moving/stationary status
B0 = 0 do not want axis moving/stationary status

B1 = 1 want W and X end of travel limits status
B1 = 0 do not want W and X end of travel limits status

B2 = 1 want Y and Z end of travel limits status
B2 = 0 do not want Y and Z end of travel limits status

B3 = 1 want 3 bit input status (E1, E2, and E3 on GPIO)
B3 = 0 do not want 3 bit input status

B4 = 1 want W axis display readout
B4 = 0 do not want W axis display readout

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B5 = 1 want X axis display readout
B5 = 0 do not want X axis display readout

B6 = 1 want Y axis display readout
B6 = 0 do not want Y axis display readout

B7 = 1 want Z axis display readout
B7 = 0 do not want Z axis display readout

You must send this command while the MC4 is idle. This is because the MC4 will not accept commands while it is busy executing another command. However, while executing a command, the MC4 will respond to a status request command.

EXAMPLE: FSF1 ;we want axis moving/stopped status and all displays

FS66 ;we want all axis end of travel limits status and the X and Y displays

---

COMMAND: FIhh

FUNCTION: Program/select events that will generate service request signal

RANGE: hh = 00H to FFH

COMMENTS: The parameter hh is HEX data where the value (1 or 0) of each bit of its binary equivalent represents your desire or non-desire to have a particular event to generate a service request.

hh byte: B7 B6 B5 B4 B3 B2 B1 B0

B0 = 1 Generate service request when W axis move is completed
B0 = 0 Do not generate service request when W axis move is completed

B1 = 1 Generate service request when X axis move is completed
B1 = 0 Do not generate service request when X axis move is completed

B2 = 1 Generate service request when Y axis move is completed
B2 = 0 Do not generate service request when Y axis move is completed

B3 = 1 Generate service request when Z axis move is completed
B3 = 0 Do not generate service request when Z axis move is completed

B4 = 1 Generate service request when any axis reaches a travel limit
B4 = 0 Do not generate service request when any axis reaches a travel limit

B5 - B7 Not presently assigned

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE: We want to generate service requests whenever the W axis move is completed, the Y axis move is completed, or an axis reaches a travel limit.

HEX byte: 0001 0101 ---> 15H
MC4 command: FI15

COMMAND: GInnnn

FUNCTION: Start executing stored program commands at line number nnnn

RANGE: nnnn = 0 to 9999

COMMENTS: The MC4 will start executing commands from line number nnnn and continue until it encounters a QP command.

nnnn can be any line number containing commands. It does not have to be the first command of a program.

If nnnn is not specified, nnnn defaults to 0000

OPTION: GInnnn-mm

- execute mm lines of commands starting with line number nnnn

EXAMPLE:
```
GI      ;start executing commands at line number 0000
GI27    ;start executing commands at line number 0027
GI30-5  ;execute 5 lines of commands starting at line number 0030
```

COMMAND: H (A, B, C, ... Z)

FUNCTION: Define a "jump to" label

RANGE: 26 different labels may be defined from A to Z

COMMENTS: This command marks a program location as the destination for the jump (J) command. The H and J commands enable looping for repetitive execution of the same commands.

To use this command in real-time mode, it must be enclosed in arrowhead brackets along with the other commands. See examples in the J command description section.

To use it as part of a stored program, it may be enclosed in arrowhead brackets or it can be on a separate line. See examples in the J command description section.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:
```
HA      ;defined this location as A
HS      ;defined this location as S
```

COMMAND: Innnn

FUNCTION: Insert a line of command(s) at line number nnnn

RANGE: nnnn = 0 to 9999

COMMENTS: This command is used to edit your existing stored programs without having to re-type all the commands of that program.

After you issue this command, the MC4 will then wait for your next command and store it at line number nnnn when received.

All commands that were previously below line number nnnn, will be shifted one line number lower to make room for the new command.

If nnnn is greater than the number of existing program lines, the MC4 will place your command as the last command of the last program.

After you perform this operation, it is wise to do ask for a listing of the programs to verify that the MC4 did it correctly.

EXAMPLE:
```
I20         ;insert the next line of command(s) at line # 20
MW;MX       ;these two commands will now be stored at line # 20
```

COMMAND: J (A, B, C, ... Z) nn

FUNCTION: Jump to specified label nn number of times and continue to execute commands from that location.

RANGE: nn = 0 to 99

COMMENTS: If nn = 0 or is not specified, the MC4 will loop indefinitely. To exit this endless loop, you can either press the STOP switch on the front panel or turn the MC4 off.

The same label may be used in the same program as long as they are not nested.

The number of times that a set of commands inside a loop will be executed is nn+1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE: **Real-Time Mode:**

1. <HA;MW;MX;JA5;>

   This is the only construct allowed in real-time mode. The above command will move W and X at the same time 6 times. However, if you want to move W and then when W is done, move X, then the command would be: Assume that NW1000 was previously sent.

2. <HA;MW;BW999;MX;JA5;>

3. <HD;MW;C1;S1;JD7;>

   - start moving W and at the same time, pulse output number 1
   - repeat 7 times for a total of 8 executions

4. <HD;MW;BW999;C1;S1;JD7;>

   - start moving W, when W done, pulse output number 1
   - repeat 7 times for a total of 8 executions

5. <HT;MW;UT1000;MX;JT;>

   - start moving W and executing the time delay at the same time; the time delay is for 1 second; 1 second after W started moving, move X
   - repeat the process indefinitely

6. <HT;MW;BW999;UT1000;MX;JT;>

   - start moving W; 1 second after W is done, move X
   - repeat indefinitely

Program Mode:

We will now show how the above examples 1-6 are done in the program mode.

1. <HA;MW;MX;JA5;>

2. HA;MW;MX;JA5

3. <HD;MW;C1;S1;JD7;>

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. HD;MW;C1;S1;JD7

5. <HT;MW;UT1000;MX;JT;>

6. HT;MW;UT1000;MX;JT

COMMAND: J(A,B,C...Z)U(1,2,3)(H,L)

FUNCTION: Jump to the specified label (U)ntil the designated bit goes to the programmed level: (H)igh or (L)ow

COMMENTS: This command enables the length of execution of a motion command or a program to be controlled by an external input signal.

EXAMPLE:
```
NX1     Set number of steps equal to 1
HA      Define a "jump to" label
MX      Move X-axis
JAU1L   Jump to label "A" until bit #1 goes low
.
.
.
```

COMMAND: K (W, X, Y, Z, A) E / D

FUNCTION: Enable or disable holding torque to the motor

COMMENTS: This command can be used to disable current flow to the motor when it is stationary. With no current flow, the motor will not dissipate any heat. This may be essential in a vacuum system.

If you disable holding torque while the motor is stationary, you must enable it before you make the next move.

EXAMPLE:
```
KWE     ;enable holding torque on W axis
KXD     ;disable holding torque on X axis -- no current -- no heat
```

COMMAND: Lnnnn

FUNCTION: Prepare a listing of currently stored programs from line number nnnn and store it in the output buffer

RANGE: nnnn = 0 to 9999

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

OPTIONS:

Lnnnn	;list commands at line number nnnn
Lnnnn-mmmm	;list line numbers from nnnn to mmmm
Lnnnn-	;list line numbers from nnnn to the end
L-nnnn	;list line numbers 0000 to nnnn

COMMENTS: **For IEEE-488 interface,**

- you must issue a read statement to the interface to place the MC4 into the Talker mode. At that time the MC4 will send the contents of its output buffer to your computer.
- the MC4 sends one line of commands at a time, followed by a carriage return/line feed combination regardless of the setting of switch S3-4 which only tells the MC4 how its commands are terminated. Your interface should be programmed to terminate the read operation when it receives the carriage return/line feed. Since a listing will normally be more than one line, you will have to put your read statement in a program loop so that it is continually executed until the MC4 has sent all line listings.
- After the MC4 has sent the last line of the program listing, it will then send the ASCII characters END in response to the next read statement. Your application program will use this flag to exit the read loop.

For RS-232-C interface,

- the MC4 will automatically send the listing to your computer in response to the L command.
- if the program listing is coming too fast for you computer to handle, you can cause the MC4 to stop until your computer catches up:

  - Hardware Handshake: Set Data Terminal Ready false

  - Software Handshake: Send XOFF, then send XON when ready

EXAMPLE:

```
L       ;list all programs
L10     ;list line number 10
L5-12   ;list lines 5 through 12
L-20    ;list lines 0 through 20
L30-    ;list lines 30 to the end
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

COMMAND: M (W, X, Y, Z)

FUNCTION: Start move on designated axis

COMMENTS: After you have set up your movement parameters (i.e. programmed the number of steps with the N command, set the stepping rate with the R or T command, and defined the direction with the + or - command) you issue the M command to perform the move.

EXAMPLE:
```
MW      ;start moving W axis
MZ      ;start moving Z axis
```

COMMAND: N (W, X, Y, Z) nnnnnnn.d

FUNCTION: Set the number of steps to be executed on the next move for the designated axis

RANGE: nnnnnnn = 1 to 7999999 ; d = 0 to 9

COMMENTS: In addition to the comments below, it is strongly advised that you refer to Chapter 10: MC4 Theory of Operation at this time to fully understand the N command.

The parameter nnnnnnn specifies the total number of full step you desire for the next move.

The parameter d specifies the fraction of a full step you desire in addition to the number of full steps.

It is possible to specify move lengths of less than one full step i.e. nnnnnnn = 0 and d = fraction of full step desired

If d = 0, it does not have to be specified.

The parameter d is not valid for the UE30 motor type. This motor cannot be moved a fraction of a full step. If you specify a fraction of a full step, the MC4 will signal an error and send the following error message: E03 PARAMETER/MOTOR ERROR

If the number of steps from move to move is the same, you do not have to re-program the N command. You only have to re-program the N command when the number of steps for the next move is different.

EXAMPLE:
```
NW2000      ;program W axis for 2000 full steps
NX3000.7    ;program X axis for 3000.7 full steps
NY.4        ;program Y axis for 0.4 full steps
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

COMMAND: O (W, X, Y, Z)

FUNCTION: Initiate an origin search on the designated axis

COMMENTS: The MC4 will search for and locate the origin point defined by its axis configuration switches. Refer to Chapter 2 for the complete discussion of the axis configuration switches.

The three origin point options are:

1. floating origin defined by location where the front panel display equals zero.
2. switch mounted on a positioning stage
3. switch mounted on a positioning stage in conjunction with a top zero pulse

The MC4 cannot perform origin search on more than one axis at a given time. If this is an inconvenience for you, a possible alternative is described below in the P command section.

EXAMPLE:
```
OW      ;initiate origin search on W axis
OY      ;initiate origin search on Y axis
```

COMMAND: P (W, X, Y, Z, A) nnnnnnn.d

FUNCTION: Move to the position specified by nnnnnnn.d relative to a predefined reference point

RANGE: nnnnnnn = -7999999 to 7999999 ; d = 0 to 9

COMMENTS: In addition to the comments below, it is strongly advised that you refer to Chapter 10: MC4 Theory of Operation at this time to fully understand the P command.

The P command is used to tell the MC4 your desired final position relative to a predefined reference point. The reference point is defined by the A command and by the reset switch on the front panel.

The P command is referred to as the absolute positioning command. It alone performs all three commands used for relative positioning.

- by comparing the present (pre-move) position and the specified parameter nnnnnnn.d it knows the direction of movement (+ or - command) and the number of steps (N command).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- when the above is calculated, it initiates the move (M command)

The parameter d is not valid for the UE30 motor type. This motor cannot be moved a fraction of a full step. If you specify a fraction of a full step, the MC4 will signal an error and send the following error message: E03 PARAMETER/MOTOR ERROR.

IMPORTANT If your UE7x stepping motor has a 200 point encoder and you have selected the encoder to drive the front panel display, you will experience the following peculiarity if you issue P commands that specify a fraction of a full step.

The P command uses the front panel information to read the present position so that it can calculate how many steps it must execute to reach the target position. Since a 200 point encoder cannot show fractions of a full step, the MC4 will assume that the fraction of a full step is zero, i.e. the present position is on a full step boundary. However, that will not be the case if you previously issued a P command with a fraction of a full step. This will cause a slight calculation error, but the final position will always be within one full step.

EXAMPLE: If you send PW100.7, the counter may read the final position as 100 or 101, depending on the alignment of the encoder. If next you send PW200.3, the MC4 will calculate a displacement of 100.3 instead of 99.6. The final position on the display will be 201.

If this is a inconvenience, you may switch the encoder count off and let the display be driven by the internal pulses. The internal pulses simulate a 2000 point encoder and thus all MC4 calculations will be correct.

The PA command will move all axes that have a motor connected to them. Any axis that does not have a motor connected to it (indicated by the illumination of both the +LIMIT and -LIMIT light bars) will be bypassed.

The MC4 cannot perform origin search on more than one axis at a time. This may be an inconvenience if you wish to do origin search for several stages and they are far from the origin point. A handy solution to this dilemma is to use the PA command as follows:

- at the beginning of your application, perform origin search sequentially on each axis.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- issue the AA command to set the present position as the reference point for the P command
- when you wish to do origin search on all axes, issue the PA0 command
- when the PA command is done, issue the origin search command to each axis. Even though your system will be at the origin point when PA0 is complete, you should perform origin search with the O command. The execution of origin search is such as to compensate for the backlash of the lead screw of the positioning stage.

EXAMPLE:

```
PX3000      ;go to position +3000 relative to zero on the X axis
PY-1000     ;go to position -1000 relative to zero on the Y axis
PA200       ;move all stages/motors connected to the MC4/MD4 to
             position 200
```

COMMAND: QE

FUNCTION: Exit program entry mode

COMMENTS: This command is used when you are finished entering stored program commands. It returns the MC4 back to real-time mode.

As you will recall, you can enter several commands on the same line number. However, the QE command must stand alone.

A QE will automatically place a QP before the end of the program.

COMMAND: QP

FUNCTION: Define the end of a program of commands

COMMENTS: This command signals the MC4 that it has reached the end of the program of commands it was executing. At this time, it will exit program execution mode and enter real-time mode.

Like the QE command, this command must be alone on a program line.

COMMAND: R (W, X, Y, Z, A) nnnn

FUNCTION: Set the stepping rate (steps / second) for the designated axis

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RANGE: nnnn = 32 to 4000

COMMENTS: The stepping rate nnnn is defined in full steps/second. However, the MC4 pulse output rate will be 10 times nnnn. Refer to Chapter 10: MC4 Theory of Operation.

After you have programmed the stepping rate, you initiate the move at that speed by issuing the M or P command.

EXAMPLE:
```
RW1000      ;set stepping rate at 1000 full steps/sec. on W axis
RX3000      ;set stepping rate at 3000 full steps/sec. on X axis
```

COMMAND: S (1, 2, 3, 4, 5, 6, 7, 8)

FUNCTION: Set designated output bit high

RANGE: There are 8 output bits which can be set high and set low

COMMENTS: The output bits are available on the MC4 General Purpose Input/Output (GPIO) connector. Refer to chapter 8 for the pin-outs. These outputs are open-collector, so you must place a pull-up resistor (value: 1K ohm to 10K ohm) between a power supply and the output pin. For the power supply, you may use the +5 volts or the +12 volts available on the GPIO connector.

EXAMPLE:
```
S1              ;set output bit #1 high
<S3;S5;S7;>     ;set output bits #3,5,7 high
```

COMMAND: T (W, X, Y, Z, A) nnnn

FUNCTION: Set the stepping rate by defining the time delay between full steps, i.e., program the full step period

RANGE: nnnn = 2 to 9999

COMMENTS: period = nnnn times 10 milliseconds

This command is similar to the R command because they both set the stepping rate. However, the R command parameter defines the stepping rate as a step pulse frequency, while the T command defines the stepping rate by specifying the full step period.

After you have programmed the stepping rate, you initiate the move at that speed by issuing the M or P command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE:

```
TX500   ;set stepping rate at one full step every 5 seconds on
        ;the X axis

TY9999  ;set stepping rate at one full step every 99.99 seconds
        ;on the Y axis
```

COMMAND: UE (1, 2, 3) H / L

FUNCTION: Wait for the designated input pin to be in the specified logic level before proceeding to execute the next command.

COMMENTS: The 3 external inputs are connected to the MC4 through the General Purpose I/O connector.

This command may be used in the MC4 interdependent mode of operation whereby you specify the command(s) to be executed when the logic state becomes true and enclose them all in arrowhead brackets.

However, you can also use this command to simply "freeze" your interface. This is done by sending the command all by itself. When the MC4 is waiting for the logic level programmed, it outputs the following signals to the interface:

IEEE-488

- not ready for data (NRFD true)

RS-232-C

- hardware handshake: clear to send false
- software handshake: it sent an XOFF and will not send an XON

The freezing of the interface is done by sending a dummy command (like C1, S2, etc.) to the MC4 while it is executing the UE command. Your interface and consequently your application program will "hang" until the programmed logic state is attained. At that time, the MC4 will accept and execute the dummy command, but more importantly, your application program will proceed to execute the next statements.

As you can see, the UE command can be used to synchronize the execution of your application program with the logic levels of up to three external signals.

The logic state of these three bits may be read with a request for status command.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

EXAMPLE: <MW;UE1L;MX;>

- start W axis move; wait for input #1 to go low, then start moving on X axis

<MX;UE3H;S1;>

- start X axis move; wait for input #3 to go high, then set output #1 high

```
UE2L    ;"freeze" interface until input #2  goes low
S1      ;then accept and execute S1
TAKE MEASUREMENT ;then application program is enabled to
                 proceed to perform the next task like take
                 a measurement,  enable a device, etc.
```

COMMAND: UTnnnn

FUNCTION: Wait the specified time delay before executing the next command

RANGE: nnnn = 1 to 9999

COMMENTS: Delay = nnnn times 1 millisecond

Similar to the UE command above, this command may be used in the MC4 interdependent mode of operation and also to "freeze" your interface. Refer to the UE command discussion on these subjects.

EXAMPLE: <MW;UT100;MX;>

- start W axis move; 1 second after W started, start X axis move

```
NW1000
<MW;BW999;UT200;MX;>
```

- start W axis move; 2 seconds after W is done, start X axis move

```
NW1000
MW;UT200;MX
```

- this is how the above example is done when entered as part of a stored program

```
UT5000 ;"freeze" interface for 5 seconds
S1      ;then accept and execute S1
TAKE MEASUREMENT ;the application program is enabled to proceed
                 to perform the next task like take a
                 measurement, enable a device, etc.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

COMMAND: V (0, 1, 2, 3, 4, 5, 6, 7)

FUNCTION: Define an MC4 command parameter as a variable

RANGE: 8 different variables may be defined

COMMENTS: The MC4 allows you to define variable parameters for some of its commands. This enables you program repetitive tasks with different parameters in a few lines of program code. The procedure for use of this command is as follows:

1. Define and initialize the variable. The format for variable definition is:

   Vm=nnnn m = 0, 1, 2, 3, 4, 5, 6, or 7, and nnnn = initial parameter value

2. The variable can then be incremented or decremented. The format is:

   Vm=Vm+pppp m = variable number, and pppp = increment/decrement value

3. When the variable has been assigned a new value, it is used as a parameter for an MC4 command. See examples below.

   The following commands allow variable definition as the numerical parameter:

   B :wait for specified number of steps to be completed
   J ;jump to specified label "n" times
   N :set number of step pulses
   P :set number of absolute position
   R ;set step pulse rate

   This command is best used as part of a stored program. When used in a stored program, all the above commands are valid. However, only the B and J commands are valid in real-time mode of operation.

EXAMPLE: **Real-Time Mode or Program Mode**

```
NW1000
NY10
<V0=100;MW;HA;BWV0;MY;V0=V0+100;JA9;>
```

- set number of steps for W axis to 1000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

- set number of steps for Y axis to 10
- initialize variable to 100 and start moving W axis
- when W reaches 100 steps, move Y axis 10 steps
- increment variable to 200
- jump back, wait for W to reach 200 steps, then move Y axis again 10 steps, etc.

<V0=50;MW;HA;BWV0;C2;S2;V0=V0+50;JA19;>

- pulse output bit #2 every 50 steps of the W axis move.

Program Mode:

```
NW2000
V1=0              ;increase the step rate by 200  steps/second
HB                ;on each successive move
V1=V1+200
RWV1
MW
JB4
V3=10000
HC
PYV3              ;decrement the absolute position by 1000
UT50              ;for each successive move
V3=V3-1000
JC9
```

COMMAND: + (W, X, Y, Z)

FUNCTION: Set the direction for the next relative move to plus

EXAMPLE:
```
+W
+Z
```

COMMAND: - (W, X, Y, Z)

FUNCTION: Set the direction for the next relative move to minus

EXAMPLE:
```
-X
-Y
```

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

COMMAND: #nnnn

FUNCTION: Delete stored program lines

RANGE: nnnn = 0 to 9999

COMMENTS: This command is used to edit stored programs. It allows you to delete a single line of commands or several lines.

OPTIONS:

#nnnn	;delete line number nnnn only
#nnnn-mmmm	;delete lines nnnn through mmmm
#nnnn-	;delete lines nnnn to the end
#-nnnn	;delete lines 0000 through nnnn

EXAMPLE:

#10	;delete line number 10
#20-30	;delete lines 20 through 30
#50-	;delete lines 50 to the end

COMMAND: ?

FUNCTION: Obtain MC4 status and store it into the MC4 output buffer

COMMENTS: Before issuing this command, you will use the FS command to tell the MC4 which status option you desire.

The following comments are interface specific:

IEEE-488

- you can only issue this command when the MC4 is idle, i.e., not executing any command.
- after sending this command, you must issue a read statement to the interface that will place the MC4 in the Talker mode. At that time, the MC4 will send the contents of its output buffer.
- if you want a status report when the MC4 is executing a command, you do not send the ? command. You simply issue a read statement to the interface to place the MC4 in the Talker mode. The MC4 assumes you want status.
- The response string is always terminated by a carriage return/line feed combination regardless of the setting of S3-4 which only tells the MC4 how its commands are terminated.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

RS-232-C

- you issue this command when the MC4 is idle and when it is executing a command.
- the MC4 will automatically send the status information to your computer in response to the ? command.
- the MC4 will send the status information only if it senses that your computer is ready to receive.
  - hardware handshake: set Data Terminal Ready true
  - software handshake: if you previously sent XOFF, send XON

EXAMPLE: For examples and the status response format see section 6.4

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 8: MC4 GENERAL PURPOSE INPUT/OUTPUT CONNECTOR

The functions provided on the input pins of this connector are:

1. duplication of every switch of the MC4 front panel. By grounding an input pin, the same result is obtained as if you pressed a switch on the front panel of the same designation. This feature may be used when you want front panel control but the front panel is not within reach.

2. the external inputs for the UE command. As you will recall, this command tests the logic level (high or low) of an external input. Based on that information, it will proceed or wait for the programmed logic level.

The functions provided on the output pins of this connector are:

1. positive and negative step pulses for each axis. These pulses are in parallel with the step pulses that drive the MD4.

2. 8 programmable output bits controlled by the C and S command.

3. +5 volts, +12 volts and ground.

PIN#:	FUNCTION:	PIN#:	FUNCTION:
1	+ direction motion enable	19	B1 (programmable output bits)
2	- direction motion enable	20	B2 controlled by C2/S2 cmd.
3	High speed enable	21	B3 controlled by C3/S3 cmd.
4	Origin search enable	22	B4 controlled by C4/S4 cmd.
5	Axis select	23	B5 controlled by C5/S5 cmd.
6	W axis + output pulse	24	B6 controlled by C6/S6 cmd.
7	W axis - output pulse	25	B7 controlled by C7/S7 cmd.
8	X axis + output pulse	26	B8 controlled by C8/S8 cmd.
9	X axis - output pulse	27	E1 External input for UE1x cmd.
10	Y axis + output pulse	28	E2 External input for UE2x cmd.
11	Y axis - output pulse	29	E3 External input for UE3x cmd.
12	Z axis + output pulse	30	Emergency stop
13	Z axis - output pulse	31	Program start
14	Program Select	32	Program pause
15	W axis display reset	33	Local Lockout
16	X axis display reset	34	+12V (50 mA)
17	Y axis display reset	35	+5V (100 mA)
18	Z axis display reset	36	Logic ground
		37	System reset

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Direction Levels: High = positive direction Low = negative direction

All inputs are internally pulled up to +5 volts.

All outputs are open collector to 30 volts. To use an output, you must provide a pull-up resistor (1K ohms to 10 K ohms) between the output pin and a voltage supply.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 9: FRONT PANEL POSITION DISPLAY

The front panel position display is in units of a full step of a stepping motor. A full step is the distance between two adjacent stator poles. A ministep is a fraction of a full step. The MC4/MD4 system is a ministep system for permanent magnet motors and a full step system for variable reluctance motors. The UE7x series motors are permanent magnet and the UE30 motor is variable reluctance.

For the permanent magnet motors, the resolution is one-tenth of a full step, i.e., a full step is divided into 10 equal increments. If your display is being driven by a ministep encoder (i.e., 2000 points/revolution) or you selected the no encoder option (refer to Chapter 2) the display will change by .1 for each single step actuation from the front panel. Also the smallest displacement you can program is, for example, NW.1. If you issue a MW command after sending the NW.1 command, the display will change by .1.

If your display is being driven by a full step encoder, the decimal point will not be present. The display will change when the motor has moved a full step, i.e., 10 ministeps. This means that you will need to single step the MC4 10 times for the display to register one step. In programming mode, if you set NW.1 you will need to issue 10 MW commands for the display to register one step. Similarly, you can issue NW1 and a single MW command to achieve the same thing. With NW1 the MC4 will automatically generate 10 step pulses in response to MW.

For the variable reluctance motor, UE30, the resolution is a full step. Each single step actuation will change the display by 1. If the display is being driven by an encoder, the decimal point will not be present. If you do not have an encoder, then the display will show the decimal point, but the number to the right of the decimal point will always be zero. Commands like NW.1 are not allowed for UE30 motors. The MC4 will signal an error if you try this.

In all cases, to obtain the total displacement, simply multiply the front panel readout with the step size of your stage.

DISPLACEMENT = DISPLAY x STEP SIZE

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 10: MC4 THEORY OF OPERATION

The MC4 is basically an intelligent pulse source. When connected to a stepping motor driver, like the MD4, it becomes an intelligent stepping motor controller. Its command structure and front panel controls are tailored for stepping motors and stepping motor systems. As a result, we will refer to the pulses produced by the MC4 as step pulses for the remainder of this chapter.

The following step pulse attributes can be controlled:

1. frequency
2. quantity
3. direction
4. time

In the following discussion we will use the term "full step". A full step is the rotational distance between two stator poles of a stepping motor. Stepping motors are usually specified as n steps per revolution. Our UE30 stepping motor is 48 steps/revolution and our UE7x series stepping motors are 200 steps/revolution.

The paragraphs below that mention ministepping and moving fractions of a full step, pertain only to the UE7x series motor types. These are permanent magnet type stepping motors and thus they can be ministepped. The UE30 stepping motor is a variable reluctance type stepping motor and thus should not be ministepped. This motor should only be operated in the full step mode.

10.1 FREQUENCY

The frequency of the step pulses is 10 times the programmed rate. If you program a rate of 1500 full steps/second, either from the front panel or with the R command, the step pulse frequency will be 15,000 Hz.

The MC4 assumes that it is driving a stepping motor driver capable of ministepping a stepping motor. In particular, it assumes that the stepping motor driver is set to divide each full step into 10 ministeps. As a result, each step pulse from the MC4 will correspond to a motor rotation of one-tenth of a full step.

When programming a desired stepping rate, the parameter specifies full steps per second. The parameter for the R command specifies full steps per second, and the displayed rate when using the front panel also represents full steps per second.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Since a stepping motor will move only one-tenth of a full step for each step pulse from the MC4, it is necessary for the MC4 to supply step pulse at a frequency of 10 times the programmed rate in order to achieve the desired stepping rate.

As we mentioned above, the MC4 is basically a pulse source which assumes that it is supplying step pulses to a stepping motor driver that it set to rotate a stepping motor one-tenth of a full step for each pulse received. However, the stepping motor driver does not have to be set for one-tenth step division. If it is set for another step division, some useful effects may be achieved.

Our MD4 stepping motor driver can be set to divide a full step such that each step pulse received will cause a rotation of:

1. one-tenth of a full step
2. one-fifth of a full step
3. one-half of a full step
4. one full step

With the R command, we can program a stepping rate range of 32 to 4000 full steps/second. This corresponds to a step pulse frequency of 320 Hz to 40,000 Hz. The following table shows the resultant full step rate for each possible MD4 setting:

MD4 Setting	Resultant Step Rate Range (full steps/second)
1/10	32 - 4000
1/5	64 - 8000
1/2	160 - 20,000
1	320 - 40,000

For the UE30 motor type, the above discussion is valid except that the MD4 will not ministep the motor. Instead, for every 10 step pulses it receives, it moves the motor one full step.

10.2 QUANTITY

The number of step pulses outputted is 10 times the programmed number. If you program 1000 steps for the W axis by issuing the NW1000 command, the MC4 will output 10,000 step pulses when you initiate the move.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

The parameter for the N command specifies the number of full steps and the fraction of a full step desired for the next move. The parameter for the P command specifies the desired final position in terms of the number of full steps and the fraction of a full step.

Since a stepping motor will move only one-tenth of a full step for each step pulse from the MC4, it is necessary for the MC4 to supply a quantity of step pulses that is 10 times the programmed number in order to reach the desired position.

For the UE30 motor type, the above discussion is valid except that the MD4 will not ministep the motor. Instead, for every 10 step pulses it receives, it moves the motor one full step.

10.3 DIRECTION

The step pulse direction controls the rotational direction of the motor. The MC4 has two channels for pulse outputs for each axis. One channel supplies positive direction step pulses which create a counterclockwise rotation of the motor. The other channel supplies negative direction step pulses which create clockwise rotation of the motor.

10.4 TIME

The MC4 has several timing options to control when it will begin to output step pulses. These are:

1. when it receives a real-time move command such as the M command and the P command and when it encounters these same commands in a stored program
2. when another axis has completed a pre-defined number of steps
3. when a pre-programmed time interval has expired
4. when an external event triggers it

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 11: SAMPLE PROGRAMS

In this chapter we present the sample programs that you will use in Chapters 4 and 5 to establish correct communication between your host computer and the MC4.

We show one sample program written for three different IEEE-488 interfaces and an RS-232-C program for the IBM-PC and compatibles.

The three IEEE-488 interfaces are:

1. Real Time Devices for the IBM-PC
2. National Instruments for the IBM-PC
3. HP-85

The RS-232-C program has two versions. One is for the hardware handshake mode and the other is for the software handshake mode. Included also is a C program for National Instruments IEEE and one for RS-232C.

11.1 IEEE-488 SAMPLE PROGRAMS

11.1.1 REAL TIME DEVICES

```
1 ' DIRECTIONS:  1. Set MC4 address to 6
2 '              2. Set MC4 command terminator to Carriage Return/Line Feed
3 '*****************************************************************************
4 ' Lines 10-190 must be included in front of your application
5 ' program.  These lines are copied from the GPINIT.BAS program
6 ' supplied on the distribution diskette by Real Time Devices,
7 ' (C) COPYRIGHT 1985
8 '*****************************************************************************
9 '
10  DEFINT A-Z
20  DEF SEG = 0 : LIBSEG = VAL("&H" + HEX$(PEEK(1018) + PEEK(1019) * 256))
30  FOR LIB = 1 TO 1
40     DEF SEG = LIBSEG
50     ID = PEEK(262) + PEEK(263) * 256 : LENGTH = PEEK(ID)
60     ID$ = ""
70     FOR I = 1 TO LENGTH
80     ID$ = ID$ + CHR$(PEEK(ID + I))
90     NEXT I
100    IF ID$ = "GP100" THEN GP100 = LIBSEG
110    LIBSEG = VAL("&H" + HEX$(PEEK(254) + PEEK(255) * 256))
120 NEXT LIB
130 IF GP100 = 0 THEN PRINT "Missing GP100 Subroutine Library" : END
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
140 ' Initialize offsets
150 INIT1 = 256
160 CALL INIT1(INIT2,INIT3,INIT4,INIT5,VERIFY,VERSION,GPRESET,GPNEW,
              GPTIMER,PULSE,HOLD,RTL,RTLHOLD,TONLY,LONLY,EVENT,TRAP,
                ADDRESS,BUS,MESSAGE,TEXT)
170 CALL INIT2(WBYTE,RBYTE,WWORD,RWORD,WDWORD,RDWORD,WQWORD,RQWORD,WSTR,
              RSTR,WARRAY,RARRAY,BWRITE,BREAD,WDMA,RDMA,DMA,TERM,NOTERM,
                CRLF,EOI,PARSER)
180 CALL INIT3(CONTROL,IFC,REN,RENCLR,RENLOC,REMOTE,TCS,TCA,STANDBY,LLO,
              DCL,PPU,SPE,SPD,GTL,LOCAL,SDC,GPCLEAR,GPGET,TRIGGER)
190 CALL INIT4(UNLISTEN,UNTALK,MYLISTEN,MYTALK,LISTEN,XLISTEN,ALISTEN,TALK,
              XTALK,BUSCOM,PASS,SPOLL,APOLL,REQUEST,PPOLL,CONFIGURE,
              RESPONSE,MYSTATUS)
230 '
240 '*************************************************************************
250 '
260 CALL GPRESET(STATUS)
270 CALL REN(STATUS)
280 CALL IFC(STATUS)
290 COUNT=50
300 CALL GPTIMER(COUNT)
310 CALL NOTERM
320 TR$=CHR$(10):SELECT=-1:CALL PARSER(SELECT,TR$)
330 CALL UNTALK(STATUS)
340 CALL UNLISTEN(STATUS)
350 LISSEN = 6
360 TALKER = 6
370 '
380 '*************************************************************************
390 '
400 'NOTE: The MC4 command (CMMD$) can be obtained directly
410 '      from the keyboard or read from a data file
420 '
430 INPUT CMMD$                              'or read data file
435 IF CMMD$="Q" THEN GOSUB 590:GOTO 430     'read MC4 status on the fly
440 CMMD$=CMMD$+CHR$(13)+CHR$(10)
450 CALL UNLISTEN(STATUS)
460 CALL LISTEN(LISSEN,STATUS):CALL MYTALK(STATUS)
470 CALL WSTR(CMMD$,BYTES,STATUS)
480 D$=LEFT$(CMMD$,1)
490 IF D$="D" THEN GOSUB 590
500 IF D$="L" THEN GOSUB 750
510 IF D$="?" THEN GOSUB 590
520 GOTO 430
530 '
540 '*************************************************************************
550 '
560 ' This routine reads the MC4 output in response
570 ' to a DISPLAY and STATUS command
580 '
590 CALL UNLISTEN(STATUS):CALL UNTALK(STATUS)
600 CALL MYLISTEN(STATUS)
620 CALL TALK(TALKER,STATUS)
630 Q$=STRING$(100," ")
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
640 CALL RSTR(Q$,BYTES,STATUS)
650 CALL UNTALK(STATUS)
660 P$=LEFT$(Q$,BYTES)
670 GOSUB 900
680 RETURN
690 '
700 '*************************************************************************
710 '
720 ' This routine reads the MC4 output
730 ' in response to a LIST command
740 '
750 FOR I=1 TO 100:NEXT I          'Delay for MC4 to prepare listing
760 CALL UNLISTEN(STATUS)
770 CALL UNTALK(STATUS)
780 CALL MYLISTEN(STATUS)
790 CALL TALK(TALKER,STATUS)
800 Q$=STRING$(100," ")
810 CALL RSTR(Q$,BYTES,STATUS)
820 CALL UNTALK(STATUS)
830 P$=LEFT$(Q$,BYTES)
840 PRINT P$;
850 IF P$<>"END"+CHR$(13) THEN 760
860 RETURN
900 NB=1
905 LSTRG=LEN(P$)
910 IF NB>LSTRG-1 THEN 990
915 A$=MID$(P$,NB,2)
920 HEAD$=LEFT$(A$,1)
925 IF HEAD$="0" THEN GOSUB 1000:NB=NB+2:GOTO 910
930 IF HEAD$="1" THEN GOSUB 1100:NB=NB+2:GOTO 910
940 IF HEAD$="2" THEN GOSUB 1200:NB=NB+2:GOTO 910
950 IF HEAD$="3" THEN GOSUB 1300:NB=NB+2:GOTO 910
960 IF HEAD$="4" THEN GOSUB 1400:NB=NB+2:GOTO 910
965 IF HEAD$="W" OR HEAD$="X" OR HEAD$="Y" OR HEAD$="Z" OR HEAD$="E" THEN
    GOSUB 1500:GOTO 990
970 PRINT"":PRINT"STATUS BYTE INVALID"
990 RETURN
1000 PRINT"":PRINT"STATUS BYTE --> MOVING/STATIONARY":PRINT""
1005 STAT$=RIGHT$(A$,1)
1007 S=ASC(STAT$)
1010 B=S AND &HF          'clear upper nibble -- save status
1020 C=B AND 1:IF C=1 THEN PRINT"W-AXIS IS MOVING" ELSE PRINT"W-AXIS IS
     STATIONARY"
1030 C=B AND 2:IF C=2 THEN PRINT"X-AXIS IS MOVING" ELSE PRINT"X-AXIS IS
     STATIONARY"
1040 C=B AND 4:IF C=4 THEN PRINT"Y-AXIS IS MOVING" ELSE PRINT"Y-AXIS IS
     STATIONARY"
1050 C=B AND 8:IF C=8 THEN PRINT"Z-AXIS IS MOVING" ELSE PRINT"Z-AXIS IS
     STATIONARY"
1060 RETURN
1070 '
1075 '*************************************************************************
1080 '
1100 PRINT"":PRINT"STATUS BYTE --> TRAVEL LIMIT WX":PRINT""
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
1105 STAT$=RIGHT$(A$,1)
1107 S=ASC(STAT$)
1110 B=S AND &HF          'clear upper nibble -- save status
1120 C=B AND 1:IF C=1 THEN PRINT"+W LIMIT TRUE" ELSE PRINT "+W LIMIT FALSE"
1130 C=B AND 2:IF C=2 THEN PRINT"-W LIMIT TRUE" ELSE PRINT "-W LIMIT FALSE"
1140 C=B AND 4:IF C=4 THEN PRINT"+X LIMIT TRUE" ELSE PRINT "+X LIMIT FALSE"
1150 C=B AND 8:IF C=8 THEN PRINT"-X LIMIT TRUE" ELSE PRINT "-X LIMIT FALSE"
1160 RETURN
1170 '
1175 '*************************************************************************
1180 '
1200 PRINT"":PRINT"STATUS BYTE --> TRAVEL LIMIT YZ":PRINT""
1205 STAT$=RIGHT$(A$,1)
1207 S=ASC(STAT$)
1210 B=S AND &HF          'clear upper nibble -- save status
1220 C=B AND 1:IF C=1 THEN PRINT"+Y LIMIT TRUE" ELSE PRINT "+Y LIMIT FALSE"
1230 C=B AND 2:IF C=2 THEN PRINT"-Y LIMIT TRUE" ELSE PRINT "-Y LIMIT FALSE"
1240 C=B AND 4:IF C=4 THEN PRINT"+Z LIMIT TRUE" ELSE PRINT "+Z LIMIT FALSE"
1250 C=B AND 8:IF C=8 THEN PRINT"-Z LIMIT TRUE" ELSE PRINT "-Z LIMIT FALSE"
1260 RETURN
1270 '
1275 '*************************************************************************
1280 '
1300 PRINT"":PRINT"STATUS BYTE --> 3 BIT INPUT":PRINT""
1305 STAT$=RIGHT$(A$,1)
1307 S=ASC(STAT$)
1310 B=S AND &HF          'clear upper nibble -- save status
1320 C=B AND 1:IF C=1 THEN PRINT"BIT 1 HIGH" ELSE PRINT"BIT 1 LOW"
1330 C=B AND 2:IF C=2 THEN PRINT"BIT 2 HIGH" ELSE PRINT"BIT 2 LOW"
1340 C=B AND 4:IF C=4 THEN PRINT"BIT 3 HIGH" ELSE PRINT"BIT 3 LOW"
1350 PRINT"":RETURN
1370 '
1375 '*************************************************************************
1380 '
1400 PRINT"":PRINT"STATUS BYTE --> WAIT UNTIL":PRINT""
1405 STAT$=RIGHT$(A$,1)
1407 S=ASC(STAT$)
1410 B=S AND &HF          'clear upper nibble -- save status
1420 C=B AND 1:IF C=1 THEN PRINT"WAITING FOR NUMBER OF STEPS"
1430 C=B AND 2:IF C=2 THEN PRINT"WAITING FOR EXTERNAL EDGE TRANSITION"
1440 C=B AND 4:IF C=4 THEN PRINT"WAITING FOR TIME OUT"
1450 RETURN
1460 '
1470 '*************************************************************************
1480 '
1500 PRINT"":PRINT"STATUS BYTE --> DISPLAY READOUT":PRINT""
1510 DISP$=MID$(P$,NB,(LSTRG-NB))
1520 PRINT DISP$
1530 PRINT"":RETURN
1540 C=B AND 4:IF C=4 THEN PRINT"WAITING FOR TIME OUT"
1550 RETURN
1560 '
1570 '*************************************************************************
1580 '
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.1.2 NATIONAL INSTRUMENTS

```
10 '
20 ' DIRECTIONS: 1. Set MC4 address to 6
30 '              2. Set MC4 command terminator to CARRIAGE RETURN/LINE FEED
40 '
50 '*************************************************************************
60 '
70 ' Lines 130 to 180 must be included in front of your application
80 ' program.  These lines are copied from the DECL.BAS program
90 ' supplied on the distribution diskette by National Instruments,
95 ' (C) COPYRIGHT 1984,1985
100 '
110 '*************************************************************************
120 '
130       CLEAR   ,59000!          ' BASIC Declarations
140       IBINIT1 = 59000!
150       IBINIT2 = IBINIT1 + 3
160       BLOAD "bib.m",IBINIT1
170       CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,
                       IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,
                       IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF)
180       CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,IBCMDA,
                       IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBRDI,
                       IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
190 '*************************************************************************
200 '
210 ' This is the beginning of the application program
220 '
230 '*************************************************************************
240 '
250 BDNAME$ = "MC4"
260 CALL IBFIND (BDNAME$,MC4%)
270 '
280 ' The label -- MC4 -- is assigned to address 6 in the
290 ' configuration program: -- IBCONF --
300 '
310 '*************************************************************************
320 '
330 ' Set and enable read string terminator
340 ' Note: This can also be done in config. program  -- IBCONF --
350 '
360 EOSV%=&HA                   'Input string terminator is line feed
370 V%=EOSV%+&H400              '0400 means to terminate when receive EOS
380 CALL IBEOS (MC4%,V%)
390 '
392 TERM$=CHR$(13)+CHR$(10)     'MC4 command terminator is CR/LF
394 '
400 '*************************************************************************
410 '
420 ' This is the beginning of the listen routine
430 '
440 INPUT CMMD$                              'or read data file
450 IF CMMD$="Q" THEN GOSUB 540:GOTO 440     'read status on the fly
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
460 CMMD$=CMMD$+TERM$
470 CALL IBWRT (MC4%,CMMD$)
480 D$=LEFT$(CMMD$,1)
490 IF D$="D" THEN GOSUB 540
500 IF D$="?" THEN GOSUB 540
510 IF D$="L" THEN GOSUB 600
520 GOTO 440
530 '
538 ' This is the beginning of the STATUS REQUEST routine
539 '
540 P$=SPACE$(255)              'Read until receive 255 bytes or terminator
550 CALL IBRD (MC4%,P$)
560 P$=LEFT$(P$,IBCNT%-2)
570 GOSUB 900
580 RETURN
600 ' This is the beginning of the LIST routine
610 '
615 FOR I=1 TO 100:NEXT I     'delay for MC4 to prepare listing
620 LST$=SPACE$(255)          'Read until receive 255 bytes or terminator
630 CALL IBRD (MC4%,LST$)
640 LST$=LEFT$(LST$,IBCNT%-2)
650 PRINT LST$
660 IF LST$<>"END" THEN 620
670 RETURN
900 ' At this location, insert statements 900-1580 from the sample IEEE
910 ' program for the Real Time Devices IEEE-488 interface.
```

11.1.3 HP-85

```
10 !
20 !  DIRECTIONS:
30 !
40 !      1. Set MC4 address to 6
50 !
60 !      2. Set MC4 command terminator to carriage return/line feed
70 !
80 !
90 CLEAR 7
100 RESET 7
110 DIM CMMD$[100]
115 DIM P$[100]
120 DIM LST$[100]
130 !
140 !   Obtain the MC4 command from the keyboard by using the
150 !   " INPUT CMMD$ " command ( or read a data file )
160 !
170 INPUT CMMD$
180 IF CMMD$="Q" THEN GOSUB 300: GOTO 170
190 OUTPUT 706 USING "K";CMMD$
200 A$=CMMD$[1,1]
210 IF A$="?" THEN GOSUB 300
220 IF A$="D" THEN GOSUB 300
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
220 IF A$="L" THEN GOSUB 400
230 GOTO 170
240 !
250 !  Receive the MC4 response to a display command and a status
260 !  request command
270 !
300 ENTER 706; P$
310 SEND 7; UNT
320 GOSUB 900
330 GOTO 170
340 !
400 !  Receive the MC4 response to a list command
410 !
420 ENTER 706; LST$
430 SEND 7; UNT
440 DISP LST$
450 IF LST$<>"END" THEN 420      !loop until receive "END"
460 RETURN
470 !
900 ! At this location insert HP-85 equivalent statements of lines
910 ! 900 - 1580 of the Real Time Devices IEEE-488 sample program
```

11.2 RS-232-C SAMPLE PROGRAMS

11.2.1 IBM-PC : HARDWARE HANDSHAKE

```
10 '
20 'DIRECTIONS: 1. SET THE FOLLOWING RS232 PARAMETERS WITH THE MC4 SWITCHES
30 '                - BAUD RATE: 9600
40 '                - EVEN PARITY
50 '                - 7 DATA BITS
60 '                - 2 STOP BITS
70 '
80 '             2. SET COMMAND TERMINATOR TO CARRIAGE RETURN
85 '
90 '             3. RUN PROGRAM
95 '
100 '*************************************************************************
110 '
120 OPEN "COM1:9600,N,8,1,CS65535,DS0" AS #1
130 '
140 ' NOTE:  IF A PARTICULAR COMMAND EXECUTION TIME IS LONGER
150 '        THAN 65.535 SECONDS, A "DEVICE TIMEOUT" WILL OCCUR
160 '        BECAUSE THE CLEAR TO SEND SIGNAL WILL STILL BE FALSE
170 '
180 '        IF THIS IS THE CASE, USE A SOFTWARE HANDSHAKE
190 '
200 '*************************************************************************
210 '
220 KOM=0:CTSEND=0
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
230 '
240 '*************************************************************************
250 '
260 INPUT CMMD$
270 '
280 ' Enter " ? " when you wish a status report.  You can issue the " ? "
290 ' command while the MC4 is idle and while it is executing a previous
300 ' command.
310 '
320 IF CMMD$="?" THEN GOSUB 620:GOTO 260
330 '
340 ' Enter " Q " if you hear a beep from the MC4 to read the error message
350 '
360 IF CMMD$="Q" THEN GOSUB 440:GOTO 260
370 IF KOM=1 THEN GOSUB 810
380 PRINT#1,CMMD$          'SEND COMMAND TO MC4
390 IF LEFT$(CMMD$,1)="D" OR LEFT$(CMMD$,1)="d" THEN GOSUB 440
400 IF LEFT$(CMMD$,1)="L" OR LEFT$(CMMD$,1)="l" THEN GOSUB 900
410 GOTO 260
420 '
430 '*************************************************************************
440 P$=""
450 WHILE LOC(1)<>0
460 INPUT#1,P$
470 GOSUB 1000              'go decipher received string
480 WEND
490 IF P$="" THEN 450
500 RETURN
510 '
520 '*************************************************************************
530 '
540 ' This routine is used to send the status request command.  We must
550 ' reconfigure the communication port so that we are able to send the
560 ' status request command even if the MC4 is outputting "CLEAR TO SEND"
570 ' false ( which is the basis of the hardware handshake ).
580 '
590 ' Note that the first status request will take some time because
600 ' openning the communication port is slow.
610 '
620 IF CTSEND=1 THEN 660
630 CLOSE 1
640 OPEN "COM1:9600,N,8,1,CS0,DS0" AS #1
650 KOM=1:CTSEND=1
660 PRINT#1,CMMD$
670 GOSUB 440
680 RETURN
690 '
700 '*************************************************************************
710 '
720 ' This routine reconfigures the communication port so that we cannot
730 ' send commands to the MC4 if its is outputting " CLEAR TO SEND "
740 ' false.  This routine is executed the first time we wish to send a
750 ' a command to the MC4 after we have previously configured the port to
760 ' ignore the " CLEAR TO SEND " signal.
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.2.3 IEEE SEND AND RECEIVE

```
extern int ibsta,iberr,ibcnt ;
#define BELL            0x07
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "decl.h"
int controller_mc4 ;


void clean_buffer (char *buffer,int size)
{
   int i=0 ;
   while (i++<size)
     *(buffer++)='\0';
}


void error (void)
{
   printf ("Device error");
   printf ("ibsta=0x%x, iberr=0x%x ,",ibsta,iberr);
   printf ("ibcnt=0x%x\n",ibcnt);
}


int init_device (void)
/*Get address number, set address, set EOS to CR,clear device.*/
{
  char num[4],device[7];
  int i,v,GPIB0;
  while (1)
    {
    printf ("\n\nEnter GPIB target device address [6]-");
    gets (num);
    if (strlen (num)==0) strcpy (num,"6");/default is address 6/
    i=atoi (num);           /*address between 1 & 16 */
    if ((i>=1) && (i<=16)) break;
    else putchar (BELL);
    }

    sprintf (device,"DEV%s",num);
    if ((GPIB0 = ibfind ("GPIB0"))<0)
        {
        error ();
        return (0);
        }
    if ((controller_mc4=ibfind (device))<0)
        {
        error ();
        return(0);
        }
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
    v=REOS|'\n';              /*finish character reading with LF */
    ibeos (controller_mc4,v);
    ibsic (controller_mc4);
    ibtmo (controller_mc4,0);
    printf ("Type READ or hit Enter to read status\n\n\n");
    return(1);
}


void write_device (int device,char *cmde)
/* terminator for MC4 is carriage return, set S3.4 UP */
{
   char command[70];
   strcpy (command,cmde);
   strcat (command,"\r");      /*If S3.4 is down then this line should read/
   /strcat (command,"\r\n");
   ibwrt (device,command,strlen(command));
   if (ibsta & ERR) error ();
}
/


void read_device (int source,char *fetch)
{
   clean_buffer (fetch,71);
   ibrd (source,fetch,70);
   if (ibsta & ERR) error ();
   printf ("  %s\r",fetch);
}

void main (void)
{
   int done=0;
   char fetch[80], command[80];
   if (!init_device ()) exit(-1);
   while (!done)
     {
     printf ("Enter command - ");
     gets (command) ;
     if (!strcmpi(command,"Q")||!strcmpi(command,"QUIT")) done=1;
     else if (!strcmpi(command,"")||!strcmpi(command,"READ"))
        read_device (controller_mc4,fetch);
     else write_device (controller_mc4,command);
     }
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

11.2.4 MC4 LISTEN WITH RS-232C

```
#include <stdio.h>
#include <bios.h>
#include <dos.h>

#define FALSE 0
#define TRUE  !FALSE
#define F1  0x3B
#define COM1  0x3f8          /* 0x2f8 for COM2 */
#define LSR   5
#define THR   0

int xmit (char ch)
{
   /*make sure that the transmitter holding register is ready for */
   /*a character before you write one to it.                      */

   register int cnt =0;
   while (!(inp (COM1+LSR) & 0x20) && cnt <10000) cnt++;
   if (cnt>=10000) return (-1);
   outp (COM1+THR,ch);
   return (0);
}

int keybd (void)
{
   char c;
   if ((c=getche())>=0)
      {
      if (c==0)
         {
         if ((c=getch())==F1) return (FALSE);
         return (TRUE);
         }
      xmit (c);
      if (c==13) printf("\n");
      }
    return (TRUE);
}


void main (void)
{
   unsigned com_data;
   com_data = _COM_CHR8|_COM_STOP1|_COM_NOPARITY|_COM_9600;
   printf ("\n\nCOM1, 8 bits, 1 stop bit, no parity, 9600 bauds\n\n");
   _bios_serialcom (_COM_INIT,0,com_data);

   while (TRUE)
      if (!keybd())
         exit (0);
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

CHAPTER 12: ILL-CONDITION STATUS MESSAGES: PROBLEMS AND SOLUTIONS

The MC4 is designed to detect user and interfacing errors and report back a message that will help the user quickly determine the nature of the error. The two major categories of errors are operational errors and communication errors. Operational errors are reported over the communication channel. We assume that the communication channel is functioning, but we have incorrectly programmed the MC4 or its status does not allows us to do what we want at this time. The format of the error message is the following:

Exx ERROR MESSAGE

E is the header byte indicating that an error message follows

xx is the code number for the particular error

ERROR MESSAGE is a description of the type of error

Interfacing errors are reported on the MC4 front panel. If we have a communication problem, we cannot use the communication channel to receive an error message. The error is reported by a flashing number indicator in the MC4 program select window.

12.1 OPERATIONAL PROBLEMS

E01 COMMAND ERROR

First character in command string is not in the MC4 command set

E02 PARAMETER ERROR

- Second character is not a valid axis designator, or
- numerical parameter is out of allowed range

E03 PARAMETER/MOTOR ERROR

Command specifies fraction of a full-step for an axis programmed to drive a UE30 stepping motor. Fractions of a full step can only be programmed for UE7x stepping motors. Check which motor type the axis is set for by using the RESET switch of that axis and the PROGRAM SELECT switch. Refer to section 1.4.

E04 Not Used

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

E05 BATTERY RAM PROTECTED

The write protect switch is active. Its position will not allow writing to the battery back-up RAM.

E06 OPERATOR BUFFER FULL

Command string too long; maximum allowed length is 255 bytes.

E07 PROGRAM BUFFER FULL

Battery back-up RAM is full; capacity is 8192 bytes of compactly coded commands -- not 8192 bytes of ASCII coded commands.

E08 VERIFICATION ERROR!!!

Stored program contains an error. This can be a command error or a parameter error. The error will be a result of a some mishap to the battery back-up circuitry. You will have to re-enter your program.

E09 Not Used

E0A Not Used

E0B PARITY ERROR

Parity error exists for current command. Check the MC4 word characteristic switches and your program's setting for desired parity.

E0C CHARACTER LOST

Current command overran the previous command. This error is a result of an improper handshake between the MC4 and your host computer. You should not send a command as long as the MC4 indicates it is busy. For the hardware handshake, the MC4 will indicate busy by holding the CLEAR TO SEND line false (at -12V). For the software handshake, the MC4 is busy as long as it has not sent an XON after it has sent an XOFF.

E0D FRAME ERROR

Error exists in the start and stops bits of the character sent. Check the MC4 word characteristic switches and your program's setting for the number of stop bits.

E34 ACTIVE LIMIT: +W

Command specified a W axis move in the positive direction and the W axis is at positive direction travel limit or travel limit reached during move.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

E35 ACTIVE LIMIT: +X

E36 ACTIVE LIMIT: +Y

E37 ACTIVE LIMIT: +Z

E38 ACTIVE LIMIT: -W

E39 ACTIVE LIMIT: -X

E3A ACTIVE LIMIT: -Y

E3B ACTIVE LIMIT: -Z

12.2 IEEE-488 AND RS-232-C COMMUNICATION PROBLEMS

The MC4 is designed to detect some communication errors. If the implementation is incorrect, the MC4 will time-out and report what it was waiting for and never received. The MC4 will indicate the particular error by flashing a number in the program select window and beeping 3 times. The possible numbers that the MC4 may flash and their corresponding meaning is given below.

Error numbers 1 and 2 pertain only to the IEEE-488 listen function.

Error numbers 3 and 4 pertain only to the IEEE-488 talk function.

Error numbers 5 and 6 pertain only to the RS-232-C.

Flashing #:	Error Indicated:
1	a. incorrect terminator programmed on MC4 dip switch b. host computer did not send terminator c. data bits garbled
2	MC4 internal chip not initialized properly; power down and power up MC4
3	MC4 did not recognize the NDAC line transition from low to high coming from your host computer. The transition is not recognized because there was none or the opposite transition from high to low came too quickly, i.e., the NDAC pulse from your computer is too narrow. You will receive only the first character of the MC4 response string.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4	MC4 did not recognize your computer asserting not ready for data true.
5	a. incorrect terminator programmed on MC4 dip switch b. host computer did not send terminator c. data bits garbled
6	host computer indicating it is not ready to receive data from the MC4 Hardware Handshake: • pin 20 of RS-232-C connector at -12V. It must be at +12V to indicate ready. Software Handshake: • pin 20 of RS-232-C connector at -12V. It must be at +12V to indicate ready. - OR - • host computer has previously sent XOFF and has not sent XON

12.3 OTHER PROBLEMS AND POSSIBLE CAUSES

12.3.1 IEEE-488

Problem: No MC4 response to a command or request for status

Possible Cause: MC4 address incorrect

12.3.2 RS-232-C

Problem: No MC4 response to a command or request for status

Possible Cause:
a. connecting cable incorrect; receive data and transmit data pins reversed; ground connection missing
b. host computer sending data to wrong communication port
c. host computer not sending data

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

12.3.3 OPERATIONAL

Problem: MC4 executes P command improperly; final position is not the one programmed with the P command

Possible Cause: MC4 not properly programmed for the encoder type driving the display counter in error

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com



Certificate No.: FM 27207

Newport Corporation, Irvine, California, has been certified compliant with ISO 9002 by the British Standards Institution.

Newport®

Corporate Headquarters
Newport Corporation
1791 Deere Avenue
Irvine, CA 92714
Telephone: 714-863-3144
Facsimile: 714-253-1800

Belgium
Telephone: 016-402927
Facsimile: 016-402227

Canada
Telephone: 905-567-0390
Facsimile: 905-567-0392

France
Telephone: 1-60 91 68 68
Facsimile: 1-60 91 68 69

Germany
Telephone: 06151-36 21-0
Facsimile: 06151-36 21-50

Italy
Telephone: 02-924-5518
Facsimile: 02-923-2448

Japan
Telephone: 03-5379-0261
Facsimile: 03-5379-0155

Netherlands
Telephone: 03405-92111
Facsimile: 03405-70242

Switzerland
Telephone: 01-740-2283
Facsimile: 01-740-2503

Taiwan R.O.C.
Telephone: 886-2-506 2366
Facsimile: 886-2-507 9268

United Kingdom
Telephone: 01635-521757
Facsimile: 01635-521348

Printed in the USA on recycled paper

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP