

## **This is a Discontinued Product**

---

Contact Kollmorgen Customer Support at  
1-540-633-3545 or email us at [support.kollmorgen.com](mailto:support.kollmorgen.com)  
if assistance is required.

**MX2000 Quick Start**  
**Programming & Set-up**  
**Guide**



## ENGINEERING CHANGES

Superior Electric reserves the right to make engineering refinements on all its products. Such refinements may affect information given in instructions, Therefore, **USE ONLY THE INSTRUCTIONS THAT ARE PROVIDED WITH THE PRODUCT.**

RECORD OF REVISION		
Revision	Date	Description
A	10/10/1997	Preliminary Issue
B	7/10/1998	Revise ISO logo and correct page formatting errors on pages 13, 16, and 22.

## **Table of Contents**

Section I	MX2000 Basic Overview and Set-Up Instructions.....	3-6
Section II	Wiring the MX2000 to a Stepping Motor Drive.....	7-8
Section III	Wiring the MX2000 to a Superior Electric Servo Motor Amplifier.....	9-10
Section IV	Connecting I/O to the MX2000.....	11
Section V	Connection Expansion I/O to the MX2000.....	12
Section VI	Checking System I/O Using Terminal Mode.....	13-14
Section VII	Create a Project (Program).....	15-17
Section VIII	Tuning a Servo with the MX2000.....	18-23
Section IX	Test Motion Program (Stepper) .....	24-25
Section X	Test Motion Program (Servo) .....	26-27
Appendix A	Test Operator Interface Program (IWS120 / 127).....	28-29
Appendix B	Sample Program - Cut To Length Application .....	30-34
Appendix C	The IWS “Include” File.....	35-38
Appendix D	Technical Support options .....	39

# *Welcome to the MX2000 Quick Start Guide*

It is the intention of this guide to aid you in the setting up, wiring, and programming of the MX2000 motion control system. This guide will reference the instruction manual, **400030-059** that was supplied with your MX. It should be noted that this guide is not intended to replace the MX2000 Instruction Manual, but to supplement it and get you started faster. We strongly encourage you to read the MX2000 Instruction Manual and become acquainted with your new control.

## **Prerequisites for Using this Quick Start Guide:**

1. You should know and practice all common methods for wiring a safe control panel and take whatever steps are needed to assure the safety of those around the motion control devices.
2. You should be familiar with the operation of your PC and its operating system. Your PC's operating system should be either Microsoft Windows 3.x or Windows 95. You will need an unused serial port to connect to the MX2000 control.
3. You should have loaded and can functionally operate the MX2000 software. If you have not done so, please load this software into your PC at this time. You may also choose to access the Superior Electric Internet web site (<http://www.superiorelectric.com>) to verify that you are using the latest version of the MX2000 software and firmware. Both the software and the MX's firmware are updated periodically, so a check of our web site might be a good idea.
4. You should have some basic knowledge of the programming language BASIC. MX2000 programs are written in BASIC, so a working knowledge of this language and its structure will prove helpful. We will provide program examples in this guide for you to use, but we are not intending to teach BASIC programming here.
5. This manual assumes that you are using the **"INSTRUCTIONS FOR SUPERIOR ELECTRIC SLO-SYN MODEL MX2000 PROGRAMMABLE MULTI-AXIS MOTION CONTROLLER"** - **REV. E** (see back cover to verify).
6. You should be using Version 3.4 of the MX2000 programming software and firmware.

## **How to Use This Manual:**

This manual has been organized and laid out so that you can start with Section I and proceed through to the last section in a logical order. Following this manual section by section will give you a good head start, and you will be well on your way to a complete and running system.

- ☐ These are action boxes. These symbols indicate steps that must be completed to properly set up the control. Feel free to place a check mark into the box when the step has been completed.
- (↵) This symbol will be used as the symbol for the ENTER key on your PC. Where you see this symbol after a command, you'll know to hit the ENTER key to enter the command.

Now, let's get started.....

## SECTION I

### MX2000 BASIC OVERVIEW AND SET-UP INSTRUCTIONS

The base MX2000 is made up of a power supply, a DSP card and at least one Axis card. Other options include additional Axis cards, Digital I/O cards and Expansion I/O cards. Please have your MX2000 available and we'll begin the set-up process.

#### **POWER SUPPLY CARD**

**DO NOT APPLY POWER. Make NO connection at this time.** When we are ready, input power (90 to 265 VAC single phase) will be connected to this card. This card also contains a bank of Expansion I/O (except on the MX2000-2A and -6A, which has digital I/O). Expansion I/O will be covered in more detail in Section V.

#### **DSP CARD**

**DO NOT APPLY POWER.**

**Remove the DSP card and inspect ALL jumper settings. Locate J10 & J11.**

J10 Configures the Auxiliary serial port for either RS-485 or RS-232 communications. Factory default is RS-485. For use with any Superior Electric Operator Interface, we suggest that this be set to RS-232.

J11 Configures the DSP card inputs (i.e.: CLR & SEL inputs) for either SINK or SOURCE mode. Factory default is SINK mode.

J4 Ignore this jumper.

**Replace the DSP Card back into the MX2000.**

**Select Indexer ID number.** On the front panel of the DSP card, select the ID # for this particular MX. This ID # is used when several MX2000 controls are to be used in one system, and they are to be linked by a common serial connection back to a host PC. Typically only one MX is used in a system, and therefore the factory default is usually acceptable. Factory default is 1.

**Select RS-232 or RS-485 communications for the HOST serial port.** Locate the bank of 4 DIP type switches on the front panel of the DSP card and select either RS-232 or RS-485. Typically, the serial port on your PC will be connected to this port, and therefore selecting RS-232 would be the correct choice.

## SECTION I - continued.

❑ **Select the baud rate.** Using the same DIP switches as before, you can select the baud rate at which your computer will communicate with the MX. The most commonly used baud rate is 9600, but 38,400 baud can be used if your PC has 16550 UARTs. (Windows 3.x users can determine the type of UARTs that your PC has by running the Microsoft MS-DOS program MSD.EXE.)

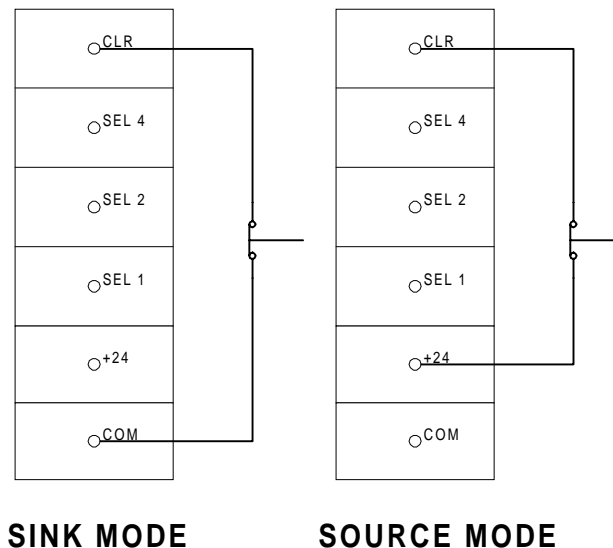
### ❑ **Connect the Program Clear Input (CLR)**

The CLR input MUST be connected to either +24Vdc or to COM (depending upon the J11 setting previously discussed) in order:

- 1) for a program to load into RAM and run upon power being applied
- 2) for any type of motion to occur
- 3) for a program to run from commands in terminal mode
- 4) to tune a servo

This connection can be a jumper or, for safety reasons, it can also be a set of normally closed contacts from a push button switch or a PLC.

If in sink mode (as determined by the J11 setting), wire a set of normally closed contacts from the CLR input to common. (If source mode is being used, connect CLR to +24vdc.)



**WARNING:** The closure of this circuit will allow a program to load upon power up and therefore could possibly cause MOTION to begin unexpectedly. DO NOT connect this input until all safety considerations have been exercised.

**SECTION I - continued.**

**DUAL - AXIS INTERFACE CARD(S)**

**DO NOT APPLY POWER.**

**Remove and inspect all Dual-Axis Interface Cards**

**Select the Address Configuration.** The card address determines the axis numbers for that particular card.

**It is required that at least one axis card be used in an MX system and that it contain the address setting 01.** While looking at the etched chart on the axis board itself, verify and set the position of the switches labeled ABC as needed by your system. Factory default for the ABC switches is for address #1. IGNORE switch D.

Address 1 yields Axis Numbers 1 (A) and 2 (B).

Address 2 yields Axis Numbers 3 (A) and 4 (B).

Address 3 yields Axis Numbers 5 (A) and 6 (B).

Address 4 yields Axis Numbers 7 (A) and 8 (B).

**Locate the J7 jumper on the axis card.** Select either SINK or SOURCE I/O. Factory default is sink. Write down your selection for later reference.

**Repeat this process for ALL Dual-Axis Interface Cards to be used.**

**Replace all Dual-Axis Interface Cards back into the MX.**



**SECRETS AND HINTS**

**Dual Axis Interface Cards:**

**Refer to Appendix A - page A-2 through A-4 of the MX2000 manual, 400030-059, for more details.**



## SECTION I - continued.

### **DIGITAL I/O CARD(S)**

**DO NOT APPLY POWER.**

- Remove and inspect all Digital I/O cards.**
- Locate jumpers J7.** These jumpers select either sinking or sourcing I/O for ALL I/O points on the card. Both jumpers must be positioned for the same mode. Factory default is SINK mode. Select the desired mode and write down your setting for future reference.
- Locate DIP switch S2.** Select the board ID# in exactly the same fashion as with the Dual Axis Interface Cards previously discussed.
- Replace all Digital I/O Cards back into the MX.**

### **EXPANSION I/O CARD(S)**

**DO NOT APPLY POWER.**

- Remove and inspect all Expansion I/O cards.**
- Locate DIP switch S1.** Select the board ID #, as you have done previously, using the A & B switches on S1.
- Replace all Expansion I/O cards back into the MX.**



#### **SECRETS AND HINTS**

For the MX2000-2 and MX2000-6, the expansion port on the power supply card *is* card address #1; therefore all extra Expansion I/O cards should start with Address #2. On the MX2000-6, up to 3 expansions cards can be added so use address #'s 2, 3 & 4. The MX2000-8 has no expansion port on the power supply card; up to 4 expansion cards can be installed, so use address #'s 1, 2, 3 & 4.

**Digital I/O Cards:**

**Refer to Appendix A - page A-5 through A-9 for more details.**

**Expansion I/O Cards:**

**Refer to Appendix A - page A-10 through A-12 for more details.**

## SECTION II

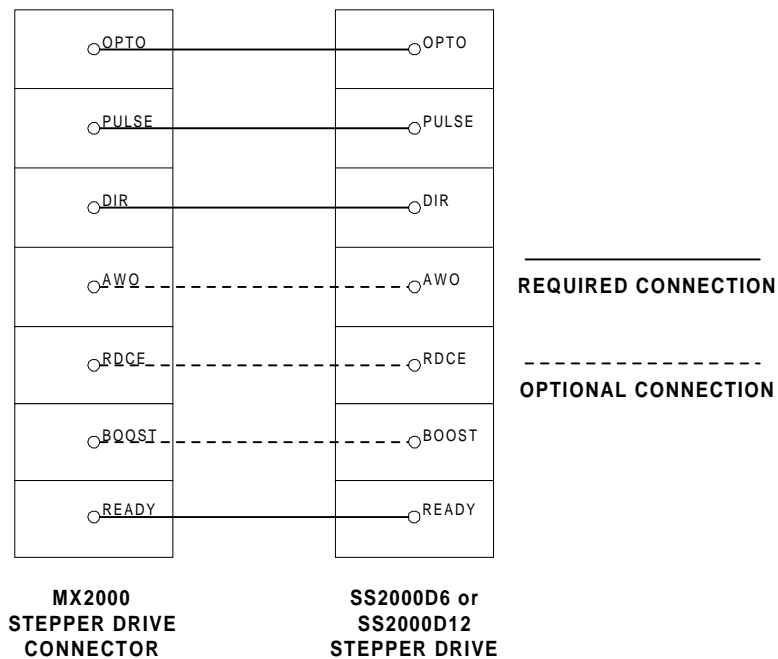
### WIRING THE MX2000 TO A STEPPING MOTOR DRIVE

Depending on the drive, usually only 3 or 4 connections are required for a stepping motor to move. They are:

OPTO  
PULSE  
DIRECTION  
READY (required on SS2000-D6 & D12 drives)

Optional connections are needed only if their functions are to be used. Examples of optional connections are:

AWO (all windings off)  
RDCE (reduce motor current)  
BOOST (boost motor current)





## SECTION III

### WIRING THE MX2000 TO A SUPERIOR ELECTRIC, SERVO MOTOR AMPLIFIER

Make **ALL** the following connections, per axis:

<u>MX2000</u> (Encoder)	<u>Wire Color</u>	<u>/</u>	<u>(old style #)</u>	<u>Amplifier pin #</u> (see Note 1)
<input type="checkbox"/> +5	NC		(NC)	NC
<input type="checkbox"/> GND	White / Green Stripe		(Red #4)	PIN-8 (COM)
<input type="checkbox"/> GND	White / Yellow Stripe		(Blk #4)	PIN-4 (COM)
<input type="checkbox"/> SHLD	Bare Wire		(Bare Wire)	NC
<input type="checkbox"/> A+	Red		(Red #1)	PIN-5 (out A)
<input type="checkbox"/> A-	Black		(Blk #1)	PIN-1 ( <del>out A</del> )
<input type="checkbox"/> B+	Green		(Red #2)	PIN-6 (out B)
<input type="checkbox"/> B-	White		(Blk #2)	PIN-2 ( <del>out B</del> )
<input type="checkbox"/> I+	White / Red Stripe		(Red #3)	PIN-7 (out Z)
<input type="checkbox"/> I-	White / Black Stripe		(Blk #3)	PIN-3 ( <del>out Z</del> )

#### MX2000(Stepper Drive)

<input type="checkbox"/> OPTO	White / Red Stripe		(Red #3)	C1-10 (Drive Up)
<input type="checkbox"/> READY	White / Black Stripe		(Blk #3)	C1-20 (Drive Up)

#### MX2000(Analog Drive)

<input type="checkbox"/> OUT	Green		(Red #2)	C1-11 (Diff Hi)
<input type="checkbox"/> AGND	White		(Blk #2)	C1-01 (Diff Lo)
<input type="checkbox"/> AGND	Bare Wire		(Bare Wire)	C1-04 (Diff Shield)
<input type="checkbox"/> EN+	Red		(Red #1)	C1-02 (Enable)
<input type="checkbox"/> EN-	Black		(Blk #1)	C1-15 (COM)
<input type="checkbox"/> RDY+	NC		(NC)	NC
<input type="checkbox"/> RDY-	NC		(NC)	NC

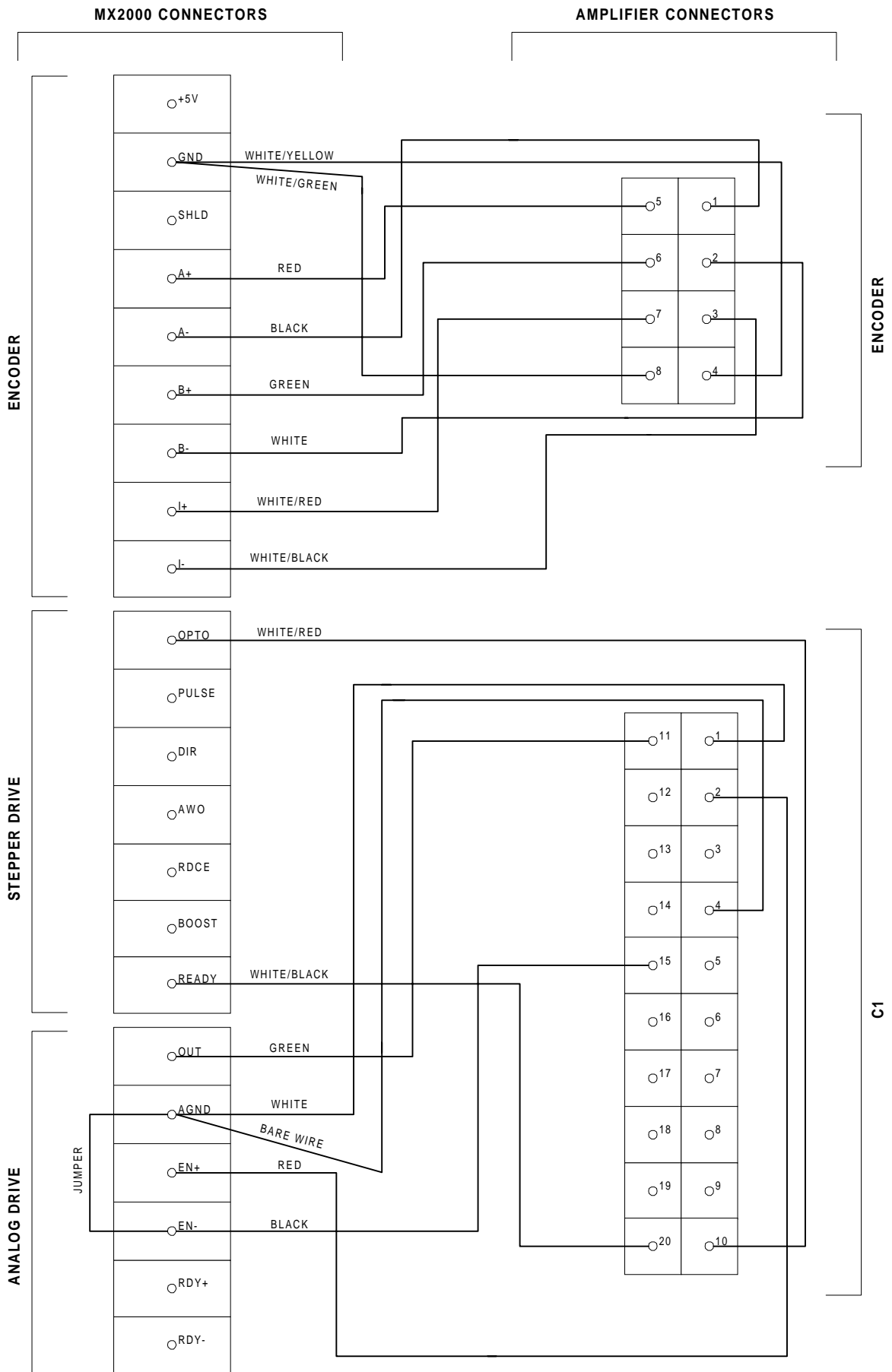
Install a **jumper** from **AGND** to **EN-** on the MX2000 Analog Drive connector.

**All of the above connections are shown in the wiring diagram on the following page.**



#### SECRETS AND HINTS

Note 1 - this connector is NOT labeled on the servo amplifier, but it is referred to as Connector 32 in the Servo Drive Instruction Manual, **400030-064**, on page C-2.



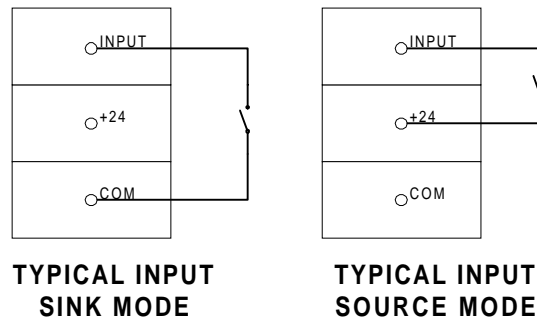
## SECTION IV

### CONNECTING I/O TO THE MX2000

**Note:** Remember the I/O convention selected, via an on-board jumper, for the Dual-Axis card(s) and for the Digital I/O card(s)?. You will need that information here to make proper I/O connections to the MX.

If SOURCE mode was selected; the I/O point will become active when the desired I/O terminal is electrically connected to + 24 Vdc.

If SINK mode is selected, the I/O point will become active when the desired I/O terminal is connected to COMMON.



❑ **CONNECT** all I/O devices (switches, sensors, etc.).

#### INPUT DEFAULTS:

**Event1**(axis) is used as the mechanical homing input.

**Event2**(axis) is used for the mark registration input.

**LIM+** is used as the + direction OVER TRAVEL input.

**LIM -** is used as the - direction OVER TRAVEL input.



#### SECRETS AND HINTS

Some axis card I/O points are not printed on the A side of the card. **Except where it is printed otherwise**, Side A I/O is symmetrical and identical with Side B I/O.

Extra programmable inputs can be obtained by changing the factory defaulted purpose for EVENT1, EVENT2, LIM+, and LIM-. Within your program, you can disable these factory set functions with these command(s):

EVENT1(axis)=0

EVENT2(axis)=0

HARDLIM(axis)=0

**If analog I/O are to be used - refer to page 5-20 in the MX2000 Manual for proper connection techniques.**

## SECTION V

### CONNECTING EXPANSION I/O TO THE MX2000

Each Expansion I/O port contains 50 gold pins for connecting to industry standard “OPTO 22” style I/O racks. Each one of these racks can provide a maximum of 24 I/O points. These points can be mixed and matched for any number of I/O combinations. Examples:

12 Inputs / 12 Outputs

7 Inputs / 17 Outputs

23 Inputs / 1 Output

#### BCD Connections to the Expansion Port(s)

Each expansion port in the system can also handle up to 4 sets of 7 digit BCD thumbwheel switches. The more BCD switches that are added into the system, the fewer expansion I/O points are available, as shown below:

<u># of BCD Switch Banks</u>	<u># of I/O Points</u>
0	24
1	12
2	8
3	4
4	0

BCD switches and connecting cables are available from Superior Electric. We suggest that you purchase your BCD switches and connecting cables from Superior, as they plug directly into the MX with no additional wiring needed.

- CONNECT all expansion I/O points at this time.**
- CONNECT BCD switches and connecting cables, if any.**



#### SECRETS AND HINTS

**Refer to Manual Section 5.3.2.5 for more details on expansion I/O and using BCD switches.**

We suggest that you use an external power supply to drive the “Opto 22” I/O card. This power supply can be 5, 15 or 24Vdc, and it does need to match your I/O module voltage ratings.

Some expansion board manufacturers have jumpers on their cards that can feed the logic supply voltage back into the MX. Make absolutely sure that no voltage is being fed back into the MX (by removing the jumpers, if necessary), as severe damage can occur.

## SECTION VI

### CHECKING SYSTEM I/O USING TERMINAL MODE

To check proper functioning of the system I/O:

- Use the supplied serial cable to connect your PC's serial port to the MX2000 host serial port.
- Run the MX2000 programming software.
- REMOVE the CLR input jumper on the DSP card.
- Wire 110 VAC to the MX on the power supply card.

**WARNING: Power is about to be supplied to the MX for the first time. Check all connections for short or open circuits and remove all personnel from the area involving motion.**

- Apply power and confirm that the POWER ON LED is illuminated.
- Within the MX2000 software, click the TERMINAL button to enter terminal mode.

Type: **DIR** (↵)

If the screen informs you of what is contained within the MX, you are communicating and you may proceed with this I/O function test. If NO information was displayed, troubleshoot your RS232 cable, connections, and ALL serial settings (such as COM port selected, BAUD rate, PARITY etc.). Serial settings in the MX2000 software are set by clicking on the System menu, then selecting "Terminal settings" and then "Com port...".

Type: **HARDLIMPOS** (↵)

You will see the current status (i.e. status at the time you hit the Enter key) of the LIM+ inputs for each axis in your system. Inactive inputs will return a 0, while active inputs will return a 1 to your screen.

Manually force and hold each positive limit switch ON, re-type HARDLIMPOS and verify that the input change is recognized. Release input and re-type. Verify input change.

Type **HARDLIMNEG** (↵) and repeat the above process for the LIM- inputs.

Type **EVENT1(axis #)** (↵) and repeat for the HOME switch input (if used).

Type **EVENT2(axis #)** (↵) and repeat for the MARK REGISTRATION input (if used).

Type **IN(xyy)** (↵) for any Digital Inputs wired to any Digital Input card.\*

Type **EXIN(xyy)** (↵) for any Expansion Inputs wired to any Expansion I/O cards.\*

\* where "x" is the Input board # and "yy" is the I/O location #. Example: IN(102) or EXIN(210)



## SECTION VI - continued.

**WARNING - The next steps will actually turn ON outputs and drive analog output channels. Be sure you are safe and ready to engage these features.**

- Type **OUT(xyy) = 1** (↵) to turn ON outputs wired to any Digital I/O card.
- Type **OUT(xyy) = 0** (↵) to turn OFF outputs wired to any Digital I/O card.
- Type **EXOUT(xyy) = 1** (↵) to turn ON outputs wired to any Expansion I/O card.
- Type **EXOUT(xyy) = 0** (↵) to turn OFF outputs wired to any Expansion I/O card.
- Type **ANALOG(xyy)** (↵) to check the status of an analog channel.
- Type **ANALOG(xyy) = x** (↵) to set an analog channel to “x” voltage out.



### SECRETS AND HINTS

Other uses for TERMINAL mode include:

Type **SPEED** (↵) to view the present speed settings for all axes.

Type **SPEED = x,x** (↵) to SET the speed on Axis 1 & 2, where x is the speed in units/second.

Type **SPEED(3) = x** (↵) to SET the speed on Axis 3 only.

Type **MOVE(1) = x** (↵) to MOVE axis 1 a distance of x units.

Type **MOVEHOME = -1,-1,-1,-1** (↵) to move Axes 1 through 4 to Mechanical Home, in the - direction. (This assumes that home switches have been wired to the EVENT1 input on each axis.)

HOST commands, as listed and described on pages 6-147 through 6-149 in the MX2000 Instruction Manual, are commands that can be executed from the terminal mode, as shown above.

WARNINGS and/or FAULTS that might occur within the MX2000 (triggering the FAULT light on the front panel of the DSP card) can be identified from within terminal mode by typing: **ERRM** (↵). A complete list of system warnings and faults can be found on page 6-77 of the MX2000 Instruction Manual.

## SECTION VII

### CREATING A PROJECT (PROGRAM)

#### MX2000 Programming Terminology:

**PROJECT** The file that will contain all the information needed to complete an application. Configuration information is stored in the project file, along with the names of the task files that are associated with the project. The project filename will have an extension of **.PRJ** in your computer.

**TASK** This file is a list of program instructions. Remember that the MX2000 is a multi-tasking device and has the capability of running up to 7 tasks at the same time. Most applications will only require one or two tasks, but more are available if needed. *One task will be opened, named, and added to the PROJECT file when the project file is created.* The task filename will have an extension of **.TSK** in your computer.

**CONFIGURATION** This information is stored in the project file. A configuration window in the MX2000 software lets you set up all system parameters for your particular application. In general, these parameters are set once and then left alone.

- Create a new Project File** (Task #1 will also be opened and added at this time)

Run the MX2000 Software

Click on: **CREATE NEW PROJECT**

Name your project file with 8 or less characters, leaving the extension as **.PRJ**.  
(The MX2000 software **does not** recognize Windows 95 long files names.)

Select your controller type and click OK. (MX2000-xx, DCS, MX2000-1C or TDC.)

(At this time, you have created a new project file with one task. Notice this task carries the same name as was given to the Project file)

- Maximize** the newly created **Axis Configuration** window

## SECTION VII - continued.

**If needed, add more tasks to the project.** (ONLY IF NEEDED — if you are unsure, assume that they are not needed for now. You can always come back and add tasks to the project.)

To add a task to the project file, select **New** from the **Task** menu. This will create a blank Task window. The name the task, select **Save As...** from the **Task** menu. In the File Name box, give this new task a name, leaving the extension as **.TSK**.

To add this new task to the project with your other tasks, be sure that the new task window is still showing. Select **Add Task...** from the **Project** menu. Verify that the proper task is about to be added and then click YES.

Repeat this process to add as many tasks as are required by the application.

**Set up the Configuration for your project.**

Click on the **CONFIGURATION** button at the top of the screen. A window is displayed that is similar to a file drawer with 10 folders. Click on each tab and fill in all needed information for ALL AXES. Be sure to use the scroll bar at the bottom of the window to see all configuration parameters under each tab.

**Be sure to read the Secrets and Hints on Configuration beginning at the bottom of this page!**

**PLEASE DO NOT CREATE YOUR APPLICATION PROGRAM AT THIS TIME.**



### SECRETS AND HINTS

READ pages 6-22 through 6-32 of the MX2000 Instruction Manual for detailed descriptions of the Configuration Tabs.

See the next page for MORE Secrets and Hints on Configuration.

## SECTION VII - continued.



### MORE SECRETS AND HINTS

To avoid programming confusion, keep the number of tasks to a minimum.

To toggle between tasks, as only one can be viewed at a time, click on the **Window** menu. The open task files are listed at the bottom of the menu and can be easily selected with your mouse.

On the **System** tab in the Configuration window, be sure to assign a task to each axis of motion. *Note that a task may control more than one axis of motion, but an axis can only be assigned to (i.e. be controlled by) one task.*

**Encoder Tab** - All standard Superior servo amplifiers provide a simulated encoder output of 1024 lines, yielding 4096 pulses per revolution. Enter this information here.

**System Tab - Units per motor revolution** - This is where you scale your axes to linear or rotary units of your choice, such as inches, feet, millimeters, degrees, radians or any other unit. This allows you to program the movement of your actual load as opposed to pulses or steps on your motor shaft. Follow this procedure for each axis in your system:

- 1) Decide which unit of measure works best the axis.
- 2) Calculate or measure the EXACT number of units the load will move per single revolution of the motor shaft. Enter this number in the “Units per motor revolution” box for that axis.

#### **Examples:**

*If you selected the unit of measure “INCH”, you have a load being carried on linear bearings, and it is driven by a 0.2” lead ball screw directly coupled to the motor shaft:*

How many inches would this load travel if the motor shaft was turned exactly one revolution?

Answer: 0.2 inches. You would enter 0.2 in the “Units per motor revolution” box.

If you have had the same ball screw, but included a 20:1 gearbox between the motor and the ball screw, what is the number of units per motor revolution?

Answer: 0.01 inches.

*If your load is a rotary load and you selected degrees as your load unit of measure:*

If your motor is directly coupled to the center of your load, with no in-line gearbox added, how many degrees would this load rotate for one exact revolution of the motor shaft?

Answer: 360°. You would enter 360 in the “Units per motor revolution” box.

If your motor is coupled to the center of your load, but with a 3:1 in-line gearbox added, how many degrees would this load rotate for one exact revolution of the motor shaft?

Answer: 120°

## SECTION VIII

### TUNING A SERVO MOTOR WITH THE MX2000

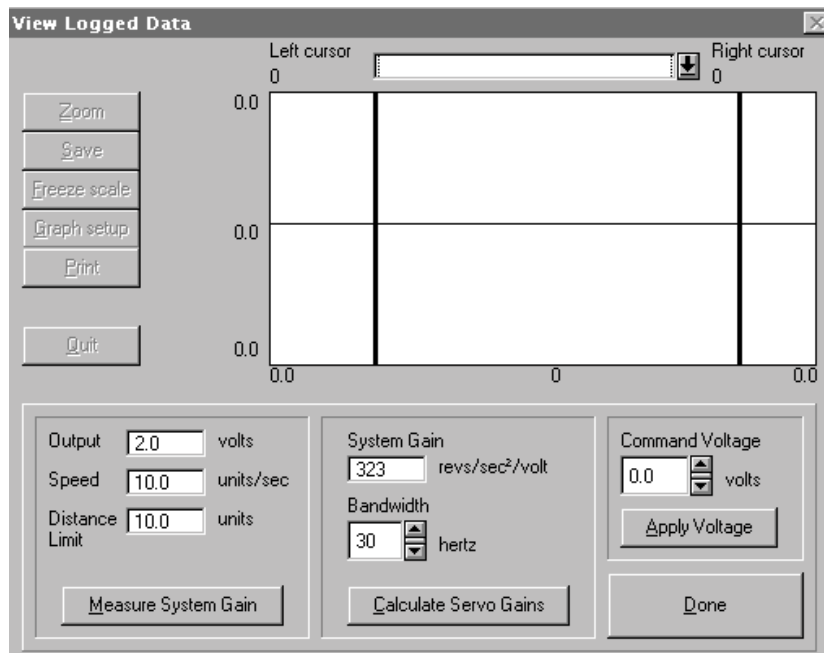
Skip this section if you are using either Open or Closed Loop Steppers and proceed to Section IX.

Follow these steps for each servo axis in your application. We will first AUTOTUNE each servo axis and then provide you with a procedure for manually tuning each axis, should you need to do so.

- Confirm that all axes have been fully wired properly.
- Click on the CONFIGURATION button, then on the System tab to make sure your servo axes have been configured as servo axes.
- Click on the Encoder tab to make sure that the correct number of lines and pulses per revolution have been entered. (Standard Superior servos use a simulated quadrature encoder of 1024 lines yielding 4096 PPR.)
- COMPILE your Project file by clicking on the COMPILE PROJECT button. Proceed only if you receive the message “Success - No Errors”. If errors were discovered, fix them and then repeat this COMPILE PROJECT procedure. (Note that it is not necessary to have written a task file at this point.)
- Connect your RS232 cable from your computer’s serial port to the MX2000.
- Open the CLR input (i.e., remove the jumper) on the DSP card.
- Apply power to the MX2000.
- Click on the Terminal button to enter terminal mode and verify that you are communicating. If you are not communicating, solve your RS-232 related problem before proceeding.
- DOWNLOAD the Project by clicking on the DOWNLOAD PROJECT button. Confirm that the downloading is complete before proceeding.

## TUNING A SERVO MOTOR WITH THE MX2000

### THE AUTO TUNING PROCESS



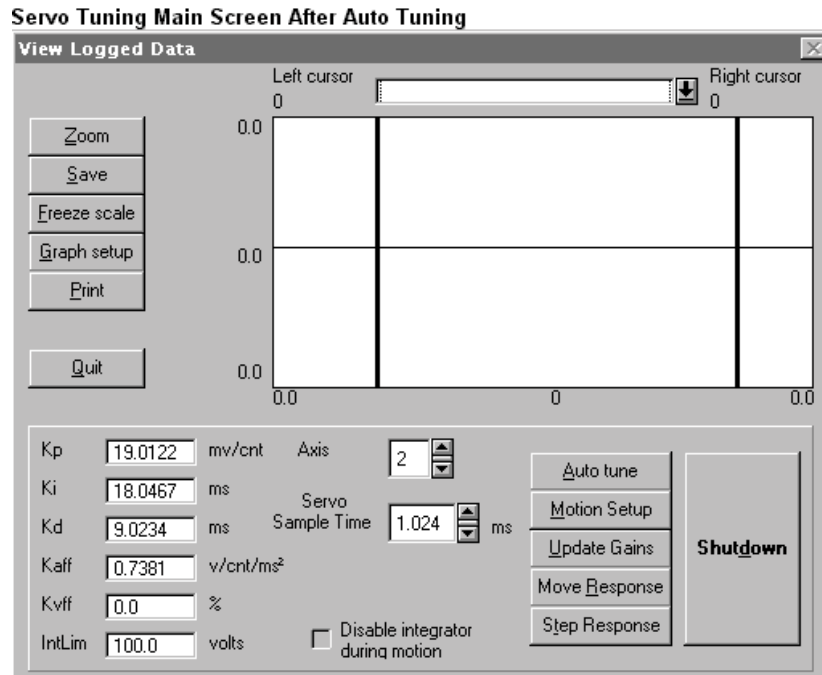
**This tuning procedure must be run with the motor connected to your production load. This is a loaded shaft tune.**

- Read and understand pages 6-33 through 6-41 of the MX2000 manual.
- Activate the CLR input on the DSP card (i.e., the jumper should be installed).
- Click the SERVO TUNING button. The tuning screen should now be displayed.
- Select the Axis Number you wish to tune at this time by clicking the axis spin controller. This is located in the center of your screen.
- Click the AUTO TUNE button.
- Enter values for OUTPUT, SPEED, & DISTANCE LIMIT exactly as described on page 6-36 of the MX2000 Instruction Manual.

**WARNING: A short surge of energy is about to be sent to your selected servo axis. Take all needed safety precautions at this time to clear personnel away from the motion devices.**

- Set BANDWIDTH (see Secrets and Hints).
- Click the MEASURE SYSTEM GAIN button. (You'll hear a surge in your motor, and then see a measured and calculated numerical representation of your motor and load in the System Gain Box.)
- Click the CALCULATE SERVO GAINS button. (This updates Kp, Ki, Kd, & Kaff on both the previous screen and in Configuration.)
- Click the QUIT button to return to the previous screen. (Observe the gain values set by the auto tuning routine.)

**Auto tuning is now complete. Repeat this process for all servo axes, and then proceed to the manual tuning section, if necessary.**



## SECRETS AND HINTS

**Bandwidth** is defined as the number of velocity changes to be followed per second. This is the responsiveness of your motor. This is how many times a second (Hz) are you trying to update the shafts position when compared to desired position. The higher your load to rotor ratio is, the lower the bandwidth number must be. If the AUTO TUNING procedure won't stabilize your load, you must have a high load to rotor inertia ratio. In this case, lower the bandwidth by 20% or more and repeat as needed. Factory default is 30 Hz.

Larger frame size motors may also require a reduction in bandwidth from default. Likewise, it may be possible to increase the bandwidth from the factory default setting on smaller frame size motors.

## SECTION VIII - continued.

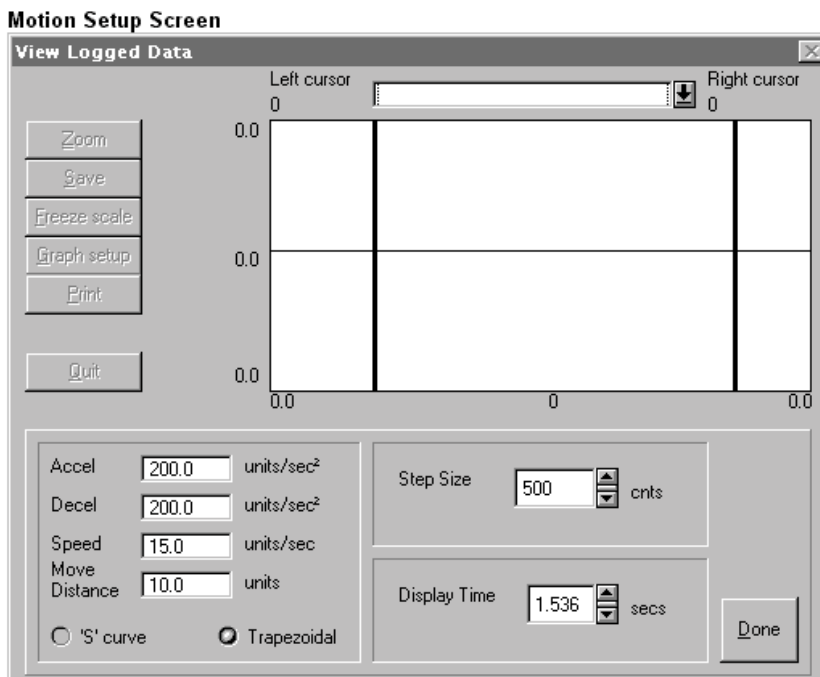
### MANUALLY TUNING A SERVO MOTOR WITH THE MX2000

This section will assist you in manually tuning your servo motor(s). This procedure should only be attempted after performing an AUTO TUNE in Section VIII. In most cases, the gain values that were measured and calculated with the Auto Tune procedure will give satisfactory performance, and no manual adjustment will be required. Only use this manual tuning procedure if:

- 1) Auto Tuning was unsuccessful due to “wind up” (elasticity) in your load.
- 2) Tighter shaft performance is needed during the movement.
- 3) You want to test and measure your error during movement and at motor rest.

Assumptions: you have just exited the Auto Tuning area and you have a fully loaded and connected system.

- Ensure that the CLR input is connected on the front panel of the DSP card.
- Click the SERVO TUNING button and observe the opening tuning screen.
- Click the MOTION SETUP button.
- Enter low values for ACCEL, DECEL, SPEED & MOVE DISTANCE.
- Select the desired profile shape — TRAPEZOIDAL or “S” CURVE.
- The parameters needed for a simple move are now complete. Click the DONE button, which returns you to the opening tuning screen.



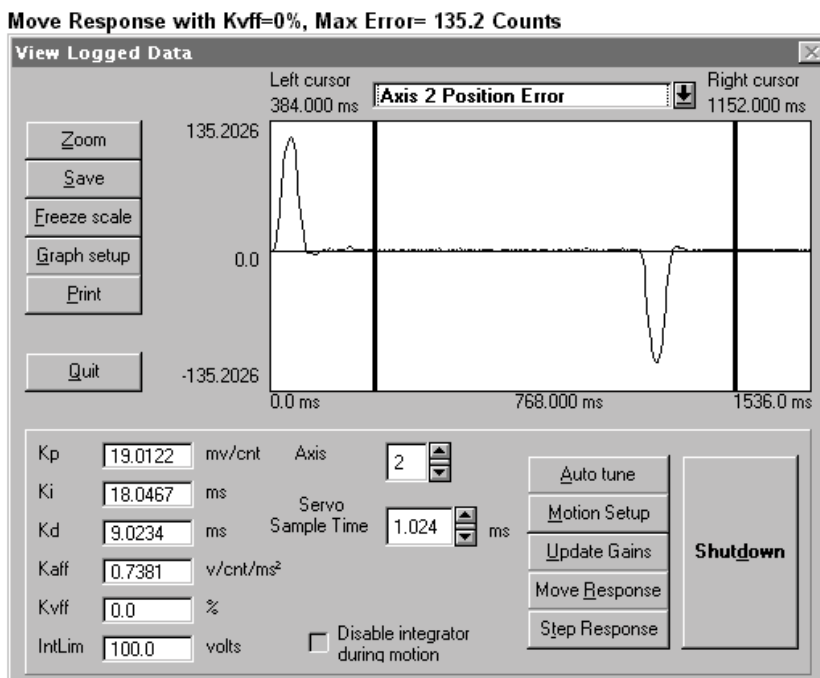
**WARNING** — a complete move is about to be executed. Make sure that all personnel are safely



**removed from the area and check to ensure the motor can complete its move without causing a stall or motor jam condition.**

- ❑ Locate and observe the SHUTDOWN button on the computer screen. Use this button to shut the servo motor down in case of system instability.
- ❑ Click the MOVE RESPONSE button. Within a 2 second delay, observe the previously configured motion routine being executed and then watch your PC's screen as the logged data is transmitted from the MX and into your computer. Next, observe the displayed error waveform.

**Follow the practice of “Doubles and Halves” for Kp, Ki, & Kd ONLY. If a gain value needs to be increased, then double it. If a gain value needs to be decreased, then halve it. After you have made a change, click the UPDATE GAINS button, and then repeat the MOVE RESPONSE process to observe how this change affects the error waveform. Repeat this “change, update, move, and observe” process for EVERY gain adjustment, no matter how slight.**

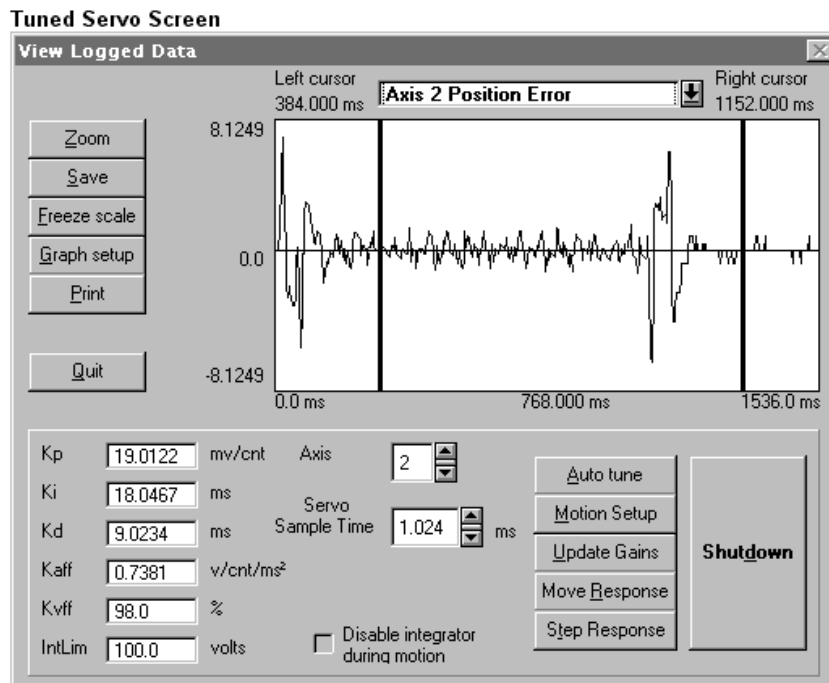


In general:

- ❑ Write down all gain values that were originally set by the Auto Tune procedure. (This will ensure that you can come back to them if needed.)
- ❑ Adjust Kp to smooth out the Accel and Decel portions of the profile. The profile should begin to resemble a trapezoid shape. Too much Kp will cause motor vibrations to be felt and/or observed, while too little Kp will produce large lead or lag errors during the accel and decel portions of the move.
- ❑ Adjust Kd until the top of the trapezoid is as smooth as possible with no overshoot. If there is too much Kd, you'll undershoot because you have electrically “damped” the shaft response.
- ❑ Set Kvff to 50%. Click Update Gains and then click Move Response. This should significantly

reduce the error, but too high a setting can produce excessive motor heating. If the steady state area (the flat part) is below the 0.0 line; decrease Kvff. If the steady state area is above the 0.0 line; increase Kvff. Make all adjustments to Kvff in 5% increments.

- Decrease Ki to reduce any oscillations that might exist after the movement has completed and/or while the shaft should be at rest. (Make small changes here, in 1 mS increments.)
- Adjust Kaff to reduce error during acceleration and deceleration. This value of this number is usually fractional (i.e., less than 1). Make slight changes here; if any at all.
- Under MOTION SETUP, increase your ACCEL, DECEL, SPEED, & MOVE DISTANCE to values that will be used in the actual application.
- Repeat this entire process until you have the servo running exactly as your application requires.
- Your tune should now be complete. Compile your project to save the gains into the configuration.
- Repeat the process for each servo axis.



## SECTION IX

### TEST MOTION PROGRAM (STEPPER)

This section will help you create, load and run a sample motion program for a single axis of motion for either an open or closed loop stepper. *Please skip this section and proceed to Section X if you need to test an axis of servo motion.* Running this program will confirm that all your hardware, connections, firmware and software are functioning normally. At this time, you need to have the following:

MX2000 programming software open and running.

Your previously created PROJECT & empty TASK file(s) should be open.

Your PC is connected to your MX2000 via a serial link and communications have been tested.

Your entire hardware system is wired correctly and is trouble free.

- If you have not done so already, return to Section VII and create a PROJECT & TASK program.
- Click on the CONFIGURATION button and then click on each Tab. Verify that the settings are correct for your application. You should have done this in Section VII.

**NOTE:** We are about to make a complete program generated move. **You must ensure that this is safe to do.** The program below is generic and can in no way be written for all circumstances and applications. Before you download this program, make absolutely sure that you can safely make the move that is called for, without risk to personnel or damage to your machine. If you wish, you can run this sample program with your motor shaft uncoupled from the load.

**The values below are generic and can be changed if needed. Please review them and see if they make sense for your application.**

Assumption:

There is a stepping motor axis (open or closed loop) assigned in the configuration screen to Axis #1.

- Type the following program into your TASK file:

```
SPEED=2           `sets the speed to 2 units/sec.
ACCEL=50          `sets the acceleration rate to 50 units/sec/sec.
DECEL=50          `sets the deceleration rate to 50 units/sec/sec.
ABSPOS=0          `sets the absolute position counter to zero.
POSMODE=0         `sets the position mode to incremental type.

MOVE=1            `move the load 1 unit.
WAITDONE=1        `waits for motion to complete
END               `end of program
```

**Note:** Insert the command ENCMODE=2, just after the POSMODE command, **only** if you have a closed loop stepper axis to test.

## SECTION IX - continued

- Click on the COMPILE PROJECT button to compile the project.
- Fix any program errors discovered by the compiler and re-compile. Repeat the COMPILE procedure until error free.
- Click on the DOWNLOAD PROJECT button to download the project into the MX2000.
- Click on the TERMINAL button to enter the terminal mode.
- Type DIR (↵).
- The correct PROJECT and TASK file(s) are present and loaded if the screen displays the following:

```
0*S  PROJECT          (this is the project name, without the .PRJ extension)
      TASK            (this is the task name, without the .TSK extension)
free space = xxxx
```

**Note:** the presence of the asterisk, as shown above, indicates that the program has been loaded into active RAM memory and it is ready to be run. This asterisk must be present to RUN the program. If it is not present, ensure that the CLR jumper connection is in place on the DSP card.

**WARNING: The program is about to begin running and MOTION is about to occur.**

- Type RUN (↵).
- Observe the load moving the distance that was specified in the MOVE statement within the task file.
- If the motion did not occur as expected, troubleshoot for the problem and make the necessary changes. RUN the test program again and check for proper load movement. Repeat this process until proper movement is observed.

## CONGRATULATIONS !!!

At this time, you should have a completely wired, programmed, tested and functioning system. You can now begin to develop your machine's application program and program your desired functions into the MX2000.

### **A STRONG SUGGESTION:**

We strongly suggest that you create a flow chart of your machine's function and logic first, before you begin programming the MX. Meet with all parties involved in the design of the machine, and establish exactly what your machine needs to do first, then second, then third and so on. Create a flow chart of this sequence first, then translate this sequence of events into MX2000 code by programming your task file(s). We believe that this will simplify the task of programming the MX2000.

- Create your program.

## SECTION X

### TEST MOTION PROGRAM (SERVO)

This section will help you create, load and run a sample motion program for a single servo axis of motion. Please refer to Section IX if you need to test a stepping motor axis of motion. Running this program will confirm that all your hardware, connections, firmware and software are functioning normally. At this time, you need to have the following:

MX2000 programming software open and running.

Your previously created PROJECT & empty TASK file(s) should be open.

Your PC is connected to your MX2000 via a serial link and communications have been tested.

Your entire hardware system is wired correctly and is trouble free.

- If you have not done so already, return to Section VII and create a PROJECT & TASK program.
- Click on the CONFIGURATION button and then click on each Tab. Verify that the settings are correct for your application. You should have done this in Section VII.
- If you have not do so, tune each servo axis. Refer to Section VIII for assistance.

**NOTE:** We are about to make a complete program generated move. **You must ensure that this is safe to do.** The program below is generic and can in no way be written for all circumstances and applications. Before you download this program, make absolutely sure that you can safely make the move that is called for, without risk to personnel or damage to your machine. If you wish, you can run this sample program with your motor shaft uncoupled from the load. You may need to re-tune for an unloaded shaft, should you choose this path.

**The values below are generic and can be changed if needed. Please review them and see if they make sense for your application.**

Assumptions:

There is a servo motor axis assigned in the configuration screen to Axis #1.

- Type the following program into your TASK file:

```
SPEED=2           `sets the speed to 2 units/sec.
ACCEL=50          `sets the acceleration rate to 50 units/sec/sec.
DECEL=50          `sets the deceleration rate to 50 units/sec/sec.
ABSPOS=0          `sets the absolute position counter to zero.
POSMODE=0         `sets the position mode to incremental type.
WNDGS=1           `enable the servo amplifier
WAIT=0.3          `provides a short delay to allow servo to energize

MOVE=1            `move the load 1 unit.
WAITDONE=1        `waits for motion to complete
END               `end of program
```

## SECTION IX - continued

- Click on the COMPILE PROJECT button to compile the project.
- Fix any program errors discovered by the compiler and re-compile. Repeat the COMPILE procedure until error free.
- Click on the DOWNLOAD PROJECT button to download the project into the MX2000.
- Click on the TERMINAL button to enter the terminal mode.
- Type DIR (↵).
- The correct PROJECT and TASK file(s) are present and loaded if the screen displays the following:

```
0*S  PROJECT          (this is the project name, without the .PRJ extension)
      TASK            (this is the task name, without the .TSK extension)
free space = xxxx
```

**Note:** the presence of the asterisk, as shown above, indicates that the program has been loaded into active RAM memory and it is ready to be run. This asterisk must be present to RUN the program. If it is not present, ensure that the CLR jumper connection is in place on the DSP card.

**WARNING: The program is about to begin running and MOTION is about to occur.**

- Type RUN (↵).
- Observe the load moving the distance that was specified in the MOVE statement within the task file.
- If the motion did not occur as expected, troubleshoot for the problem and make the necessary changes. RUN the test program again and check for proper load movement. Repeat this process until proper movement is observed.

## CONGRATULATIONS !!!

At this time, you should have a completely wired, programmed, tested and functioning system. You can now begin to develop your machine's application program and program your desired functions into the MX2000.

### **A STRONG SUGGESTION:**

We strongly suggest that you create a flow chart of your machine's function and logic first, before you begin programming the MX. Meet with all parties involved in the design of the machine, and establish exactly what your machine needs to do first, then second, then third and so on. Create a flow chart of this sequence first, then translate this sequence of events into MX2000 code by programming your task file(s). We believe that this will simplify the task of programming the MX2000.

- Create your program.

## APPENDIX A

### TEST OPERATOR INTERFACE PROGRAM (IWS120SE / 127SE)

This section will assist you in configuring, wiring and programming either an IWS120SE or an IWS127SE.

- Set COM1 for RS-232 communications on the IWS. Consult the manual for proper DIP switch settings.
- Wire and apply 120VAC to your Operator Interface.
- Verify that your unit contains TERMINAL firmware and *not* SE BASIC firmware. After the initial power-up self checks have been performed, the screen should read “IWS Terminal Mode – Screen will clear when any key hit or character received.” if you have TERMINAL firmware installed. If this appears on your display, then continue on to the next “action box.”

IF your display indicates that you have SE BASIC firmware installed in your IWS display, contact your Superior Electric Representative or the factory for further instructions. They will provide you with software and instructions on how to upload the Terminal software into your display. You cannot continue with this section until you have the Terminal firmware properly installed.

- Remove power from the IWS and the MX.
- Connect and wire an RS-232 cable between your MX2000 and the IWS display. The connections are as follows:

<u>MX2000 Auxiliary Terminals</u>	to	<u>IWS PIN #</u>
GND		PIN # 7
RX -		PIN # 2
TX-		PIN # 3

- Connect IWS PIN # 4 to IWS PIN # 5.
- Verify the Auxiliary Port communications setting. **Remove power** from the MX2000 and remove the DSP card. Locate jumper J10 (far upper left corner) and confirm the RS-232 setting. Change if needed. Re-install card. Do NOT apply power to the MX2000.
- Install communications cable between MX2000 and the IWS Terminal.
- Remove the jumper from the CLR input on the DSP Card. (This for safety reasons only.)
- Apply power to the MX2000 and to the IWS Display. Wait for IWS self checks to complete.
- Replace the CLR input jumper on the DSP Card.
- Enter Terminal Mode.
- Type ERASE DIR (↵)
- Exit terminal mode.
- Create a new PROJECT and TASK file. (There’s no need to go through CONFIGURATION for this test)
- In the TASK window, type the following:

```
SETCOM#2,9600,0,8,1      `sets up communications from the MX to the IWS
PRINT#2, "IT WORKS FINE" `PRINT statement
END                      `program end
```

## APPENDIX A - continued

- Click on the **COMPILE PROJECT** button to compile the project.
- Fix any program errors discovered by the compiler and re-compile. Repeat the **COMPILE** procedure until error free.
- Click on the **DOWNLOAD PROJECT** button to download the project into the MX2000.
- Click on the **TERMINAL** button to enter the terminal mode.
- Type **DIR (↵)**.
- The correct **PROJECT** and **TASK** file(s) are present and loaded if the screen displays the following:

```
0*S PROJECT          (this is the project name, without the .PRJ extension)
      TASK           (this is the task name, without the .TSK extension)
free space = xxxx
```

**Note:** the presence of the asterisk, as shown above, indicates that the program has been loaded into active RAM memory and it is ready to be run. This asterisk must be present to **RUN** the program. If it is not present, ensure that the **CLR** jumper connection is in place on the **DSP** card.

- Type **RUN (↵)**.
- Check the **IWS** display — it should now read “**IT WORKS FINE**” on the screen.

If it does not, you should troubleshoot your **DSP** card jumper settings, your cable connections and cable integrity and/or the program statements in your task. Check to make sure the **CLR** input jumper is installed on the **DSP** card. You should also verify that your program is active. (This is done in terminal mode by checking for the presence of the asterisk, as described above.)



## APPENDIX B

### SAMPLE PROGRAM #1 - CUT TO LENGTH APPLICATION

These sample programs are included to provide programmers with examples of code and are not necessarily intended to be used for real world application solutions. All the sample programs and the IWS.INC file have been added to the diskette included with this guide. We strongly suggest that you follow standard "copy and paste" techniques to obtain perfect copies of these programs.

**All of the program examples in this guide were written for MX2000 Version 3.4 and its corresponding IWS.INC file. These applications will give numerous compilation errors if you attempt to compile while using an older version of the IWS.INC file. See APPENDIX D for technical support options, should you need a software upgrade.**

The following program was written to handle a basic feed-to-stop / cut-to-length application.

Through an IWS127SE operator interface panel, the operator will be able to select the number of indexes and the length of each index. The operator will also be able to feed the product forward and then cycle the cutting mechanism to set up a new supply of material.

```
*****
' MEMORY LOCATIONS UTILIZED:
' NVR(1) used to store the total number of indexes needed.
' NVR(2) used to store the index length.
' NVR(3) used to store the number of indexes that have occurred.
*****

#INCLUDE C:\MX2000\IWS.INC      'path MUST be correct to locate IWS.INC
SETCOM#2,9600,0,8,1           'sets COM parameters to the IWS
WAIT=21                        'Allows IWS127 self test to complete. 21 is min.
FLUSH_RX_BUFFER               'clears the receiving buffer of trash
CLRSCRN                       'clears the IWS screen
CURSOR_OFF                    'turns the cursor OFF
DISP 1,1,"Superior Electric"  'displays the words SUPERIOR ELECTRIC
DISP 3,11,"DANA Corp."       'same
WAIT=3                         'delays for 3.0 seconds

OPENING_SCREEN:               'label
    CLRSCRN                   'clears the screen
    CURSOR_OFF                'turn the cursor off
    DISP 1,1,"F1: Cycle"      'display statement
    DISP 3,1,"F2: Feed and Cut" 'display statement
    DO
        KEYPRESS=INCHAR(2)    'sets KEYPRESS equal to button pushed.
    LOOP UNTIL KEYPRESS=F1 OR KEYPRESS=F2 'loops until F1 or F2 is pressed
    IF KEYPRESS=F1 THEN GOTO SET_UP1 'goto statement
    IF KEYPRESS=F2 THEN GOTO SET_UP2 'goto statement
```

## APPENDIX B - continued

```
SET_UP1:
  CLRSCRN                'clears the IWS display
  CURSOR_ON              'turns the cursor ON
  DISP 1,1,"# of Indexes" 'display statement
  FMT$="000"             'format display to xxx number
  FDISP 1,17,NVR(1)     'display NVR(1) on row 1, column 17 per FMT
  DISP 2,1,"Length"     'display statement
  FMT$="000.000"        'format display to xxx.xxx number
  FDISP 2,13,NVR(2)     'display NVR(2) on row 2, column 13 per FMT

  DISP 4,1,"F1: Begin"  'display statement
  DISP 4,10,"F2:Change#" 'display statement

  MIN_VAL = 1 : MAX_VAL = 999 'sets up acceptable input range of #s.
  FMT$= "000"              'format display to xxx number
  VALUE$=STR$(NVR(1))     'sets value$ to NVR(1)
  ROW=1 : COL=17 :       'calls the MSB location of NVR(1)
  GOSUB EDIT              'use Edit subroutine to handle $
  NVR(1) = VAL(NEWVALUE$) 'sets NVR(1) to new number for storage

  MIN_VAL = 0.001 : MAX_VAL = 999.999 'same as above
  FMT$="000.000"         'same as above
  VALUE$=STR$(NVR(2))    'same as above
  ROW=2 : COL=13        'same as above
  GOSUB EDIT            'call the EDIT subroutine
  NVR(2)= VAL(NEWVALUE$) 'set NVR(2) equal to a newvalue

  DO                      'loop until F1 or F2 is pressed
    KEYPRESS=INCHAR(2)    'set variable (KEYPRESS) to IWS button press
  LOOP UNTIL KEYPRESS=F1 OR KEYPRESS=F2
                          'exit loop if F1 or F2 were pressed

  IF KEYPRESS=F2 THEN GOTO SET_UP1 'if F2 was pressed, go to a label location

RUNNING_SCREEN:          'label
  CLRSCRN                'same as above
  CURSOR_OFF             'turn cursor off
  SPEED=20               'set speed on Axis 1 to 20 units/sec.
  NVR(3)=NVR(1)          'sets NVR(3) equal to NVR(1)
  FOR X = 1 to NVR(1)    'FOR / NEXT LOOP
    JUMP_BACK:           'label
    DISP 4,1,"F1:STOP AFTER CYCLE" 'display
    DISP 1,1,"Indexes left:"      'same as above
    KEYPRESS=INCHAR(2)            'same as above
    IF KEYPRESS=F1 THEN GOSUB HOLD_SCREEN 'same as above
    MOVE(1)= NVR(2)               'move axis 1 by NVR(2) amount in units.
    NVR(3)=NVR(3)-1              'subtraction
    FMT$="000"                   'same as above
    FDISP 1,15,NVR(3)            'same as above
    WAITDONE=1                   'waits for motion to complete at this time
    EXOUT(111)=1                 'turns ON expansion output #11, on board #1
    WAIT=0.5                     'delay for 0.5 second
    EXOUT(111)=0                 'turn the output back off
  NEXT X                          'ends the FOR/NEXT LOOP

GOTO OPENING_SCREEN            'goto statement
```

## APPENDIX B - continued

```
HOLD_SCREEN:
  CLRSCRN
  CURSOR_OFF
  DISP 1,1,"Cycle has stopped"
  DISP 3,1,"F1: resume cycle"
  DISP 4,1,"F2: abort cycle"

  DO
    KEYPRESS=INCHAR(2)
  LOOP UNTIL KEYPRESS=F1 OR KEYPRESS=F2

  CLRSCRN
  IF KEYPRESS=F1 THEN
    GOTO JUMP_BACK
  ELSE
    GOTO OPENING_SCREEN
  END IF

SET_UP2:
  CLRSCRN
  CURSOR_OFF
  DISP 1,1,"F1: Jog Forward"
  DISP 2,1,"F2: Fire Cutter"
  DISP 4,1,"F4: Main Screen"

  DO
    KEYPRESS=inchar(2)
  LOOP UNTIL KEYPRESS=F1 OR KEYPRESS=F2 OR KEYPRESS=F4

  IF KEYPRESS=F1 THEN GOTO JOG_ROUTINE
  IF KEYPRESS=F2 THEN GOTO OUTPUT_ROUTINE
  IF KEYPRESS=F4 THEN GOTO OPENING_SCREEN

JOG_ROUTINE:
  CLRSCRN
  CURSOR_OFF
  DISP 1,7,"JOGGING!!"
  DISP 4,1,"F1: Jog STOP"
  SPEED=1
  JOGSTART(1)=1
  DO
    KEYPRESS=inchar(2)
  LOOP UNTIL KEYPRESS=F1
  JOGSTOP(1)=1
  CLRSCRN
  DISP 2,3,"Jogging Stopped"
  WAIT=1.0

  GOTO SET_UP2

OUTPUT_ROUTINE:
  CLRSCRN
  CURSOR_OFF
  DISP 2,1,"Cutter has fired"
  EXOUT(111)=1
  WAIT=0.2
  EXOUT(111)=0
  GOTO SET_UP2
```

## APPENDIX B - continued

```
EDIT:
'*****
'IWS120 SERIES TERMINAL MODE EDIT SUBROUTINE
'
'The following variables are required:
'   VALUE$ - a string containing the initial value to edited
'   FMT$ - the display format of the data ex. fmt$="+000.000"
'   ROW - the row where the variable is to be displayed
'   COL - the column where the variable is to be displayed
'   MIN_VAL - the minimum value of the variable
'   MAX_VAL - the maximum value of the variable
'   if minval and maxval are both zero no range check is performed
'   NEWVALUE$ - a string containing the new value
'*****
ECHO_OFF
BEEP_ON
CURSOR_ON
FDISP row,col,VAL(value$)

    dp_pos = INSTR(fmt$,".")
    plus_sign=INSTR(fmt$,"+")
    minus_sign=INSTR(fmt$,"-")
    IF plus_sign > 0 OR minus_sign > 0 THEN
        IF dp_pos > 0 THEN dp_pos = dp_pos + col - 1
    ELSE
        IF dp_pos > 0 THEN dp_pos = dp_pos + col
    END IF
    cursor = col + 1
    start_col = col + 1
    end_col = col + LEN(fmt$)

    IF LEFT$(value$,1) = "-" THEN
        minus = 1
    ELSE
        minus = 0
    END IF

    IF cursor = dp_pos THEN cursor = cursor + 1

    FLUSH_RX_BUFFER

    DO
        LOCATE row,cursor
        GOSUB GetKeyPress          'wait for a key to be pressed

        IF keypress = BACKSPACE THEN
            PRINT#2,"0";
            cursor = cursor - 1
            IF cursor = dp_pos THEN cursor = cursor-1
            IF cursor < start_col THEN
                BELL
                cursor = start_col
            END IF
            IF cursor = dp_pos THEN cursor = cursor+1
        END IF

        IF keypress = LEFT THEN
            cursor = cursor - 1
            IF cursor = dp_pos THEN cursor = cursor - 1
            IF cursor < start_col THEN
                BELL
                cursor = start_col
            END IF
            IF cursor = dp_pos THEN cursor = cursor + 1
        END IF
    END IF
```

## APPENDIX B - continued

```
    IF keypress = RIGHT THEN
        cursor = cursor + 1
        IF cursor = dp_pos THEN cursor = cursor + 1
        IF cursor > end_col THEN
            BELL
            cursor = end_col
        END IF
        IF cursor = dp_pos THEN cursor = cursor + 1
    END IF

    IF keypress >= 48 AND keypress <= 57 THEN
        PRINT#2,CHR$(keypress);
        cursor = cursor + 1
        IF cursor = dp_pos THEN cursor = cursor + 1
        IF cursor > end_col THEN cursor = end_col
        IF cursor = dp_pos THEN cursor = cursor + 1
    END IF

    IF keypress = MINUS_SIGN THEN
        minus = minus ^ 1
        IF minus = 1 THEN
            DISP row,col,"-"
        ELSE
            DISP row,col,"+"
        END IF
    END IF

    IF keypress = DECIMAL_PT AND dp_pos > 0 THEN cursor = dp_pos + 1
LOOP UNTIL keypress = ENTER
'added by AH
'LOOP UNTIL keypress=ENTER OR keypress=CANCEL OR keypress=UP OR keypress=DOWN
'commented out by AH
CURSOR_OFF
IF keypress = CANCEL THEN
    newvalue$=value$

    ELSE
        FLUSH_RX_BUFFER
        TX_LINE
        INPUT#2,newvalue$

        newvalue$=MID$(newvalue$,col,end_col)
        IF min_val <> 0 OR max_val <> 0 THEN
            IF VAL(newvalue$) < min_val OR VAL(newvalue$) > max_val THEN
                newvalue$ = value$
                FDISP row,col,VAL(value$)
                BELL
            END IF
        END IF
    END IF
END IF

RETURN

GetKeyPress:
    FLUSH_RX_BUFFER
    DO
        KEYPRESS=INCHAR(2)
    LOOP WHILE KEYPRESS=0
    RETURN
```

## APPENDIX C

### IWS Include File

IWS.INC - Include file for use with IWS30 and IWS120 series Operator Interface Panels

```
*****
'NOTE1:      The IWS panel must have the terminal firmware installed for the functions
'            in this include file to work properly.

'NOTE2:      To use this include file in your project, use the include '#INCLUDE' directive followed
'            by a complete path and file name. The include file directive must appear in the task before
'            any references to the defined functions can be used. example 'C:\MX2000\IWS.INC'

'NOTE3:      The semicolon (;) is used extensively through out this file. The semicolon is used to supress
'            the carriage return and line feed that is automatically appended to a PRINT command.
'            The semicolon is also used to structure the various codes sent to the IWS operator panel.
'

*****
'The letters ESC are only a representation of the ASCII escape code.
'ESC - Represents the ASCII 27 character

#DEFINE ESC CHR$(27)      'Escape code
#DEFINE FALSE 0
#DEFINE TRUE 1

*****
'The MX2000 has two comports for serial communication with external devices.
'Port #1 is the host port and used for program development. Port #2 is the aux.
'port is typically used with an operator interface panel.

'COM - Select which of the two comports are used in the following defines.

#DEFINE COM 2            'Aux. serial port
#DEFINE COM 1            'Host serial port

*****
'FLUSH_RX_BUFFER      Read and dump any characters that might be in the receive buffer.

#DEFINE FLUSH_RX_BUFFER DO:KEYPRESS=INCHAR(COM);LOOP UNTIL KEYPRESS=0

*****
'The cursor indicates where the next displayed character appears. The
'cursor is normally a flashing block, but it can instead be a flashing
'underline or invisible. If the cursor is disabled, it may be changed
'between a block cursor and underline cursor. To display the cursor
APPENDIX C - continued

'the cursor must be enabled.

'CURSOR_ON            Make cursor visible.
'CURSOR_OFF           Make cursor invisible.
'SOLID_CURSOR        Make cursor a solid block.
'LINE_CURSOR         Make cursor an underline.

#DEFINE CURSOR_ON      PRINT #COM,ESC+"[>51";
#DEFINE CURSOR_OFF     PRINT #COM,ESC+"[>5h";
#DEFINE SOLID_CURSOR   PRINT #COM,ESC+"[>4h"; 'does not apply to IWS127
```

**APPENDIX C** – continued

```

#DEFINE LINE_CURSOR      PRINT #COM,ESC+"[>4l"; 'does not apply to IWS127

*****

'The MX2000 can tell the IWS to save the current cursor position
'in order to restore it later. The IWS does not have the ability
'to stack multiple saves. The restore function will move the cursor
'to the last saved position.
'
'BELL                    Sound buzzer in the IWS operator interface panel
'TX_LINE                Transmit the current line (line cursor is on) from the IWS to the MX200.

#DEFINE SAVE_CURSOR      PRINT #COM,ESC+"[s";
#DEFINE RESTORE_CURSOR   PRINT #COM,ESC+"[u";
#DEFINE BELL             PRINT #COM,CHR$(7);
#DEFINE TX_LINE          PRINT #COM,CHR$(27);"[1p";

*****

'CLRSCRN                Home the cursor and clear the entire screen
'CLRLINE                Clear the current line
'LOCATE                 Move the cursor to @1(row),@2(col).
'                       LOCATE 2,7 - move the cursor to row 2 column 7.
'DISP                  Move the cursor to @1(row),@2(col)and display @3(value).
'                       DISP 1,5,"MX2000" - move the cursor to row 1 col 5 and display 'MX2000'.
'FDISP                 Move the cursor to @1(row),@2(col)and display @3(value).
'                       @3 is a numeric variable displayed using the format information in fmt$.
'                       DISP 1,5,x - move the cursor to row 1 col 5 and display the contents of x.

#DEFINE CLRSCRN         PRINT #COM,CHR$(12);
#DEFINE CLRLINE         PRINT #COM,ESC+"[2K";
#DEFINE LOCATE @1,@2    PRINT #COM,ESC+"["+STR$(@1)+";"+STR$(@2)+"H";
#DEFINE DISP @1,@2,@3  LOCATE @1,@2 @3;
#DEFINE FDISP @1,@2,@3 PRINT USING #COM,fmt$,ESC+"["+STR$(@1)+";"+STR$(@2)+"H";@3;

*****

'ECHO_ON               Whenever a key is pressed the IWS will display the key on the screen
APPENDIX C - continued

'                       and transmit the key's ASCII code to the MX2000(Half duplex).
'ECHO_OFF              Whenever a key is pressed the IWS will transmit the key's ASCII code
'                       to the MX2000, the key will not be displayed on the screen(Full duplex).

#DEFINE ECHO_ON         PRINT #COM,ESC+"[>14h";
#DEFINE ECHO_OFF        PRINT #COM,ESC+"[>14l";

*****

'FLASHING_ON           All subsequent characters will be displayed as flashing characters.
'FLASHING_OFF          All subsequent characters will be displayed normally.

#DEFINE FLASHING_ON     PRINT #COM,ESC+"[5m";
#DEFINE FLASHING_OFF    PRINT #COM,ESC+"[0m";

```

## APPENDIX C – continued

\*\*\*\*\*

'BEEP\_ON        A beep will sound each time a key is pressed.  
'BEEP\_OFF      Beep is disabled.

#DEFINE BEEP\_ON                                PRINT #COM,ESC+"[23h";  
#DEFINE BEEP\_OFF                               PRINT #COM,ESC+"[231";

\*\*\*\*\*

'KYBRD\_ON                                      Enable keyboard, keys will be processed.  
'KYBRD\_OFF                                     Disable keyboard, keys will not be processed.

#DEFINE KYBRD\_ON                               PRINT #COM,ESC+"[21";  
#DEFINE KYBRD\_OFF                               PRINT #COM,ESC+"[2h";

\*\*\*\*\*

'SET\_TIME        Set the clock in the IWS @1(hours hh),@2(minutes mm),@3(seconds ss)  
'SET\_DATE        Set the date in the IWS @1(month mm),@2(day dd),@3(year yy)  
'TIME\_ON         Enable display of time at @1(row),@2(col)  
'DATE\_ON         Enable display of date at @1(row),@2(col)  
'TIME\_OFF        Disable display of time  
'DATE\_OFF        Disable display of date

#DEFINE SET\_TIME @1,@2,@3                    PRINT #COM,ESC+"["+STR\$(@1)+";"+STR\$(@2)+";"+STR\$(@3)+"?s";  
#DEFINE SET\_DATE @1,@2,@3                    PRINT #COM,ESC+"["+STR\$(@1)+";"+STR\$(@2)+";"+STR\$(@3)+"?t";  
#DEFINE TIME\_ON @1,@2                        LOCATE @1,@2:PRINT#COM,ESC;"[?15h";  
#DEFINE DATE\_ON @1,@2                        LOCATE @1,@2:PRINT#COM,ESC;"[?14h";  
#DEFINE TIME\_OFF                               PRINT#COM,ESC;"[?15!";  
#DEFINE DATE\_OFF                               PRINT#COM,ESC;"[?14!";

## APPENDIX C - continued

\*\*\*\*\*

IWS120 series keyboard response codes

'  
#DEFINE F1 28  
#DEFINE F2 29  
#DEFINE F3 30  
#DEFINE F4 31  
#DEFINE ENTER 13  
#DEFINE CANCEL 24  
#DEFINE LEFT 8  
#DEFINE RIGHT 9  
#DEFINE DOWN 10  
#DEFINE UP 11  
#DEFINE BACKSPACE 127  
#DEFINE MINUS\_SIGN 45  
#DEFINE DECIMAL\_PT 46  
#DEFINE MENU 14  
#DEFINE HELP 15  
#DEFINE SETUP 25



## **APPENDIX D**

### **TECHNICAL SUPPORT OPTIONS**

#### **MOTION CONTROL APPLICATIONS ENGINEERING & SUPPORT**

PHONE: (800) 787-3532  
FAX: (800) 766-6366  
HOURS Monday - Friday from 8:00 AM to 5:00 PM Eastern Time

#### **WORLD WIDE WEB**

<http://www.superiorelectric.com>

#### **ON-SITE START-UP ASSISTANCE**

On-site start-up assistance is available. For scheduling and cost information, please contact your local Superior Electric Factory Representative or contact the Applications Engineering Department at the phone number listed above.

**MX2000**

**QUICK START GUIDE**

**SAMPLE PROGRAMS**

**DISK**

## **Distribution Coast-To-Coast and International**

Superior SLO-SYN products are available worldwide through an extensive authorized distributor network. These distributors offer literature, technical assistance and a wide range of models off the shelf for fastest possible delivery and service.

In addition, Superior Electric sales engineers are conveniently located to provide prompt attention to customers' needs. Call the nearest office listed for ordering and application information or for the address of the closest authorized distributor.

### **In U.S.A. and Canada**

383 Middle Street  
Bristol, CT 06010  
Tel: 860-585-4500  
Fax: 860-589-2136  
Customer Service: 1-800-787-3532  
Product Application: 1-800-787-3532  
Product Literature Request: 1-800-787-3532  
Fax: 1-800-766-6366  
Web Site: [www.superiorelectric.com](http://www.superiorelectric.com)

### **In Europe**

Warner Electric (Int.) Inc.  
La Pierreire  
CH-1029 Villars-Ste-Croix, Switzerland  
Tel: 41 021 631 33 55  
Fax: 41 021 636 07 04

---

[www.superiorelectric.com](http://www.superiorelectric.com)



MIDDLE STREET • BRISTOL, CT 06010  
(860) 585-4500 • FAX: (860) 589-2136

