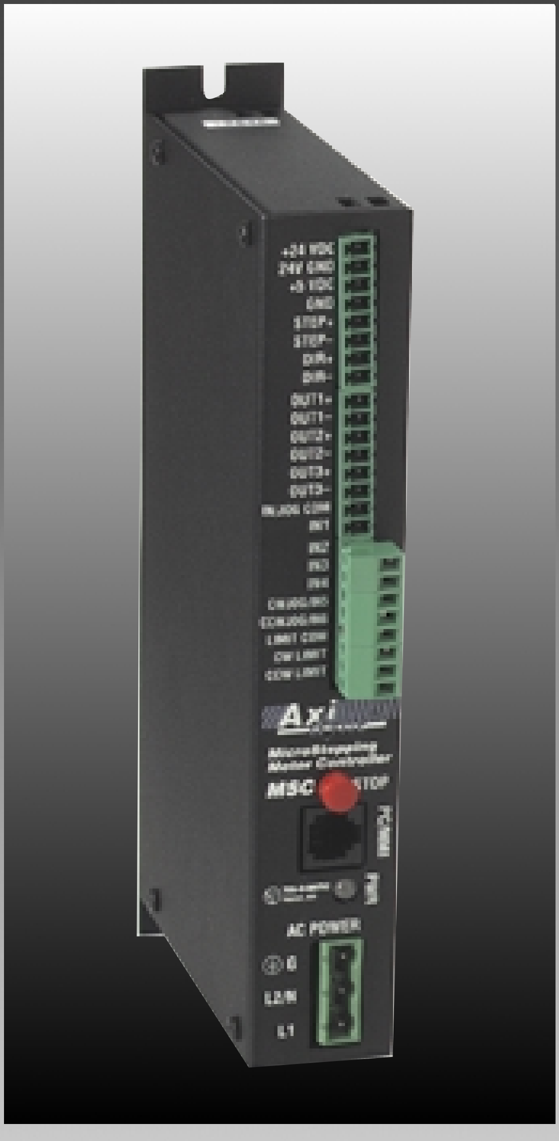
Axidyne®

ELECTRIC LINEAR MOTION PRODUCTS

MSC Microstepping Controller

User Manual

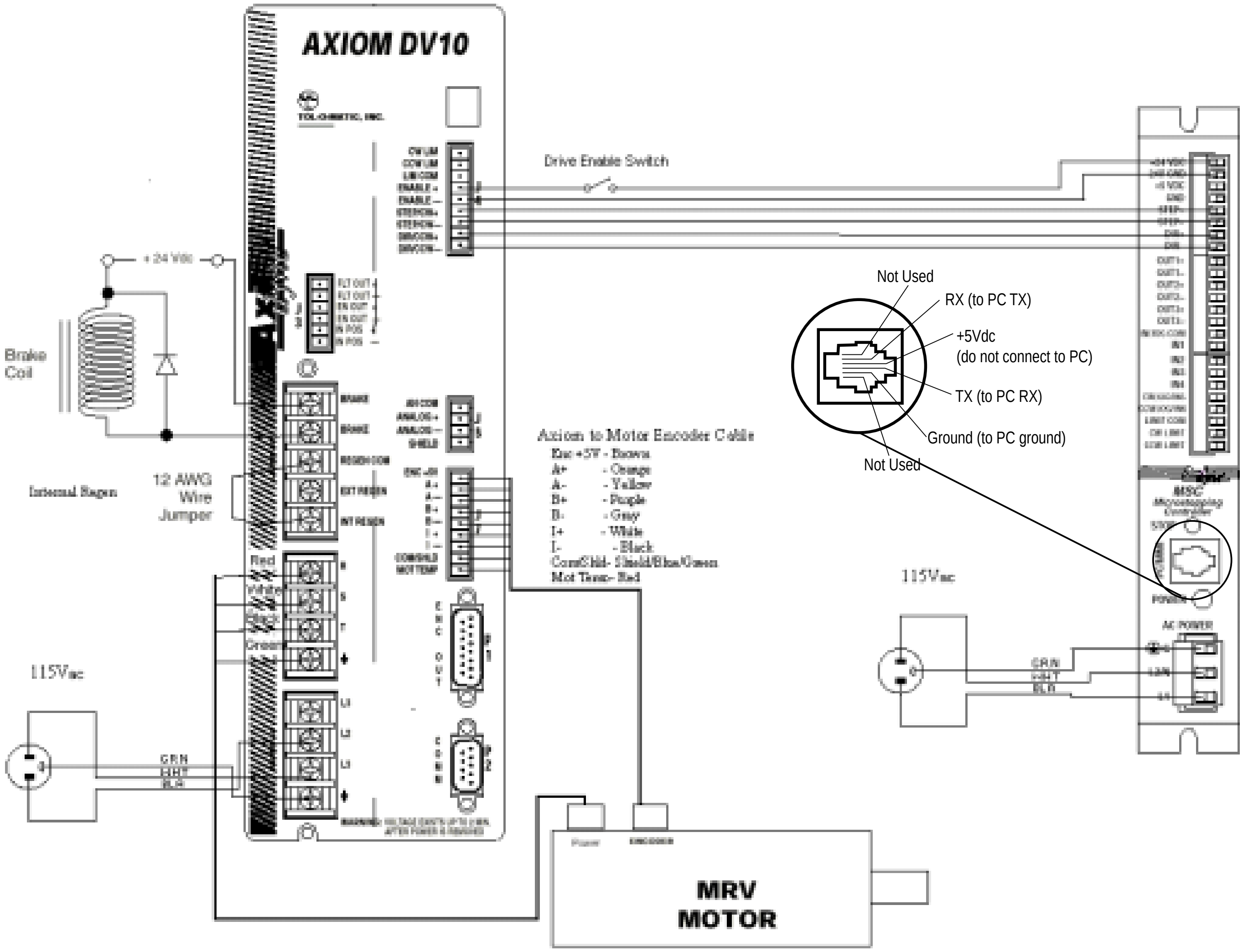


TOL-O-MATIC, INC

Excellence in Motion®

3600-4607B

12/00


AXIOM DV10
TOL-O-MATIC, INC.
CW LIM
CCW LIM
LIM COM
ENABLE +
ENABLE –
STEP/CW+
STEP/CW–
DIR/CCW+
DIR/CCW–
Drive Enable Switch
+ 24 Vdc
FLT OUT +
FLT OUT –
EN OUT +
EN OUT –
IN POS +
IN POS –
Brake Coil
BRAKE
BRAKE
REGEN COM
EXT REGEN
INT REGEN
Internal Regen
12 AWG Wire Jumper
AN COM
ANALOG +
ANALOG –
SHIELD
Axiom to Motor Encoder Cable
Enc +5V - Brown
A+ - Orange
A- - Yellow
B+ - Purple
B- - Gray
I+ - White
I- - Black
ComShld- Shield/Blue/Green
Mot Temp- Red
ENC +5V
A +
A –
B +
B –
I +
I –
COM/SHLD
MOT TEMP
Red
White
Black
Green
R
S
T
L1
L2
L3
ENC OUT
COMM
115Vac
GRN
WHT
BLK
WARNING: VOLTAGE EXISTS UP TO 2 MIN. AFTER POWER IS REMOVED
Power
ENCODER
MRV MOTOR
Not Used
RX (to PC TX)
+5Vdc
(do not connect to PC)
TX (to PC RX)
Ground (to PC ground)
Not Used
+24 VDC
24V GND
+5 VDC
GND
STEP+
STEP-
DIR+
DIR-
OUT1+
OUT1-
OUT2+
OUT2-
OUT3+
OUT3-
IN1
IN2
IN3
IN4
MSC
Microstepping Controller
STOP
POWER
AC POWER
LINE
115Vac
GRN
WHT
BLK

Contents

Chapter 1: Introduction

Chapter 2: Connections

Chapter 3: Wiring Inputs

Chapter 4: Wiring Outputs

Chapter 5: Stop Button

Chapter 6: Tol-O-Motion MSC Programming Software

Chapter 7: PIT Interface

Chapter 8: Programming Tools

Chapter 9: Programming Instructions

Introduction 1

About This Manual

This manual provides the information necessary to configure and install the Tol-O-Matic MSC Micro Stepping Controller for use with Tol-O-Matic AXIOM™ brushless servo drives or other drives accepting pulse and direction signals.

If difficulty is encountered while configuring, installing, or programming the system, please contact the nearest distributor for help, or call Tol-O-Matic at 1-800-328-2174.

This manual includes software features available with MSC units 3600-0040 and software 1.50.0x. Programs developed with prior software can be run with 1.50.0x.

SAFETY SYMBOLS

The following symbols are used throughout this manual to alert the user to potential safety hazards.

⚠ **Caution!** When this symbol appears, exercise care to avoid the possibility of sustaining slight operator injury or equipment damage.

☠ ***WARNING! When this symbol appears, exercise extreme caution to avoid an IMMEDIATE DANGER of sustaining severe operator injury or irreparable equipment damage.***

NOTE: Failure to comply with cautions and requirements in this manual, may result in damage to equipment not covered under Tol-O-Matic warranties.

CW, CCW DEFINITION

For all references in this manual, clockwise or counterclockwise rotation of the motor shaft is as viewed looking at the motor mounting face.

If the motor is direct-coupled to a right-hand screw actuator, *CCW* rotation will move the carrier *toward* the motor.

Features

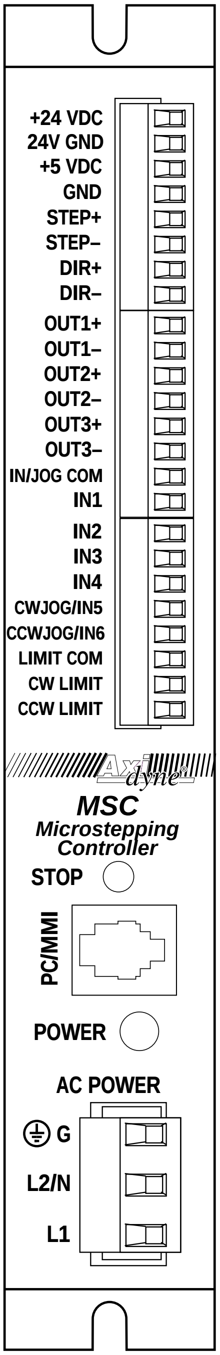


Figure 1

The MSC microstepping controller converts speed and position demands from the operator inputs into pulse rate and pulse count outputs which, when fed into the drive and motor, result in the required motion. The MSC is most appropriate for controlling Tol-O-Matic AXIOM™ brushless servo drives and motors.

The MSC features include:

- ***Powerful, flexible, easy to use indexer.***
- ***Programmable microstepping to 50,800 steps/rev revolution.***
- ***Separately programmable acceleration and deceleration.***
- ***Automatic homing function.***
- ***Connects by a simple cable to a PC for programming (cable included).***
- ***Tol-O-Motion MS Microsoft Windows®-based software for set up and programming.***
- ***Eight optically isolated inputs for interacting with the user and other equipment.***
- ***Three optically isolated outputs for coordinating external equipment.***
- ***Accepts 110 or 220 Vac power (switch selectable)***
- ***Rugged, compact 1.25 x 3.86 x 8 inch metal case with mounting brackets.***
- ***Pluggable screw terminal connectors for I/O, motor and ac power (all mating connectors included).***
- ***Bi-color red/green LED indicates power and indexer status.***
- ***Optional Panel-mount Interface (PIT) allows operator to enter distances, speeds, loop counts and more.***

Connections 2

Connections and Adjustments

Refer to Figure 2 for important connection and adjustment points. All mating connectors are included.

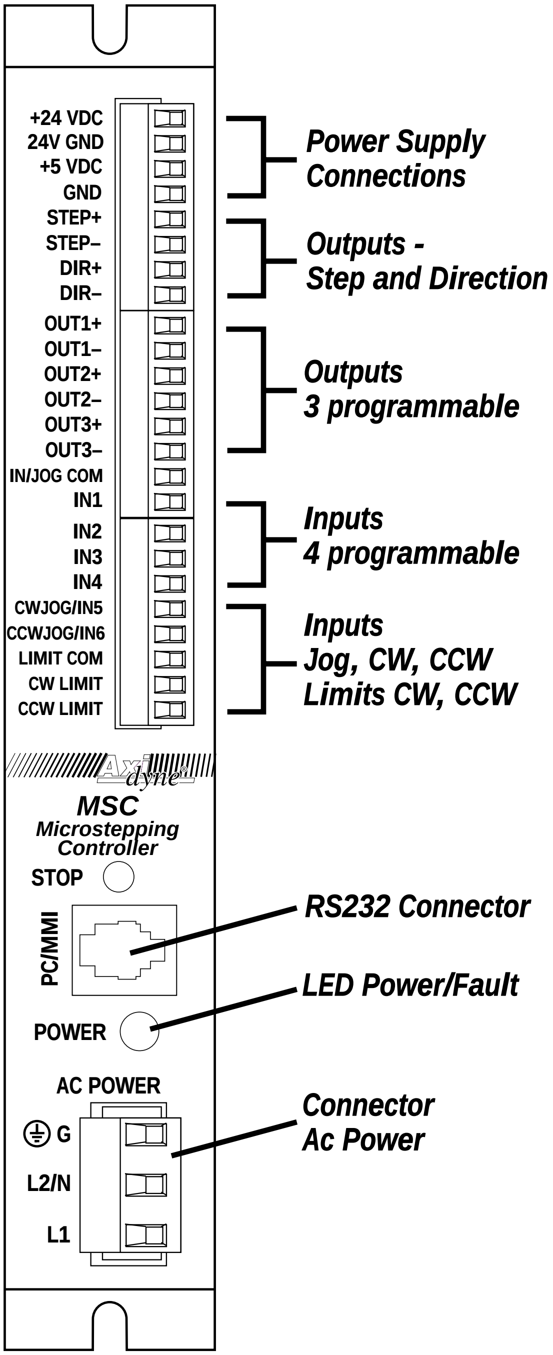


Figure 2 - CONNECTOR AND ADJUSTMENT POINTS

Connecting the AC Line

Caution! If the MSC is to operate on 220 volts, the setting of the 110/220V switch must be changed before applying power to the unit. The MSC is factory preset for 110 volts.

110 Volts

The MSC is set for 110 volt operation at the factory. Install a power cord and plug it in. Before direct wiring the MSC to ac power, consult a qualified electrician and observe all building and electrical codes.

WARNING! Ac power can be dangerous. Use extreme caution when installing the Ac cord.

220 Volts

The MSC is set for 110 volt operation at the factory. In order to use 220 volts, a switch setting must be changed inside the case.

Before changing the 110/220 volt switch setting, remove all connections from the MSC front panel, then remove the four screws from the cover as shown in Figure 3.

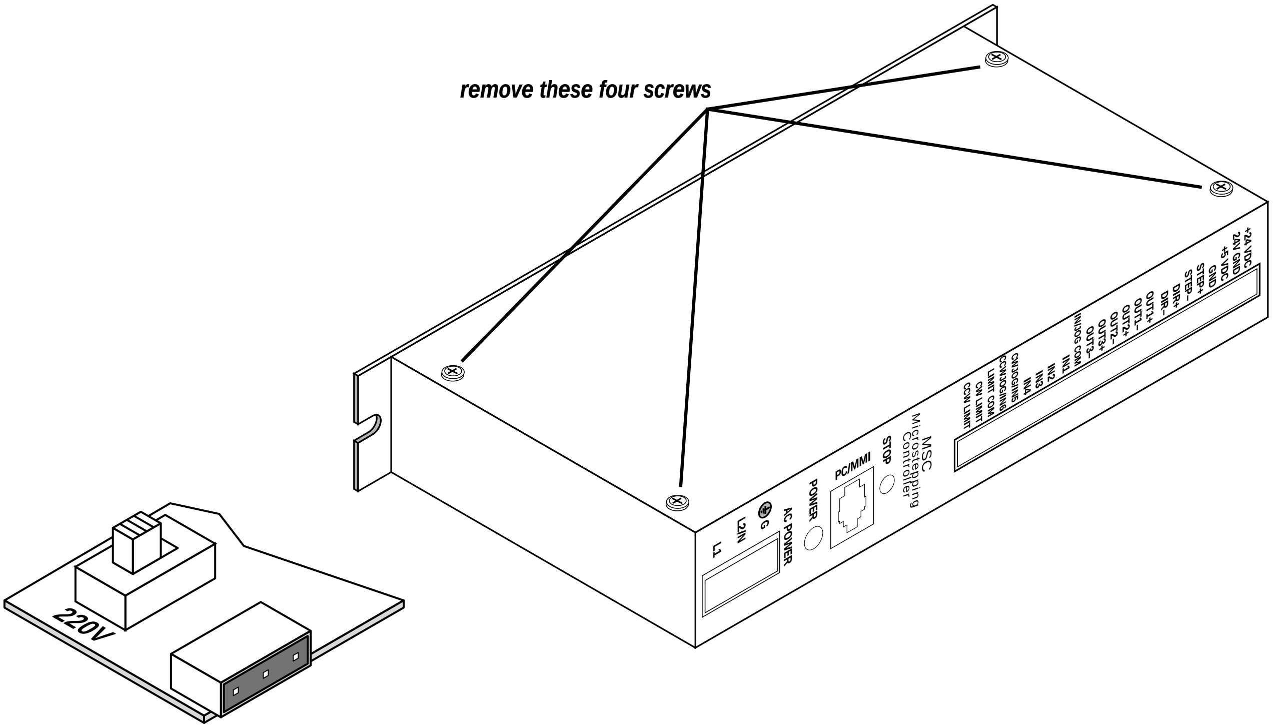


Figure 3 - 220V SWITCH SETTINGS

INSTALLING AN AC LINE CORD

WARNING! Always disconnect the line cord from the source before attaching it to the MSC.

Remove approximately 5 mm ($^3/_{16}$ inches) of insulation from each of the three wires of the line cord. Depending on its source, the power cord may have black, white and green wires or brown/blue/green.

Connections

Refer to Figure 4.

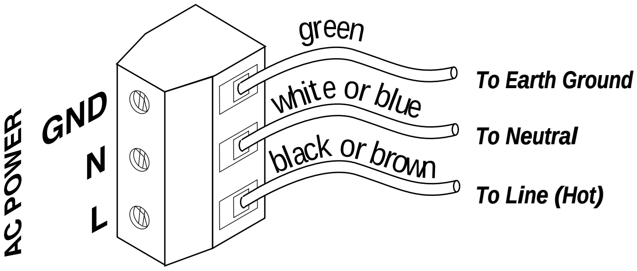


Figure 4 - LINE CORD INSTALLATION

1. Connect the black or brown wire to the MSC “L” (line or “hot” terminal of the ac power connector.
2. Connect the white or blue wire to neutral (the “N” terminal).
3. Finally, connect the green wire to the GND(ground) terminal.

Connecting to the Drive

The MSC can source step and direction inputs to most step motor drives. Some servo motor drives also accept step and direction inputs. The MSC is particularly appropriate for use with Tol-O-Matic brushless servo motor drives and motors.

Different step motor drives use different input configurations. There are three basic types that the MSC can be used with:

- Differential. Drive has STEP+, STEP-, DIR+ and DIR- inputs. Many high speed microstep drives use differential inputs.
- Common anode. Drive has STEP and DIR inputs that require sinking signals, and a “+5V” terminal.
- Common cathode. Drive inputs are STEP & DIR (requiring sourcing signals) and a “common” or “ground” terminal. This is the configuration of the Tol-O-Matic SSD drive.

Figure 5 shows the output circuits for the MSC.

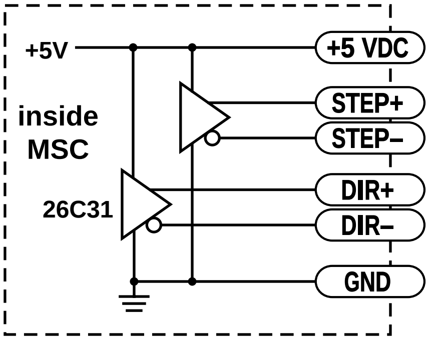
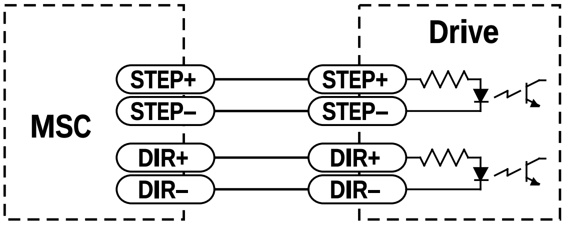


Figure 5 - MSC OUTPUT CIRCUITS

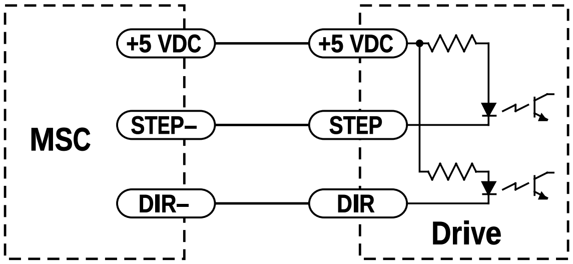
Wiring diagrams for each type of drive are shown in Figure 6 on page 8.

Figure 6 - DRIVER WIRING DIAGRAMS

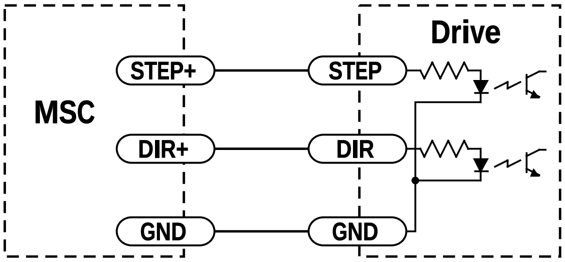
CONNECTING TO A DRIVE WITH DIFFERENTIAL INPUTS



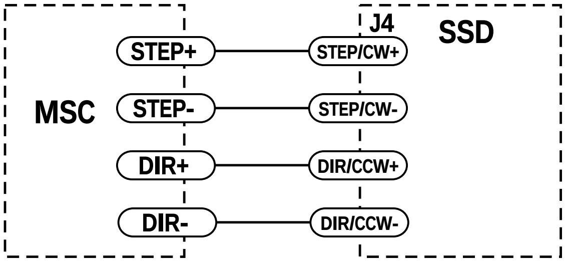
CONNECTING TO A DRIVE WITH COMMON ANODE (SOURCING) INPUTS



CONNECTING TO A DRIVE WITH COMMON CATHODE (SINKING) INPUTS



CONNECTING TO THE TOL-O-MATIC AXIOM DRIVE



see the Axiom manual for details

Computer Connection

To Connect to the Computer for Programming:

1. Locate the MSC within 6 feet of the computer.

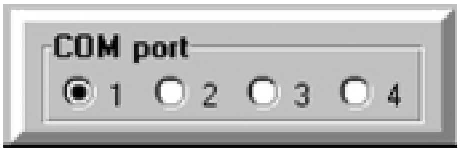
2. The MSC was shipped with a black adapter plug. It has a telephone-style jack at one end and a larger 9-pin connector at the other. Plug the large end into the computer's (COM1) serial port. Secure the adapter with the screws on the sides. If the COM1 port is already allocated, use of the COM2 port will be suggested. On some computers, COM2 will have a 25-pin connector that does not fit the black adapter plug. If this is the case and COM2 must be used, a 25-to-9 pin serial adapter will be required.

3. The MSC was also shipped with a 7-foot telephone line cord. Plug one end into the adapter just attached to the computer, and the other end into the RS232 jack on the MSC.

Caution! Never connect the MSC to a telephone circuit. The MSC uses the same connectors and cords as telephones and modems, but the voltages are not compatible.

4. It may be necessary to set the COM port in the MSC Windows software. When the software is loaded, it looks for the first available COM port, but doesn't always find the one selected by the connection.

   The port can be selected by clicking on one of the "COM port" option buttons (see Figure 7). If the port exists and is not already in use, the programming software will use it to communicate with the MSC.



COM port
● 1 ○ 2 ○ 3 ○ 4

Figure 7 - COM PORT

Programming Note: *Always open and run the programming software on the computer before applying power to the MSC.*

Wiring Inputs 3

Overview

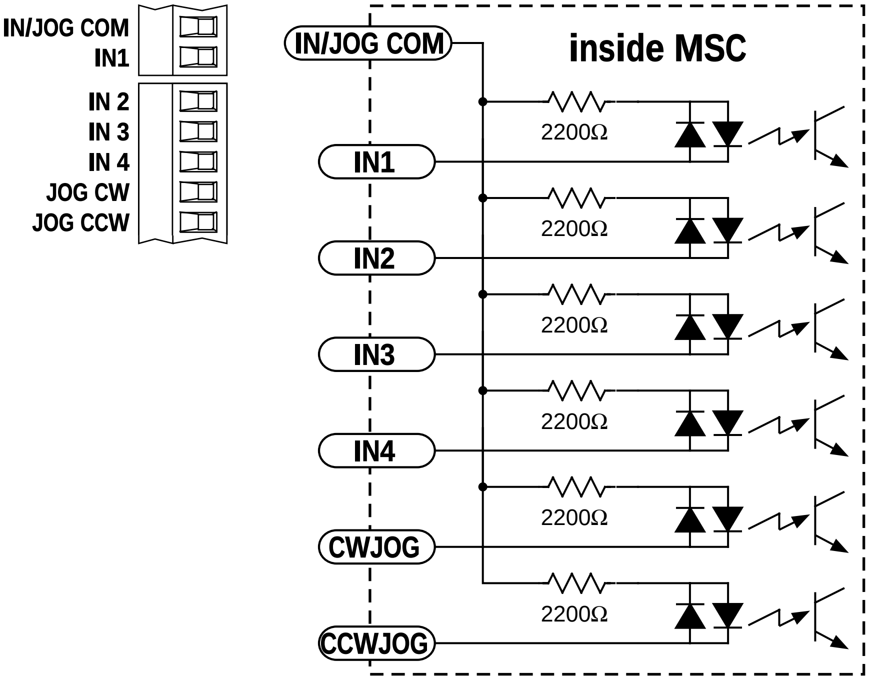


Figure 8 - INPUT CIRCUIT SCHEMATIC

Caution! The maximum voltage that can be applied to an input terminal is 24 Vdc.

The MSC input circuits can be used with sourcing or sinking signals, 5 to 24 Volts. This allows connection to TTL circuits, PLCs, relays and mechanical switches. Because the input circuits are isolated, they require a source of power. If connecting to a TTL circuit or to a PLC, power should be available from the PLC or TTL power supply. If using relays or mechanical switches, the MSC's built-in 24 Volt power supply can be used.

NOTE: If **current is flowing** into or out of an MSC input, the logic state of that **input is low**. If **no current is flowing**, or the input is not connected, the **logic state is high**.

The diagrams on the following pages show how to connect MSC inputs to various devices.

Connecting an Input to a Switch or Relay

When connecting an input to a switch or relay, use a ***normally open momentary contact switch*** to trigger the MSC using a *WAIT INPUT* instruction. A ***single throw switch*** may be used for parameter selection using an *IF INPUT* instruction. A normally open momentary switch can be used for jogging. This connection is appropriate for Tol-O-Matic Reed Switches.

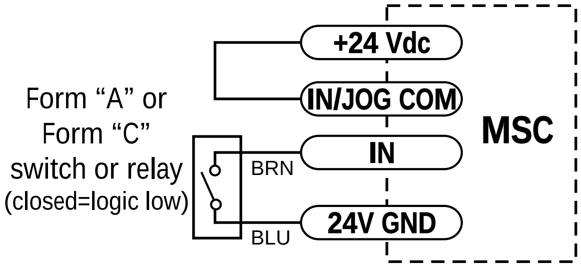


Figure 9 - SWITCH OR RELAY CONNECTION

Connecting an NPN-Type Proximity or Hall Effect Sensors to an Input

When the proximity sensor or Hall Effect Sensor activates (closes) the input will go active ("low"). This connection is appropriate for Tol-O-Matic Hall Effect Sinking Switches.

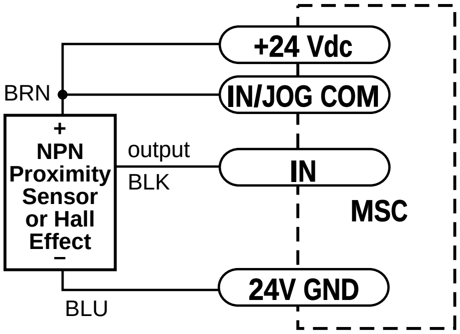


Figure 10 - NPN SINKING SENSOR CONNECTION

Connecting PNP-Type Proximity or Hall Effect Sensors to an Input

When the proximity sensor activates (closes), input will be "low" (active). This connection is appropriate for Tol-O-Matic Sourcing Switches.

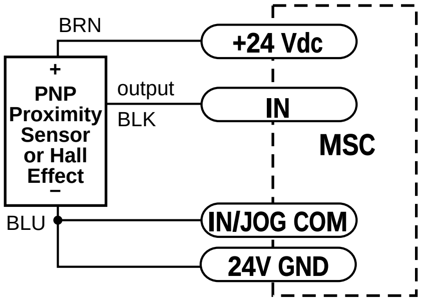


Figure 11 - PNP-TYPE PROXIMITY SENSOR CONNECTION

Jog Inputs

Two of the MSC input terminals are provided for jogging the motor. The inputs are labeled "JOG CW" and "JOG CCW". Taking one of the inputs low commands the drive to move the motor at a pre-designated speed until the contact is opened. A relay or mechanical switch can be used to activate the jog inputs. Five-volt circuitry can also be used. The schematic diagram of the input circuit is shown in Figure 12.

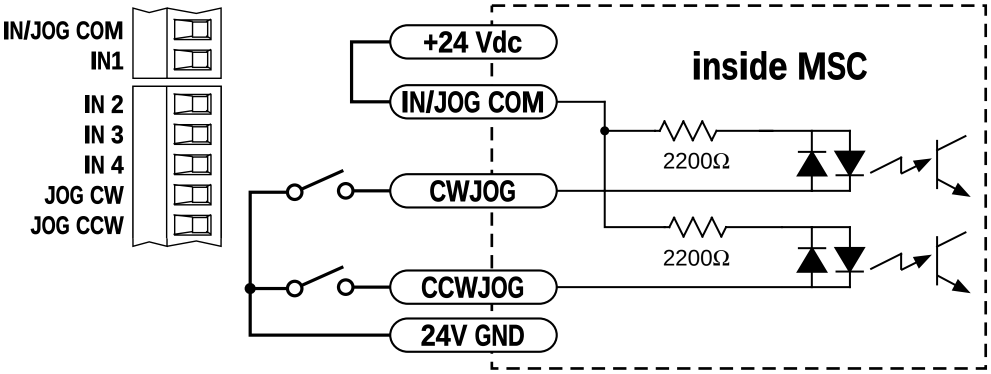


Figure 12 - JOG INPUT CIRCUIT SCHEMATIC (for use with sinking inputs)

If a switch or relay is to be used, wire one end to the *JOG* input and the other to the *24V GND* terminal. Then connect the *IN/JOG COM* and *+24 Vdc* terminals. (If the 24 Vdc is connected for sourcing inputs, i.e. *IN/JOG COM* to *24V GND*, the jog switches will connect from *JOG CW, CCW* to the *+24 Vdc*.)

JOG CCW will drive the carrier toward the motor for a right hand screw actuator with a direct coupled motor.

When the MSC is connected to a computer with the programming software running, the jog inputs will function under two conditions:

1. If the program is not executing, or
2. If the program is executing a *WAIT* for *INPUT* or a *PIT WAIT* for *ENTER* instruction.

If the MSC is operating in stand alone mode (i.e. without a computer attached) then the jog inputs work when the program is executing the Wait Input instruction, or when waiting for a *PIT* (Interface Terminal) *ENTER* input. All other program execution modes inhibit the jog inputs.

Jogging can also be controlled from the arrow keys on the PIT while waiting for a PIT "*ENTER*".

Limit Switches

The MSC has two limit switch inputs, *LIMIT CW* and *LIMIT CCW*. By connecting switches or sensors that are triggered by the motion of the motor or load, the operating range of the MSC is limited. This is useful if a program error could cause damage to the system by exceeding the mechanical operating range.

The limit inputs are optically isolated. This allows a choice of voltage for the limit circuits of 5 to 24 Vdc. It also allows the use of long wires on limit sensors that may be far from the MSC with less risk of introducing noise to the MSC. The schematic diagram of the limit input circuit is shown in Figure 13.

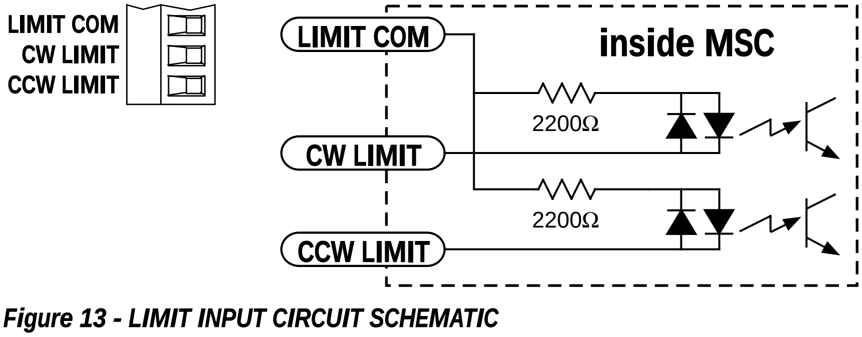


Figure 13 - LIMIT INPUT CIRCUIT SCHEMATIC

WIRING A LIMIT SWITCH

Normally open or normally closed limit switches may be used. Either is wired as shown in Figure 14.

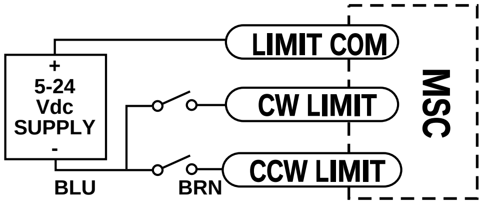


Figure 14 - LIMIT SWITCH WIRING DIAGRAM

Limit Sensors

Some systems use active limit sensors that produce a voltage output rather than a switch or relay closure. These devices must be wired differently from switches.

If the sensor has an ***open collector*** output or a ***sinking*** output, wire it as shown in Figure 16.

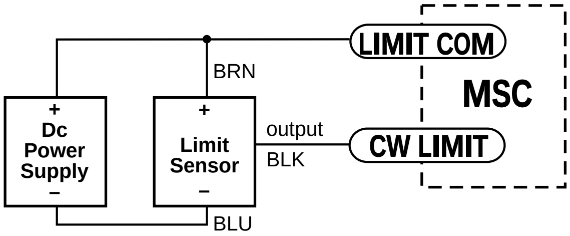


Figure 16 - WIRING FOR OPEN COLLECTOR OR SINKING OUTPUT FROM SENSOR

If the sensor output goes low (current flows) at the limit, select the option “Closed.” If the output is open, (or high voltage), choose “Open.”

Other sensors have ***sourcing*** outputs. That means that current can flow out of the sensor output, but not into it. In that case, wire the sensor as shown in Figure 17.

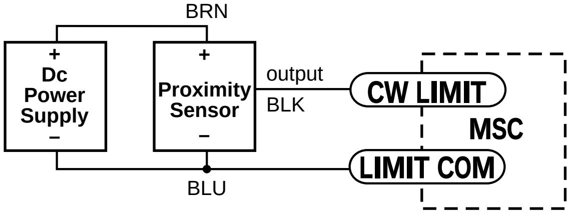
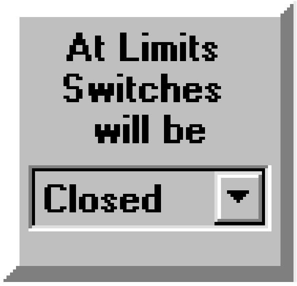


Figure 17 - WIRING FOR SOURCING OUTPUT FROM SENSOR

If the sensor output goes high (current flows) at the limit, choose the program option "Closed." If the output is low (open) at the limit, select "Open."

Programming for Normally Open or Normally Closed

The main window of the MSC programming software (see page 27) contains a panel for selecting the type of limit switches or sensors being used (see Figure 15).



At Limits Switches will be

Closed

Figure 15 - LIMIT SWITCH SELECTION PANEL

If the limit switch closes (current flows) when reached, select the option button marked "Closed." This is often referred to as a *normally open* switch. If the limit switches are closed when the motor is not at a limit, and will open when reached, select the option button "Open." This type of switch is frequently called *normally closed*.

MSC Operation When a Limit Switch is Activated

If a limit is reached during a *FEED TO LENGTH*, *FEED TO SENSOR*, *FEED & RETURN* or *FEED TO SENSOR & RETURN* move, the MSC will immediately stop the motor (with no deceleration). The red *POWER* LED on the front panels will flash, and no future motion is possible. Ac power must be removed from the drive to reset this condition. If the drive is connected to a computer, the programming software will alert the user to this condition. Reset can be from the computer (instead of removing ac power).

If a limit is reached while jogging (using the *JOG CW* or *JOG CCW* inputs), motion will be disabled in the direction of travel. Jog can be activated in the reverse direction to back away from the limit.

During a *SEEK HOME* instruction, the motor will reverse direction when a limit is encountered, and continue seeking the home sensor.

Notes:

Wiring Outputs 4

Overview

All three MSC programmable outputs are optically isolated, meaning there is no electrical connection between the indexer-drive and the output terminals. The signal is transmitted to the output current as light. This signal drives a transistor (NPN type) that closes, or conducts current, when the output is low. When the output is high, the transistor is open.

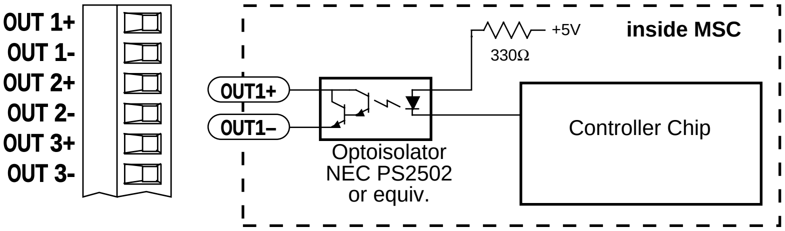


Figure 18 - OUTPUT CIRCUIT SCHEMATIC

Caution! The maximum permissible voltage between any pair of + and - output terminals is 24 Vdc. Never connect ac voltages to the MSC output terminals. Maximum current is 100 mA per output.

Programmable Outputs

IMPORTANT NOTE: At power-up, the MSC sets all three programmable outputs high (open circuit).

Since there is no electrical connection to the MSC, current and voltage must be sourced from a PLC (programmable logic controller) or from a separate power supply. The current must be limited to less than 100 mA so that the output transistor is not damaged. A resistor is normally used for this, but some loads (such as PLC inputs) limit the current automatically.

Figure 19 shows how to connect an MSC output to an optically-isolated PLC input.

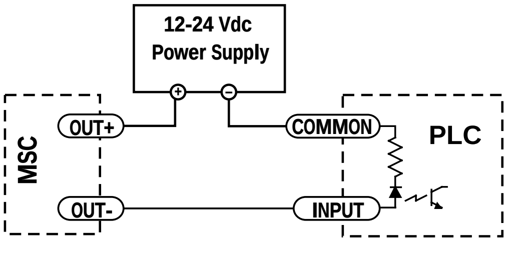


Figure 19 - CONNECTING AN MSC OUTPUT TO AN ISOLATED PLC INPUT

Notes:

STOP Button 5

Front Panel STOP Button

There is a button marked “STOP” on the front panel of the MSC. This button can be used to interrupt motion at any time. After pressing the *STOP* button, the output pulses to the motor drive will stop.

The front panel *POWER* LED will then flash until the MSC is reset by removing and restoring the power. If the MSC is connected to a computer running the *Tol-O-Matic MSC Software*, the software will alert the user on screen to the condition, and provide the option to reset the drive from the computer.

Notes:

Tol-O-Motion MSC Programming Software 6

Overview

The MSC is designed to be both configured and programmed through software. There is only the stop button on the front panel and there are no jumpers inside. Virtually all the MSC's functions are controlled by the software. The MSC programming software that comes with the drive provides for setting the step resolution, jogging parameters and limit switch polarity. It also allows the writing of complex motion control and machine interaction programs.

Installing the Programming Software

The MSC comes with two 3.5" software diskettes. This software requires a computer configured as follows:

- IBM compatible 386, 486, or higher CPU
- Microsoft Windows 3.1, Windows 95, or Windows NT
- At least 8 MB memory (16 MB will make the software run much faster)
- 4 MB available hard drive space
- VGA monitor, or better
- Mouse or other input device
- 3.5" floppy disk drive
- A 9-pin serial port

Like most Windows-based programs, installing the MSC software is highly automated.

To Install the Software:

1. Put Disk 1 in the 3.5" drive.
2. From the Windows Program Manager, select *RUN* from the *FILE* menu. (In Windows 95, choose *RUN* from the *START* menu.)
3. If the 3.5" drive is drive A, enter the command line “A:\SETUP”. If the 3.5" drive is B, type “B:\SETUP.”
4. The *SETUP* program will provide screen prompts needed to finish the installation.

If errors are encountered during installation, it is usually due to lack of memory or conflicts with other programs already running on the computer. If an error occurs while installing the programming software, quit all other Windows applications and try again. Holding down the *ALT* key and repeatedly pressing *TAB* will show all the programs currently running on the computer.

NOTE: Laptop computers generally present the biggest challenge to installation as they often come preloaded with programs that automatically execute on startup such as Microsoft Office and battery managers. Furthermore, laptops usually have the least memory.

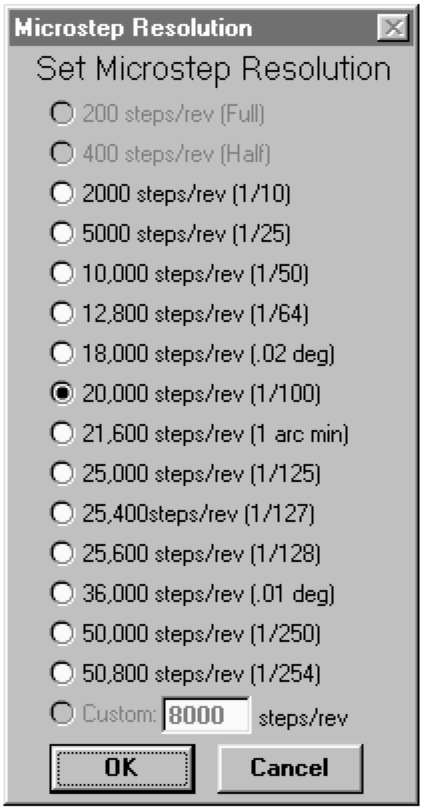
The programming software will install more easily and run much faster with more memory. Eight megabytes of RAM on a Windows 3.1 system is recommended (16 megabytes with Windows 95).

Upon successful installation of the software, an icon is created in Windows. Double-click this icon to initiate the software.

Programming Note: Always apply power to the MSC ***after*** the Programming software is running on the computer.

Using the Software to Input System Settings

SETTING THE MICROSTEPPING RESOLUTION



Microstep Resolution

Set Microstep Resolution

- ○ 200 steps/rev (Full)
- ○ 400 steps/rev (Half)
- ○ 2000 steps/rev (1/10)
- ○ 5000 steps/rev (1/25)
- ○ 10,000 steps/rev (1/50)
- ○ 12,800 steps/rev (1/64)
- ○ 18,000 steps/rev (.02 deg)
- ● 20,000 steps/rev (1/100)
- ○ 21,600 steps/rev (1 arc min)
- ○ 25,000 steps/rev (1/125)
- ○ 25,400steps/rev (1/127)
- ○ 25,600 steps/rev (1/128)
- ○ 36,000 steps/rev (.01 deg)
- ○ 50,000 steps/rev (1/250)
- ○ 50,800 steps/rev (1/254)
- ○ Custom: 8000 steps/rev

Figure 20 - CHANGE RESOLUTION DIALOG BOX

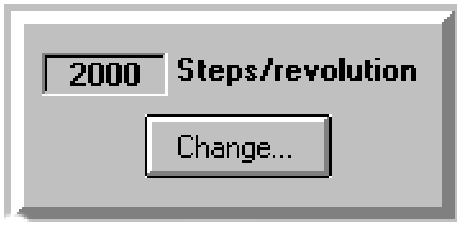
Most non-microstep step motor drives offer a choice between full-step and half-step resolutions. In full-step mode, both motor phases are used all the time. Half-stepping divides each step into two smaller steps by alternating between both phases ON and one phase ON.

Microstepping drives electronically subdivide the steps even further. The MSC software offers a choice of 15 step resolutions, starting at 200 steps per revolution or a custom resolution can be chosen (see Figure 20). The highest setting divides each full step into 254 microsteps, providing 50,800 steps per revolution. The MSC can be used with stepper drives and with servo drives that accept pulse and direction signals such as the Tol-O-Matic Axiom servo series, (brush or brushless).

Tol-O-Matic MRB motors use a 1000-line encoder and require the MSC to be set for 4,000 steps per revolution.

The MRV brushless servos use a 1000-line encoder with quadrature connection and therefore also require the MSC to be set for 4000 steps per revolution.

The microstep resolution of the MSC is set using the programming software. The resolution appears at all times in the *STEPS/ REVOLUTION PANEL* on the left side of the *MAIN WINDOW* (see Figures 21 and 25).



2000 Steps/revolution

Figure 21 - STEP/REVOLUTION PANEL

To change the resolution, click on the *CHANGE* button to access a dialog box with *OPTION* buttons for each resolution (see Figure 20). Click the resolution required, then click the *OK* button.

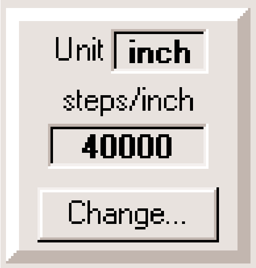
The selection of Microstep Resolution is important if either the optional PIT or the User Unit Selection is to be used to scale the distances or speeds entered by the user.

SETTING THE USER UNITS

The main programming window (see Figure 22) allows selection of User Units. This selection is used in conjunction with the Microstep Resolution selection of steps/revolution to set a scaling factor which allows all move requirements to be entered (whether in programming or via the PIT) in the users preferred units.

To make the selection click the cursor on the change box for User Units name (Figure 22) and enter a one to four digit description (mm, inch, feet, ????). Using the selected steps/resolution calculate the steps/unit (using screw pitch or belt wheel diameter and gear ratio) and enter in the user unit box.

The units for the Jog Parameters will be seen to change, as will the units for all other screens for move data entry.



Unit inch

steps/inch

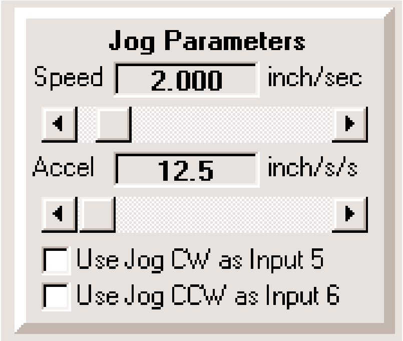
40000

Figure 22 - USER UNITS PANEL

NOTE: All max speeds will be subject to the 100 rev/sec maximum motor rotation speed limit set within the software. The steps/user unit must be an integral number. Decimals will be rounded up or down to the nearest integral. Using high values for the steps/revolution will reduce the error Also for using metric user units with inch lead screws it is helpful to use 25,400 steps/ revolution for the resolution.

SETTING THE JOG PARAMETERS

To set the *JOG SPEED* and *JOG ACCEL/DECEL RATE*, adjust the *SCROLL BARS* in the *MAIN PROGRAMMING WINDOW* (see Figure 23).



Jog Parameters

Speed 2.000 inch/sec

Accel 12.5 inch/s/s

☐ Use Jog CW as Input 5

☐ Use Jog CCW as Input 6

Figure 23 - SETTING JOG PARAMETERS

Modest accel/decel rates are appropriate for jogging. Twenty-five rev/s/s usually works well except for high inertial loads, in which case choose a lower rate. The range of jog accel is 1 to 3000 rev/s/s.

The range of jog speed is .025 to 50 rev/sec. The selection will depend on the application. If jogging is not required in the clockwise direction, the *CW JOG* input can be used as a general purpose input by checking the box marked “Use Jog CW as Input 5.” The same is true for the *JOG CCW* input. These selections can be implemented after system set-up.

Using the Tol-O-Motion Software for Programming Motion

OVERVIEW

The Tol-O-Motion MSC software has a user program capacity of 100 lines. In this space, multiple motion and machine control programs can be designed. Sixteen high-level commands, or *instructions,* are available for this purpose.

Seven of the instructions involve pure motion: *FEED TO LENGTH* and *FEED & RETURN* are fixed distance moves. *FEED TO SENSOR* and *FEED TO SENSOR & RETURN* move relative to a sensor that is wired to one of the inputs. *FEED TO POSITION* moves to an absolute position relative to a previously set absolute reference position. *FEED/SET OUTPUT* allows an output to be set at any point during a move. *SEEK HOME* searches for a home sensor, “bouncing off” the limits if necessary to find it.

SET ABS POSITION allows the present position to be defined as any absolute position, at any point in the program.

Two instructions handle timing. *WAIT TIME* causes the program to stop for a specified amount of time. *WAIT INPUT* waits for one of the inputs to reach a specified state before continuing the program.

Four instructions control program flow. *GO TO* causes the program to jump to a particular line. *IF INPUT* jumps to a line if one of the inputs meets a specified condition. Otherwise, the program goes on to the next line. *REPEAT* and *END REPEAT* set up a loop wherein the same instructions can be repeated many times.

The *SET OUTPUT* instruction can be used to signal other equipment that a particular place in the program has been reached.

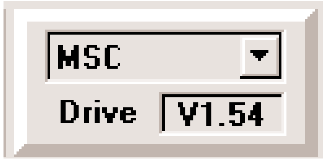
Using the *PIT PROMPT* instruction with the optional Panel-mount Interface (PIT), (or operator panel), the operator can enter distances, speeds and repeat loop counts on a keypad. The drive can also display messages for the operator, pause the program until the operator presses the *ENTER* button, or ask the user to make a decision and respond by pressing the *YES* key or *NO* key.

A *COMMENT* tool allows notes to be inserted in the program so that it is easier to understand.

By combining the 16 instructions in different ways, a nearly infinite variety of useful programs and motion profiles can be constructed. Planning the objectives and sequence prior to entering instructions eases the programming process.

There are actually two software programs associated with the MSC. The first is the Windows program installed on the computer from the floppy disks. After the program is loaded, click on the Tol-O-Matic logo to see the software version and Tol-O-Matic's phone, fax and website numbers.

A second software program resides in a chip inside the MSC. Software in a chip is usually called ***firmware.*** It is the MSC firmware that runs the drive and executes the program. The drive firmware version is displayed near the top of the screen when the MSC is connected to the computer and turned on (see Figure 24). Make sure that the MSC software is selected in the drop box.



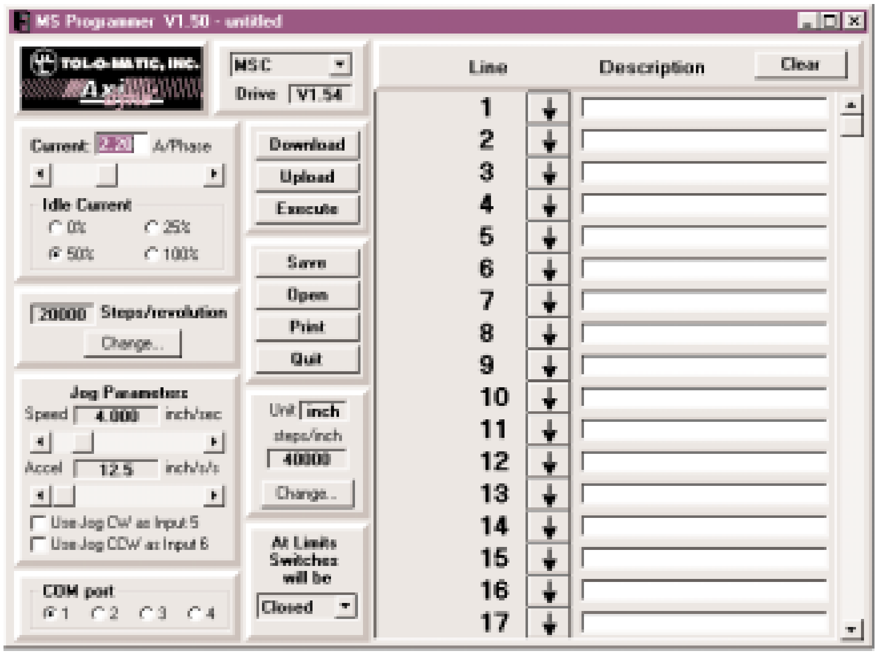
MSC

Drive V1.54

Figure 24 - FIRMWARE VERSION DISPLAY

Entering The Program

To activate the MSC software, go into Windows and locate the *PROGRAMMER* icon. Double-click on the icon to run the software. The *MAIN PROGRAMMING WINDOW* will soon appear, as shown in Figure 25. If COM 1 is used by the computer, the software will prompt for use of COM 2.



MS Programmer V1.50 - untitled

MSC
Drive V1.54

Current: [illegible] A/Phase
Idle Current: ○ 0% ○ 25% ● 50% ○ 100%

20000 Steps/revolution

Jog Parameters
Speed 4.000 inch/sec
Accel 12.5 inch/s/s
□ Use Jog CW as Input 5
□ Use Jog CCW as Input 6

COM port: ● 1 ○ 2 ○ 3 ○ 4

Unit inch
steps/inch 40000

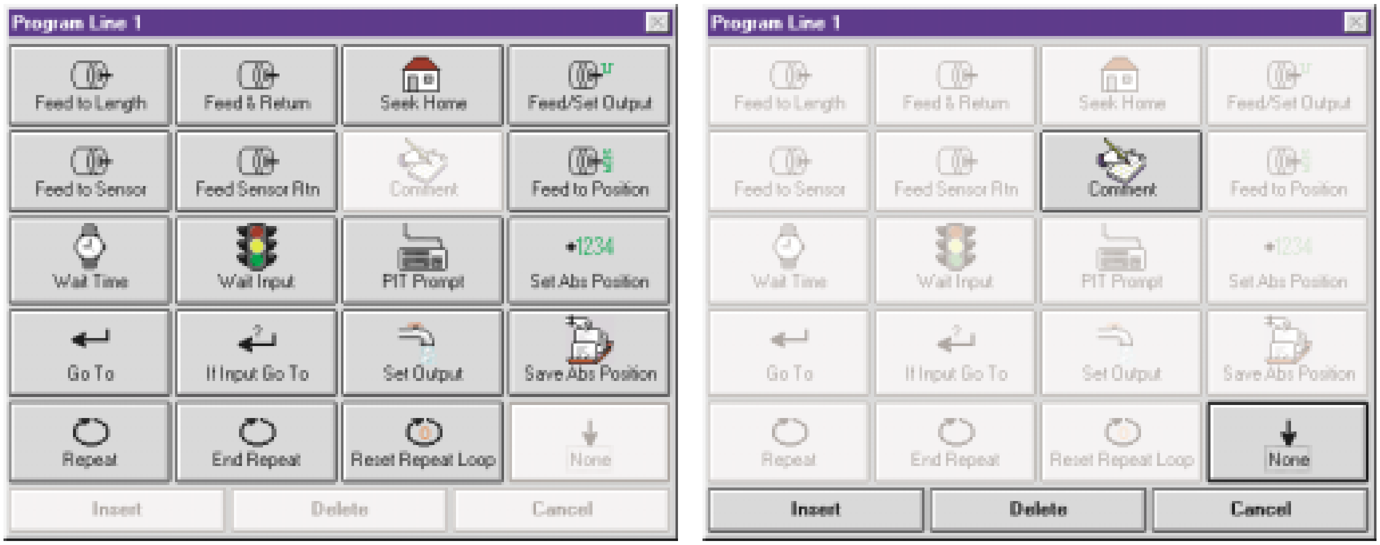
At Limits Switches will be Closed

Figure 25 - MAIN PROGRAMMING WINDOW

If an MSC is connected to the computer, turn it on now. After applying power, the computer should beep. The Version box will display the version number of the MSC firmware that is in the drive. Program parameter setup is described on page 6-14.

INSTRUCTIONS & TOOLS

Refer to Figure 26. The programming software provides 16 easy-to-use high-level instructions for building programs. In addition, there are five tools.



INSTRUCTION SET

TOOL SET

Figure 26 - INSTRUCTION/TOOL MENU

	NAME	PAGE REF.	NAME	PAGE REF.
Instructions	Feed to Length	9-3	Feed & Return	9-5
	Feed to Position	9-10	Feed/Set Output	9-8
	Seek Home	9-13	Feed to Sensor	9-6
	Feed to Sensor & Return	9-7	Set Abs Position	9-11
	Wait Input	9-15	Save Abs Position	9-12
	Go To	9-17	Wait Time	9-15
	Set Output	9-22	PIT Prompt	9-1
	End Repeat	9-18	If Input Go To	9-24
	Reset Repeat Loop	9-20	Repeat	9-18

	NAME	PAGE REF.	NAME	PAGE REF.
Tools	Comment	8-2	None	8-1
	Insert	8-1	Delete	8-1
	Cancel	8-2		

Most programs will begin with the instruction *WAIT INPUT* or a PIT prompt. This ensures that when power is applied to the MSC, it doesn't do anything until instructed to do so. It is easy to put a *WAIT INPUT* instruction on the first line.

Next to the large number 1 in the *PROGRAM WINDOW* is a button showing the *NONE* icon which indicates there is no instruction for that line. Whenever the MSC encounters a "None" program line, it simply moves on to the next line, as the icon implies with a downward pointing arrow. After the MSC executes the instruction on line 100, it automatically jumps to line 1, unless the instruction on line 100 makes it jump somewhere else.

To enter *PROGRAM LINE 1*, click once on the *PROGRAM* icon. The *PROGRAM LINE...* dialog box will appear (see Figure 27), displaying the programming instructions and tools.

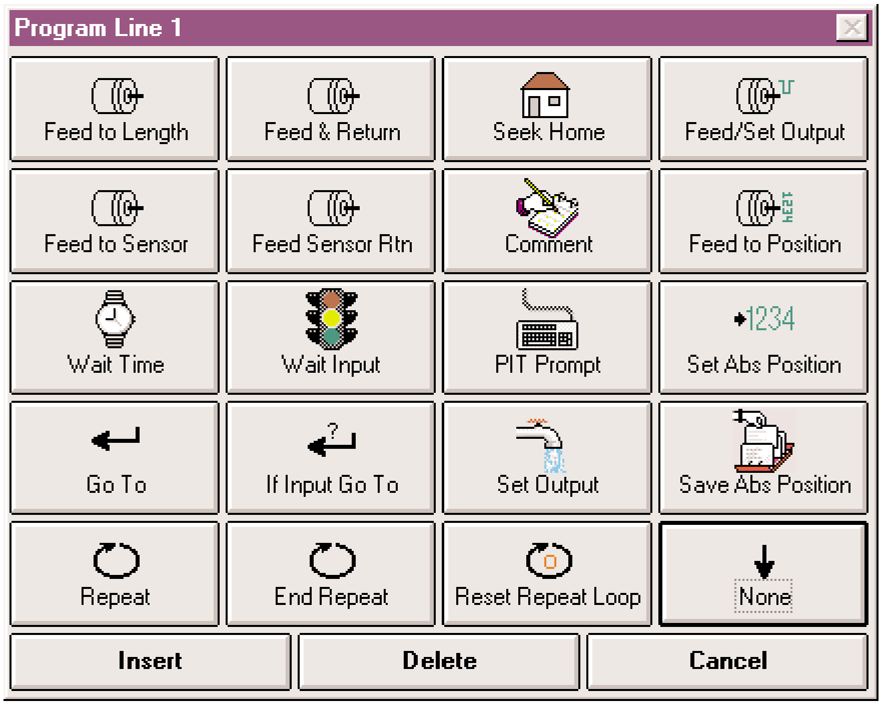


Figure 27 - "PROGRAM LINE" DIALOG BOX

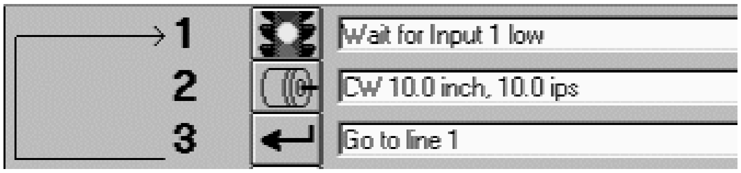
Click on the button marked "Wait Input". The *WAIT INPUT* dialog box will appear. Click on the option button marked "Low" (in the "Condition" group), then click *OK*.

The first line of the program should now display the *WAIT INPUT* icon, and the description “Wait for input 1 low.”

Click on the icon button for *PROGRAM LINE 2*. When the *PROGRAM LINE...* dialog box appears, click on *FEED TO LENGTH*. In the *FEED TO LENGTH* dialog box, enter the distance as “5 inch,” then slide the speed bar to “10 in/sec.” Click *OK*.

The second program line should now show the motor icon and the caption “CW 5 inch, 10 ips.”

Click on the *PROGRAM LINE 3* icon. Choose *GO TO*. When the *GO TO* dialog appears, the line number will already be set to “1.” Click *OK*. The program should now look like the program shown in Figure 31.



1 Wait for Input 1 low

2 CW 10.0 inch, 10.0 ips

3 Go to line 1

Figure 28

More complex programs are entered in the same manner, with more lines and more concern about the exact parameters and their importance in the application.

NOTE: To edit an instruction that is already in the program, hold down the shift key while clicking on the instruction icon. This is a shortcut to the instruction dialog box.

DOWNLOAD, UPLOAD & EXECUTE

The MSC is designed to operate without a host computer once the program is finished and tested. However, running it from the computer makes it easy to quickly make changes in the program to fix errors or conduct experiments. In the normal mode running from the computer, the active program line is highlighted for program tracing. This introduces a 7msec delay (approx.) between program lines. Also the program can be run in a pause and step mode for testing and troubleshooting.

The programming software provides four command buttons for interacting with the indexer-drive: *DOWNLOAD, UPLOAD, EXECUTE,* and *STOP.*

DOWNLOAD - transfers the program from the Windows software to the MSC plugged into the serial port. The transfer takes about 3 seconds. The program must be downloaded before it can be executed. **NOTE:** If the *EXECUTE* button is pressed before downloading, the program previously loaded in the indexer-drive will execute and the results may not match what is shown on the computer screen.

Press the *DOWNLOAD* button near the middle of the screen. If the drive is on and connected properly, the *DOWNLOAD* dialog box will appear and show the progress of the download, which takes 1-3 seconds. (The transfer time is governed by the speed at which the MSC can rewrite its internal, nonvolatile memory, and by the size of the program.) Once the download has been completed, the program is ready to execute.

UPLOAD - allows extraction of whatever program is in the MSC memory and display it on the screen. To modify a program already in the MSC but never saved to the hard drive, use the *UPLOAD* command to bring it back from the MSC.

EXECUTE - tells the MSC to begin running the program currently in its internal memory, starting on line 1. After clicking the *EXECUTE* button, a box Figure 32 will appear with options for the mode of execution, and status indicators for inputs and outputs.

If your MSC has firmware prior to 1.40, the Execute box will be simpler than this and some commands will not be available. If the program to be executed contains any PIT instructions (see page 7-2) then a different Execute box with a PIT emulation will appear (page 7-8).

STOP - interrupts the indexer-drive at any point in the program and closes the Execute box. This feature is useful when the drive starts doing things not intended for it to do. **NOTE:** Remember that the MSC is a small computer and computers do not do what the operator wants them to do, they do what the operator tells them to do.

If a motor and 2 turn per inch actuator is connected to the drive, the program can be tested. A normally-open-type momentary contact switch is required. Connect one end of the switch to *INPUT 1*. Connect the other end to any of the *GND* terminals.

Jog to the motor end of the actuator. Click the *EXECUTE* button. The *EXECUTE* dialog box will appear (see Figure 29).

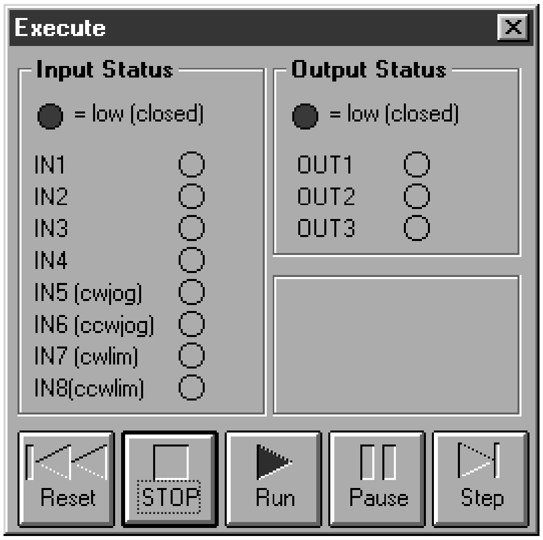


Figure 29 - EXECUTE DIALOG BOX

RUN - The *RUN* button is lighted green (ready) and the active line of program is highlighted.

Each time the switch is closed the motor should move 5 inch and return. (If the switch remains closed, the program will continually repeat.)

I/O STATUS - The execute box displays the status of the I/O during program execution.

PAUSE - The *PAUSE* button halts the program but does not close the Execute box. While paused the display of inputs is continuously updated in response to the source status.

STEP - When in Pause, clicking on Step will execute the highlighted program line, then automatically pause again. Clicking on Run will make the program run again from where paused.

RESET - Clicking Reset sends the program back to line 1 in either Pause or Run mode.

The MSC will run its stored program immediately on application of power when not connected to a computer with MSC Programmer Software open. Starting the program with a *WAIT PROMPT* or a *PIT PROMPT* will prevent unintended moves.

COPYING INSTRUCTIONS

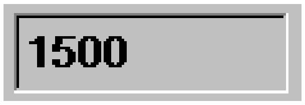
There may be occassions where it would be useful to make an exact copy of an instruction, or perhaps a copy with only one or two parameter changes. The fastest way to copy an instruction from one line to another is to point the mouse at an instruction icon that is to be copied and drag the icon onto another one elsewhere in the program. This is referred to as “Drag and Drop.”

For example, if there is a *SET OUTPUT* instruction on line 5 of your program, and an identical *SET OUTPUT* is required on line 11. Position the mouse over the *SET OUTPUT* icon on line 5, then click and hold the mouse button. Move the mouse until the icon is over the line 11 icon. let go of the button, and the instruction with all of its parameters will be copied to line 11.

Setting Program Parameters from the Computer

Windows-based programming makes it easy to enter data into the program.

TEXT BOXES



1500

Large numbers that must be entered precisely, such as the number of steps to move, are entered in ***text boxes***. This is much like using a word processor, but only one word is being edited. Just click in the box and type in the number. If a mistake is made, select all or part of the number and type something else instead. When a dialog box containing text boxes is first encountered, the text inside one box is automatically selected. This allows typing in the number without having to click at all.

If a number is entered that is too large, the MSC Programmer will change the entry so that it is “in bounds” when it moves to the next entry in the dialog box. This prevents asking the indexer-drive to do something beyond its capability.

SCROLL BARS

Many parameters are set using ***scroll bars***. Scroll bars work like the temperature controls in many cars. Sliding the small box to the right increases the value of the parameter. Sliding the box all the way to the right provides the maximum number. As the slider moves, the numerical value is shown in a box next to the scroll bar.

Refer to Figure 30. To make precise adjustments, try clicking next to the ***slider*** (on the bar, between the sliding box and the arrow). To make even more precise adjustments, click on the ***arrow*** at either end of the scroll bar.

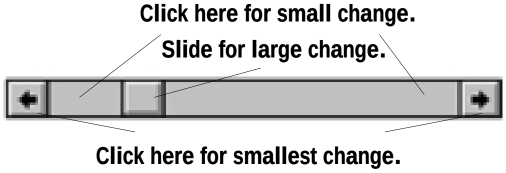


Figure 30 - PRECISE SCROLL BAR ADJUSTMENT

OPTION BUTTONS

Option buttons are very common in Windows. They are normally used to pick one item from a group, such as *CW* or *CCW* direction. To pick an item that is controlled by option buttons, click on the circle so it shows a black dot inside as shown in Figure 31.

Figure 31 - OPTION BUTTON

SPIN BUTTONS

Some parameters have a limited number of possible values, all numerical, but too many for option buttons. For example, there are 100 possible values in setting the line number in a *GO TO* instruction. For each, use the ***spin button***. If the down arrow part of the button is clicked (see Figure 32), the value goes down by one. Clicking the ***up arrow*** makes it go up by one. Clicking and holding down the mouse button will cause the parameter value to "spin up" quickly.

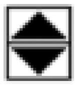


Figure 32 - SPIN BUTTON

Notes:

7 PIT Interface

Overview

The MSC is available with an optional Panel-mount Interface (PIT), sometimes called an ***operator panel.*** The optional *PIT INTERFACE* can be used to change parameters such as *MOVE DISTANCES, MOVE SPEED* or *REPEAT COUNT NUMBER*.

Using the Optional PIT

The PIT attaches to the same RS232 port used to connect to the computer, using the same cable. The PIT has a four-line liquid crystal display (LCD) and 20 keys for entering data. There are eight primary functions of the PIT:

1. ***Display a message*** on the LCD. *Examples:* Identify the machine ("ABC Bottle Filling Co. Model 20"), or display a status message ("Machine Running - Status OK").

2. ***Pause the program until the user presses* ENTER.** *Example:* Halt the process of applying preprinted labels while a new roll of labels is loaded.

3. ***Prompt the user to make a decision.*** *Example*: Offer the user an option such as changing set up parameters, that can be responded to by pressing the *YES* or *NO* keys.

4. ***Prompt the user to enter a move distance***. *Example:* Specify how long the material will be when used in a machine that feeds out material and then cuts it off.

5. ***Prompt the user for a move speed.*** *Example*: Adjust a feed rate, flow rate or other motor speed related setting.

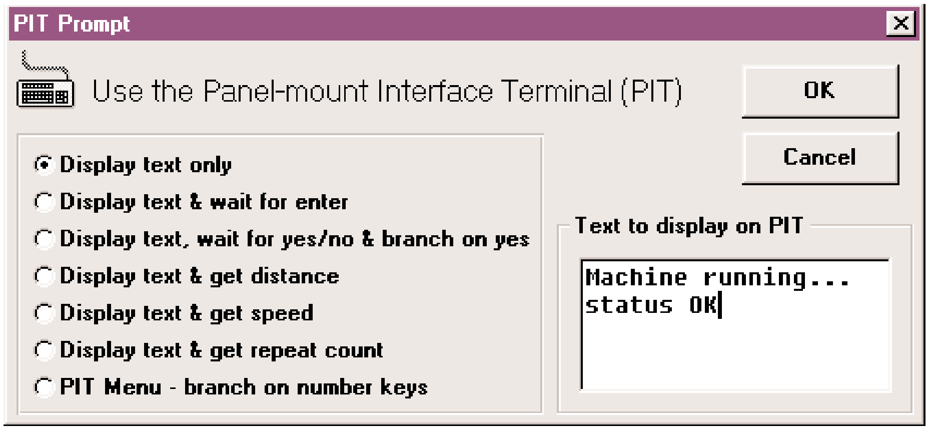
6. ***Prompt the user for a repeat count.*** *Example:* The user sets the number of parts that are processed or combines a repeat loop with a *WAIT TIME* instruction to adjust dwell time.

7. ***Multiply branching using keys 1-8.*** Offer the user a choice of jumping to eight different sub-routines, within the overall program.

8. ***Perform jogging.*** Use the left and right arrow keys for jogging while in waiting for ENTER.

TO DISPLAY A MESSAGE ON THE PIT (see Figure 33)

1. Click on the program line arrow icon.
2. Select the *PIT PROMPT* instruction.
3. Type a message (“Machine Running Status OK”) in the text box.
4. Select the *DISPLAY TEXT ONLY* option button.
5. Click the *OK* button. The message will stay on the LCD until another instruction uses the PIT.

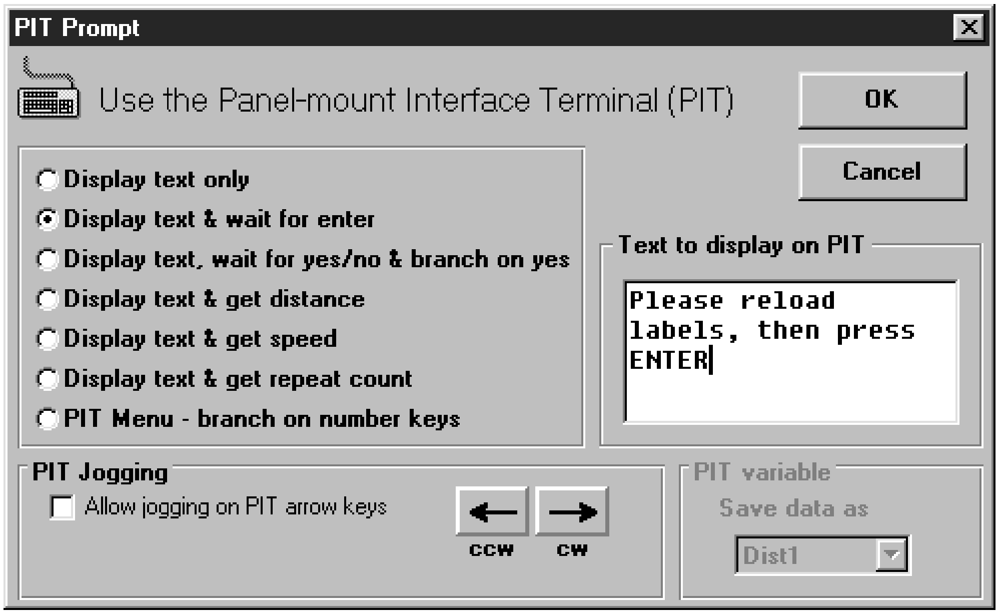

PIT Prompt
Use the Panel-mount Interface Terminal (PIT)
OK
Cancel
● Display text only
○ Display text & wait for enter
○ Display text, wait for yes/no & branch on yes
○ Display text & get distance
○ Display text & get speed
○ Display text & get repeat count
○ PIT Menu - branch on number keys
Text to display on PIT

```
Machine running...
status OK
```

Figure 33 - DISPLAYING A MESSAGE ON PIT

TO PAUSE UNTIL USER PRESSES ENTER (see Figure 34)

1. Click on the program line arrow icon.
2. Select the *PIT PROMPT* instruction.
3. Type a message (“Please reload labels, then press ENTER”) in the text box.
4. Select the *DISPLAY TEXT & WAIT FOR ENTER* option button.
5. To allow jogging using the PIT arrow keys, check the “Allow jogging” box.
6. Click *OK*.



PIT Prompt

Use the Panel-mount Interface Terminal (PIT)

○ Display text only
● Display text & wait for enter
○ Display text, wait for yes/no & branch on yes
○ Display text & get distance
○ Display text & get speed
○ Display text & get repeat count
○ PIT Menu - branch on number keys

Text to display on PIT: Please reload labels, then press ENTER

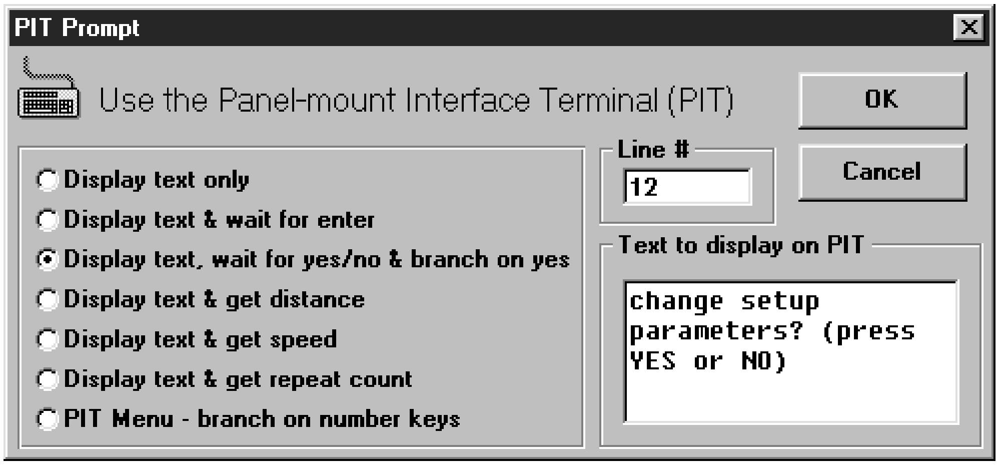
PIT Jogging
□ Allow jogging on PIT arrow keys

PIT variable
Save data as: Dist1

Figure 34 - PAUSING UNTIL USER PRESSES ENTER

TO LET USER MAKE A DECISION (PIT BRANCHING) (see Figure 35)

1. Select the *PIT PROMPT* instruction for the program line.
2. Type a message ("Change setup parameters? (press yes or no)") in the text box.
3. Select the option button *DISPLAY TEXT, WAIT FOR YES/NO & BRANCH ON YES.*
4. Type the Program Line # ("12") to go to on *YES* in the line # box.
5. Click *OK.*
6. Enter program or comment(s) (parameter setting instructions), starting on Program Line # ("12").
7. If required, place a *GO TO LINE #* instruction at the end of the parameter-setting instructions.



○ Display text only
○ Display text & wait for enter
● Display text, wait for yes/no & branch on yes
○ Display text & get distance
○ Display text & get speed
○ Display text & get repeat count
○ PIT Menu - branch on number keys

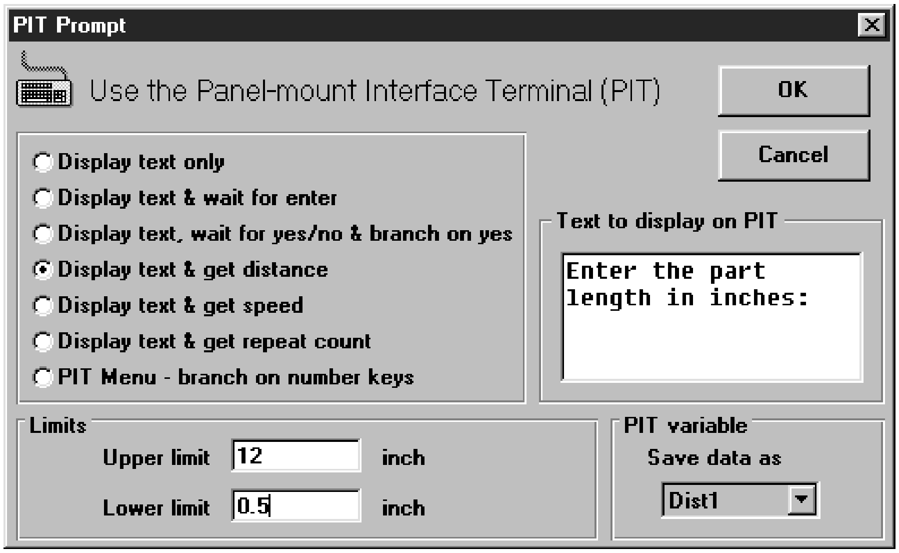
Line #: 12

Text to display on PIT: change setup parameters? (press YES or NO)

Figure 35 - PIT BRANCHING

TO ASK THE USER FOR A MOVE DISTANCE (see Figure 36)

1. Select the *PIT PROMPT* instruction for the program line.
2. Type a message ("Enter part length, in inches") in the text box.
3. Select the option button *DISPLAY TEXT AND GET DISTANCE.*
4. Enter upper and lower limits (this example allows the operator to enter distances between 0.5 and 12 inches).
5. Select a PIT variable to store the distance in (choose *Dist1* this time, but any of the eight PIT variables is acceptable for storing any type of data).
6. Later in the program, provide a feed instruction (*FEED TO LENGTH, FEED & RETURN*, etc.) that uses the *Dist1* variable for distance.



○ Display text only
○ Display text & wait for enter
○ Display text, wait for yes/no & branch on yes
● Display text & get distance
○ Display text & get speed
○ Display text & get repeat count
○ PIT Menu - branch on number keys

Text to display on PIT: Enter the part length in inches:

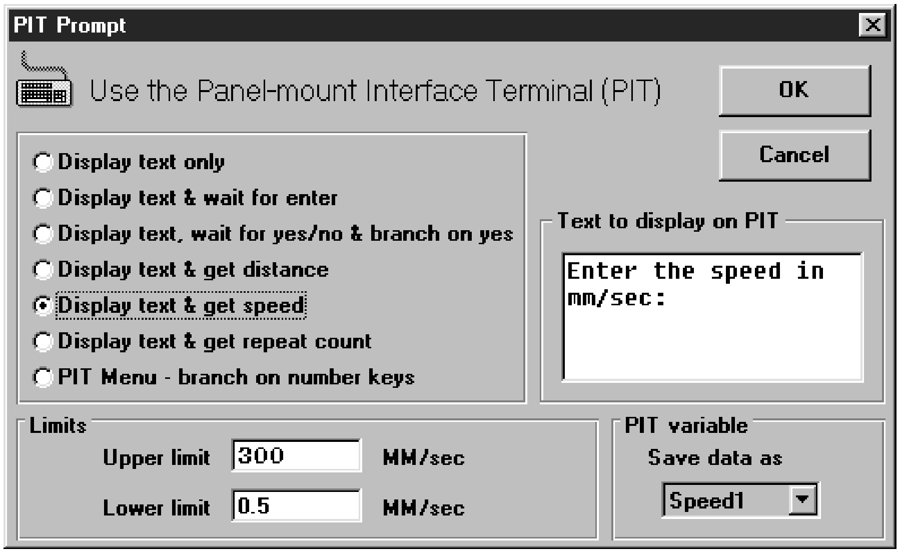
Limits
Upper limit: 12 inch
Lower limit: 0.5 inch

PIT variable
Save data as: Dist1

Figure 36 - PROMPT USER FOR MOVE DISTANCE

TO GET A SPEED FROM THE USER (see Figure 37)

1. Select the *PIT PROMPT* instruction for the program line.
2. Type a message (“Enter the speed, in mm/sec”) in the text box.
3. Select the option button *DISPLAY TEXT AND GET SPEED.*
4. Enter upper and lower limits (this example allows the operator to enter speeds between 0.5 and 50 mm/sec).
5. Select an PIT variable in which to store the speed (choose *Speed1*).
6. Later in the program, provide a feed instruction (*FEED TO LENGTH, FEED & RETURN*, etc.) that uses the *Speed1* variable for speed.



○ Display text only
○ Display text & wait for enter
○ Display text, wait for yes/no & branch on yes
○ Display text & get distance
● Display text & get speed
○ Display text & get repeat count
○ PIT Menu - branch on number keys

Text to display on PIT: Enter the speed in mm/sec:

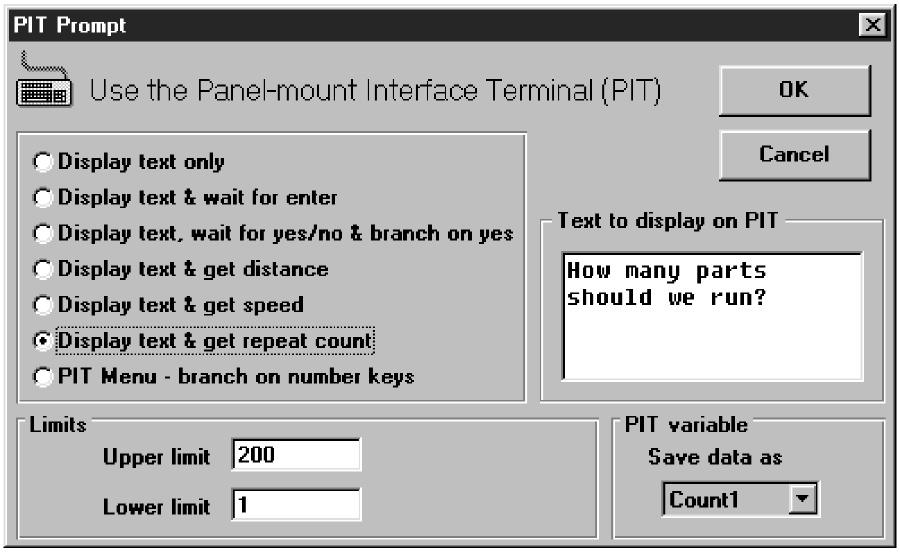
Limits
Upper limit: 300 MM/sec
Lower limit: 0.5 MM/sec

PIT variable
Save data as: Speed1

Figure 37 - PROMPT USER FOR SPEED

TO GET A REPEAT COUNT FROM THE USER (see Figure 38)

1. Select the *PIT PROMPT* instruction for the program line.
2. Type a message ("How many parts should we run?") in the text box.
3. Select the option button *DISPLAY TEXT AND GET REPEAT COUNT.*
4. Enter upper and lower limits (this example allows the operator to enter a number between 1 and 200).
5. Select an PIT variable to store the count in (choose *Count1*).
6. Later in the program, there must be a *REPEAT* instruction that uses the *Count1* variable for the repeat count.



○ Display text only
○ Display text & wait for enter
○ Display text, wait for yes/no & branch on yes
○ Display text & get distance
○ Display text & get speed
● Display text & get repeat count
○ PIT Menu - branch on number keys

Text to display on PIT

```
How many parts
should we run?
```

Limits

Upper limit 200

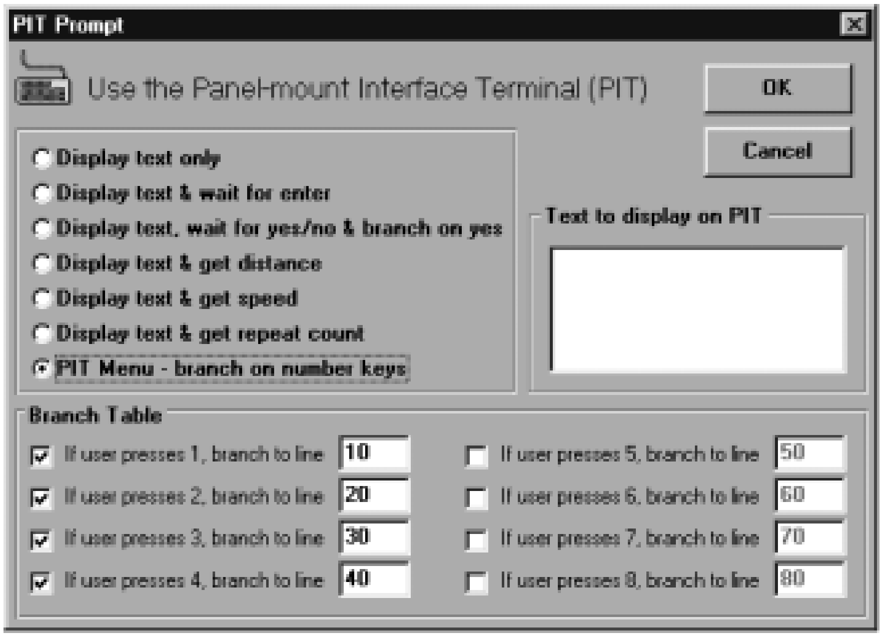
Lower limit 1

PIT variable

Save data as Count1

Figure 38 - PROMPT USER FOR REPEAT COUNT

TO CREATE A PIT MENU (see Figure 39)



PIT Prompt

Use the Panel-mount Interface Terminal (PIT)

○ Display text only
○ Display text & wait for enter
○ Display text, wait for yes/no & branch on yes
○ Display text & get distance
○ Display text & get speed
○ Display text & get repeat count
● PIT Menu - branch on number keys

Text to display on PIT

Branch Table

☑ If user presses 1, branch to line 10
☑ If user presses 2, branch to line 20
☑ If user presses 3, branch to line 30
☑ If user presses 4, branch to line 40
☐ If user presses 5, branch to line 50
☐ If user presses 6, branch to line 60
☐ If user presses 7, branch to line 70
☐ If user presses 8, branch to line 80

Figure 39 - PROMPT USER FOR SUBROUTINE SELECTION

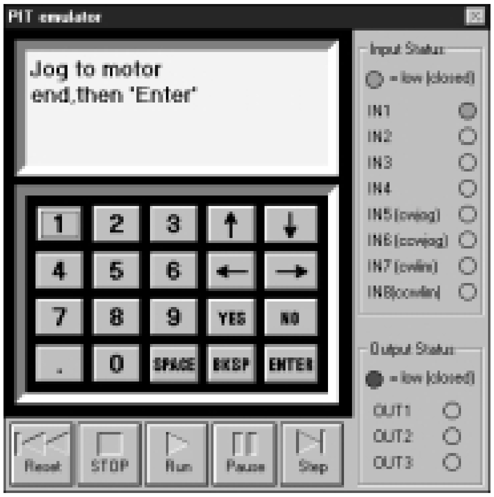
1. Select the *PIT PROMPT* instruction for the program line.
2. Select the option button “PIT Menu…”
3. Type a message in the text box.
4. Check the boxes indicating which numeral keys are to be used (in this example, 1, 2, 3 and 4).
5. Assign a program line number to each key (10, 20, 30 and 40).
6. Later in your program, there must be instructions at each of the lines you’ve specified. These are the instructions that will execute if the user presses the appropriate key on the MMI. For example, when the user presses ‘1’, the program will branch to line 10.

NOTE: If you add extra spaces to your display text to get the look “just right”, watch out: sometimes the *MSC Programmer* removes those spaces when you re-open the instruction dialog. It is safer to format your text using other characters, like ‘.’ (period) or ‘_’ (underscore).

NOTE: You need firmware version 1.41 or later to execute the PIT Menu function.

TO RUN PROGRAM USING THE PIT FROM A COMPUTER (see Figure 40)

To run a program (which includes PIT instructions) from the computer, click the *EXECUTE* button. The PIT *EXECUTE* dialog box looks and acts like the real PIT (see Figure 33). It will display messages, and clicking on the buttons will enter data. The PIT emulator *EXECUTE* box has all the display, run and step functions of the basic *EXECUTE* box (page 35).



Jog to motor
end,then 'Enter'

Figure 40 - ON-SCREEN PIT EMULATION

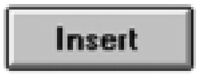
Programming Tools 8

Overview

There are five programming tools which appear in the *PROGRAM LINE* dialog box (page 6-10, Figure 27). In addition the *CLEAR* button at the top right of the Main Programming window provides an additional tool function.

Programming Tools

INSERT NEW STEP



This command is used to insert a new instruction anywhere in the program.

To insert a new step:

1. Click on the program icon where the new instruction is to go. The *PROGRAM LINE...* dialog box will appear.
2. Click on the command button marked “Insert”. The instructions are re-ordered, and the line needed for the new program line is available.
3. Click on the *OPEN* line and select an instruction.

DELETE STEP

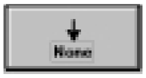
In addition to inserting a line in the program, deletions can be made to make room for other lines farther down. For example, space exists in the middle of the program, but additional instruction is needed near the end.

To delete a step:

1. Click on a program line that is not needed. The *PROGRAM LINE...* dialog box will appear.
2. Click on *DELETE* to remove the selected line and move the other lines up one position. A blank spot will be created at the end of the program.

NOTE: By combining insertions and deletions, program lines can be placed wherever and whenever needed.

NONE



This tool on the *PROGRAM LINE* dialog box leaves a blank program line for future use without the need to Insert later.

CANCEL

Cancel

This tool exits from programming without saving or downloading the program.

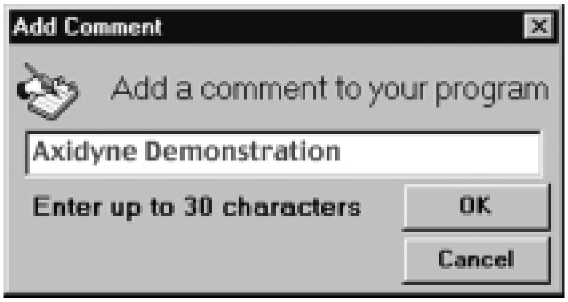
COMMENT

Comment

This tool provides for insertion of notes into the program. Comments help organize the program and make future changes easier.

Whether the program is saved to disk or downloaded to the drive, comments stay with it. They do not affect the way the program runs. When the MSC executes a program, it simply skips over the comments.

Refer to Figure 41. Placing a comment on the first line of the program lets future programmers know who wrote the program, when it was written, and what it does. Adding comments to the program may save a great deal more time later on, when program steps are no longer obvious.



Add Comment

Add a comment to your program

Axidyne Demonstration

Enter up to 30 characters

Figure 41 - COMMENT DIALOG BOX

There is a limit to the number of comments the program can have. The MSC has a ***string pool*** of 400 characters. All *PIT PROMPTS* text goes into the string pool, as any *COMMENT, WAIT INPUT* or *IF INPUT* instructions whose strings exceed 12 characters.

CLEAR

Clicking on the *CLEAR* button at the top right of the programming screen and then confirming *OK* will erase all the program stored and displayed in the active main programming window Lines 1-100. *CLEAR* does not clear the program stored in a connected MSC.

COPYING INSTRUCTIONS

The fastest way to make an exact copy of an instruction or perhaps a copy with small changes is to point the mouse at the instruction icon to be copied, click and hold and drag the icon to the copy location. Release the mouse button. This is “Drag and Drop”.

Notes:

Programming Instructions 9

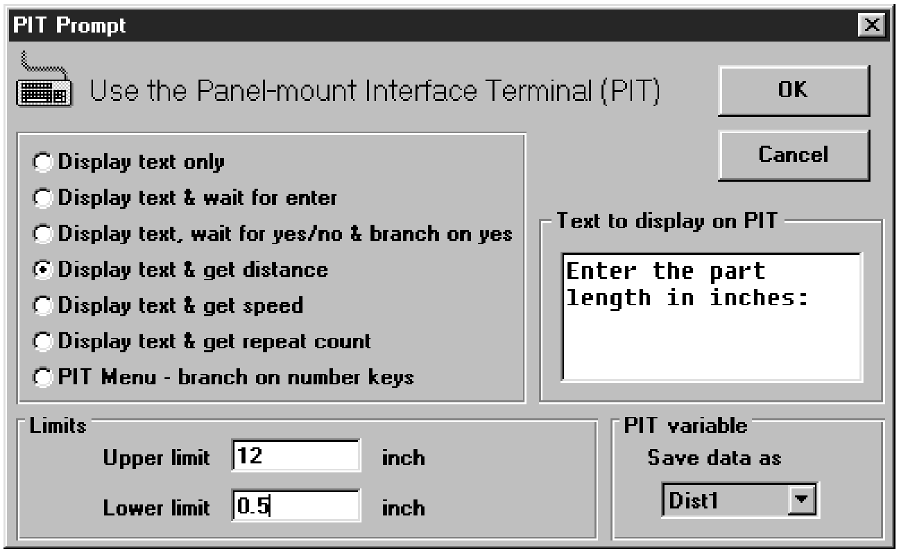
Overview

There are 16 Programming Instructions which appear in the *PROGRAM LINE* dialog box. Each icon, (except End Repeat) accesses an individual dialog box in which the instruction is fully defined.

Programming Instructions

PIT PROMPT

The *PIT PROMPT* instruction is used with the optional PIT (Panel Mount Interface). *PIT PROMPTs* allow the program to display messages on the PIT screen, and can gather data from the operator to be used by other instructions (see Figure 42). The PIT can also pause the program until the user presses the *ENTER* button. It can allow the user to make a decision, then press the *YES* or *NO* button. If the user presses *YES*, the program branches to another program line. If the user presses *NO*, the program goes to the next line. If authorized the PIT can be used as a jog switch, while waiting for enter.



Use the Panel-mount Interface Terminal (PIT)

○ Display text only
○ Display text & wait for enter
○ Display text, wait for yes/no & branch on yes
● Display text & get distance
○ Display text & get speed
○ Display text & get repeat count
○ PIT Menu - branch on number keys

Text to display on PIT

```
Enter the part
length in inches:
```

Limits
Upper limit 12 inch
Lower limit 0.5 inch

PIT variable
Save data as Dist1

Figure 42 - PIT PROMPT DIALOG BOX

To display a message (such as “Machine Running - Status OK”), put an *PIT PROMPT* instruction in the program at the point where the message is to appear. Check the option button marked “Display Text Only” and type in the message. Once the *PIT PROMPT* instruction has been executed, the message will stay on the screen until changed by another instruction that uses the PIT display.

If it is required that the operator be able to change parameters like *DISTANCE, SPEED* or *REPEAT COUNT*, an *PIT PROMPT* is required to ask the user for data and to store it in nonvolatile memory. In this case, click on the option button for the type of required: *DISTANCE, SPEED* or *REPEAT COUNT.*

Upper and lower limits must be set. The *PIT PROMPT* instruction will check the data entered against the limits specified and tell the user if a value is out of range. For example, if the *PIT PROMPT* is set to gather a *REPEAT COUNT*, and the upper and lower limits are set to 100 and 1, the operator will not be able to enter any value bigger than 100 or smaller than 1. The PIT will not accept limits greater than the MSC can allow.

The *PIT PROMPT* instruction must be told where to store the data in nonvolatile memory. There are eight locations to choose from. They are named *DIST1, DIST2, DIST3, SPEED1, SPEED2, COUNT1, COUNT2* and *COUNT3*. Remember where the *PIT PROMPT* was instructed to put the data. When an instruction is set up to use data from an PIT variable, that instruction must be told which variable to use (*DIST1, DIST2,* etc.). Any location can be used for any variable (i.e. distances could be put in speed or count locations).

For example, if the operator is to be able to set the number of parts the machine produces in a given run, put an *PIT PROMPT* instruction in the program to ask for a *REPEAT COUNT* and to save it as *COUNT1*. Set up a *REPEAT* loop somewhere else in the program to process the parts. The loop will start with a *REPEAT* instruction configured to get its repeat count from the PIT variable *COUNT1*.

If a *PIT PROMPT* instruction is to get distance or speed data, User Units will be as selected on the Main Programming window.

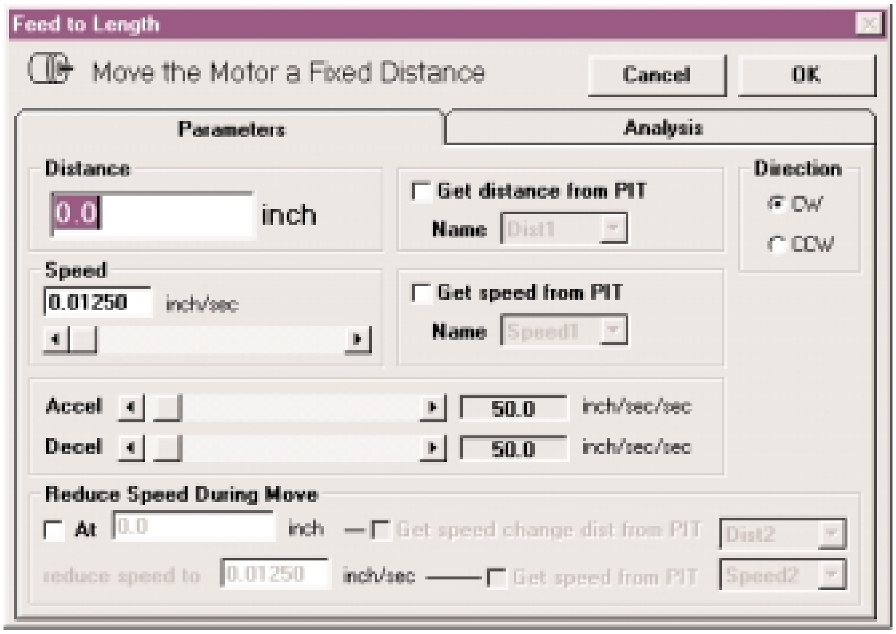
To pause the program until the user presses the *ENTER* key on the PIT, choose the option marked “Display text & wait for enter.”

To allow the user to make a decision, select *DISPLAY TEXT,* wait for "Yes/No & branch on yes.” Be sure to enter a line number in the *LINE # BOX*. The program will jump to that line if the user presses *YES*. If the user presses NO, the program will execute the next line after the *PIT PROMPT*.

FEED TO LENGTH

This instruction is used for point-to-point incremental moves. This is the instruction to use to move the motor a fixed number of user units. Speed or distance data previously gathered by an *PIT PROMPT* instruction can also be used.

Clicking on the *FEED TO LENGTH* button in the *PROGRAM LINE...* dialog box causes the *FEED TO LENGTH* dialog box appear (see Figure 43). This is where the parameters for the move are entered.



Distance: 0.0 inch

□ Get distance from PIT — Name: Dist1

Direction: ● CW ○ CCW

Speed: 0.01250 inch/sec

□ Get speed from PIT — Name: Speed1

Accel: 50.0 inch/sec/sec

Decel: 50.0 inch/sec/sec

Reduce Speed During Move

□ At 0.0 inch — □ Get speed change dist from PIT: Dist2

reduce speed to 0.01250 inch/sec — □ Get speed from PIT: Speed2

Figure 43 - FEED TO LENGTH DIALOG BOX

DISTANCE - the number of user units to move. (The maximum number of steps is 16,000,000.) Selecting the check box marked “Get distance from PIT” allows a choice of one of the eight PIT variables as the distance. **NOTE:** Checking “Get distance from PIT” does not automatically make the MSC stop and ask the user for an entry. A *PIT PROMPT* instruction is needed elsewhere else in the program for that.

SPEED - the maximum speed the motor is to go, in user units. (The speed can be set anywhere between .025 and 50 rev/sec, in increments of .025 rev/sec.) Selecting the check box marked “Get speed from PIT” allows a choice of one of the eight PIT variables as the speed.

The Feed to Length instruction also allows for reducing speed during the move. Check the option box and enter the required values, or choose to enter the values via the PIT.

ACCEL - step motors cannot achieve a high speed instantly. The indexer-drive must gradually accelerate the motor to speed. The acceleration rate depends on the inertia of the motor and load, the torque available from the motor, and how fast it is to go. (For trapezoidal move profiles (speed vs time), acceleration and deceleration are limited to 3000 revs/sec/sec. For triangular move profiles, where the target speed is never reached, the sum of the accel and decel rates must not exceed 4000 rev/sec/sec. Thus, the fastest profile is achieved with accel and decel at 3000 revs/sec/sec and speed at the trapezoidal limit. This profile may not be possible in the application because of mechanical limitations.)

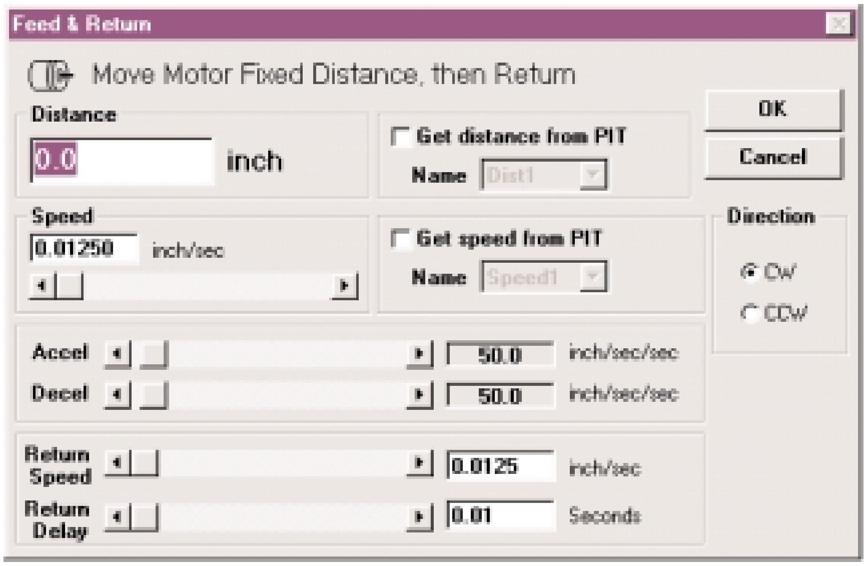
DECEL - this is the rate at which the drive decelerates to a stop at the end of the move. The range is the same as for acceleration. Because friction encourages a motor to stop, decel can often be set higher than accel.

DIRECTION - Choose *CW* or *CCW* as the direction for the move. Simply dot the appropriate circle by clicking on it.

ANALYSIS - This button presents a speed-time and a speed-distance graph of the move along with a calculation of the move duration and the times and distances spent accelerating and decelerating. Peak speed is also displayed.

FEED & RETURN

The *FEED & RETURN* instruction (Figure 44) is used for point-to-point moves with return to the starting point (e.g. a motor driving a cut-off knife is to retract the knife after cutting).



Move Motor Fixed Distance, then Return

Distance: 0.0 inch

□ Get distance from PIT
Name: Dist1

Speed: 0.01250 inch/sec

□ Get speed from PIT
Name: Speed1

Direction
● CW
○ CCW

Accel: 50.0 inch/sec/sec
Decel: 50.0 inch/sec/sec

Return Speed: 0.0125 inch/sec
Return Delay: 0.01 Seconds

Figure 44 - FEED & RETURN DIALOG BOX

FEED & RETURN requires many of the same parameters as *FEED TO LENGTH*: *DISTANCE*, *SPEED*, *ACCEL*, *DECEL* and *DIRECTION*. For explanations of these, see “Feed to Length” on page 9-3.

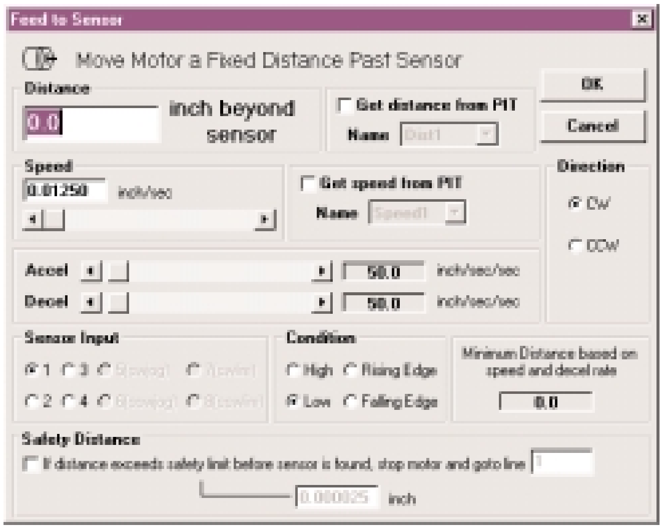
RETURN SPEED must also be set. In the case of the cut-off knife, set to feed slowly as the knife is cutting, then retract quickly. Thus, the return speed would be higher than the forward speed.

RETURN DELAY determines how long the MSC waits between the end of the feed move and the start of the return. This could, for example, give the machine time to remove a part before retracting. Since a motor and load need time to “settle out” after moving, do not set return delay to less than 0.2 seconds unless it is certain that the motor and load settle more quickly than normal.

FEED TO SENSOR

This instruction (Figure 45) allows the motor to move until an external event changes the state of an input.

One useful application for *FEED TO SENSOR* is when the motion distance varies. An example of this is using a step motor to dispense labels that come on a roll. The spacing of the labels is not exact, so it is not practical to simply feed out the same number of steps each time. Instead, use a sensor on the feed mechanism that “sees” the edge of each label and signals one of the MSC inputs to stop motion.



Feed to Sensor

Move Motor a Fixed Distance Past Sensor

Distance: 0.0 inch beyond sensor

□ Get distance from PIT Name: Dist1

Speed: 0.01250 inch/sec

□ Get speed from PIT Name: Speed1

Direction: ● CW ○ CCW

Accel: 50.0 inch/sec/sec

Decel: 50.0 inch/sec/sec

Sensor Input: ● 1 ○ 3 ○ 5(ccwjog) ○ 7(ccwlim) ○ 2 ○ 4 ○ 6(cwjog) ○ 8(cwlim)

Condition: ○ High ○ Rising Edge ● Low ○ Falling Edge

Minimum Distance based on speed and decel rate: 0.0

Safety Distance

□ If distance exceeds safety limit before sensor is found, stop motor and goto line 1

0.000025 inch

Figure 45 - FEED TO SENSOR DIALOG BOX

FEED TO SENSOR will ask for many of the same parameters as the other feed programs: *SPEED, ACCEL, DECEL* and *DIRECTION*. It is also necessary to specify a distance since the MSC must have enough space to decelerate to a stop once the sensor is tripped. The higher the speed, the longer it will take to stop. If the decel rate is increased, the motor can stop in fewer steps. The *MINIMUM DISTANCE* box tells how many steps must be allowed based on the speed and decel rate that is set. The distance cannot be set to less than this minimum.

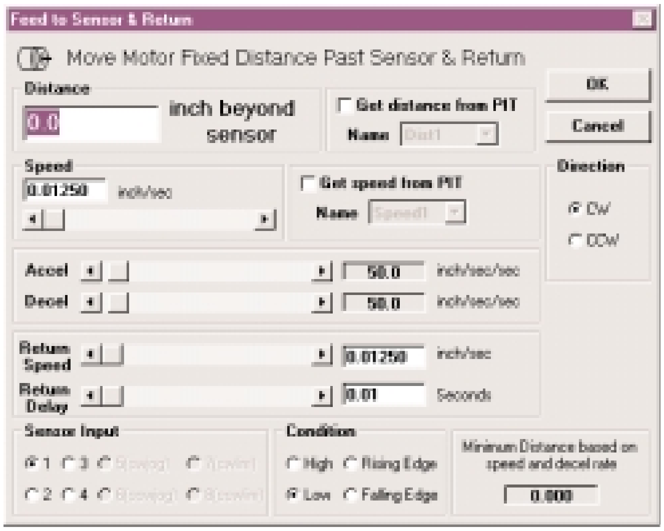
The MSC must also be told which input the sensor is wired to and what input condition to look for. The four input conditions are:

HIGH - move until the specified input reaches a high voltage (no current) state. This is the default state of an input if nothing is connected to it.

LOW - move until specified input is at a low voltage (closed) state.
RISING EDGE - move until the signal goes from low (current flowing) to high (no current flowing). This is similar to the high condition, but the difference is important. Assume a sensor is wired to the MSC, that will go high when motion is to stop. However, the sensor signal stays high after motion is complete, going low at a later time. This often happens in labeling applications where there isn’t much space on the roll between labels. By choosing high as the input condition, the MSC will complete the motion, then refuse to start again because the input signal is still high. If a rising edge is chosen, the MSC can be restarted with the input voltage high and stop again when the sensor signal again goes from low to high.
FALLING EDGE - the opposite of rising edge. The MSC only recognizes the high to low transition (no current to current flow).
SAFETY DISTANCE - A safety distance and program line can be entered such that if the system does not see or reach the sensor, then, at the safety distance, motion (feed instruction) will stop and the program will go to the referenced line.

FEED TO SENSOR & RETURN

This instruction is the same as *FEED TO SENSOR*, except the motor returns to the starting point after the move.



Feed to Sensor & Return

Move Motor Fixed Distance Past Sensor & Return

Distance: 0.0 inch beyond sensor

□ Get distance from PIT — Name: Dist1

Speed: 0.01250 inch/sec

□ Get speed from PIT — Name: Speed1

Direction: ● CW ○ CCW

Accel: 50.0 inch/sec/sec

Decel: 50.0 inch/sec/sec

Return Speed: 0.01250 inch/sec

Return Delay: 0.01 Seconds

Sensor Input: ● 1 ○ 3 ○ 5(cwjog) ○ 7(cwlim) ○ 2 ○ 4 ○ 6(ccwjog) ○ 8(ccwlim)

Condition: ○ High ○ Rising Edge ● Low ○ Falling Edge

Minimum Distance based on speed and decel rate: 0.000

Figure 46 - FEED TO SENSOR & RETURN DIALOG BOX

Most of the parameters are the same as *FEED TO SENSOR*, except two new ones are added: *RETURN SPEED* and *RETURN DELAY.*

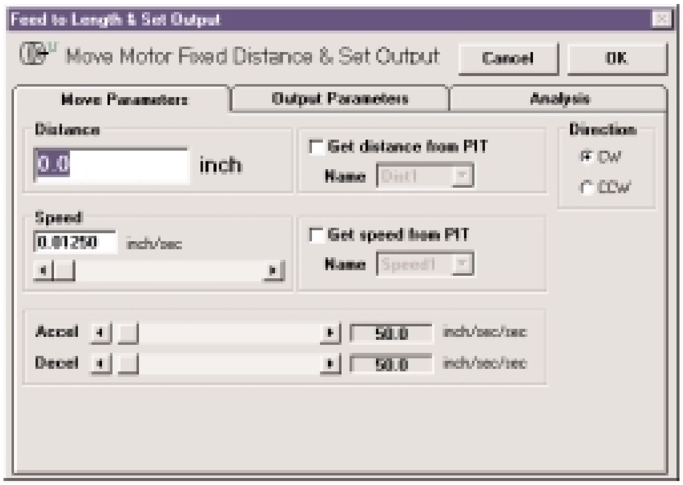
A useful application of *FEED TO SENSOR & RETURN* is a variable distance application. If a machine cuts fabric of different sizes and the MSC is driving the cutoff knife, it is appropriate to set a sensor at the end of the cut off stroke. That way, manual adjustment can be for the width of material being used on a particular day without having to reprogram the MSC.

The indexer-drive feeds each time it is triggered until the knife trips the sensor, then returns to the starting point.

NOTE: The maximum distance for any program is 16,000,000 steps, the longest distance the MSC can track. If a move of more than 16 million steps is made before reaching the sensor, the MSC will not return to the correct position. If this is a problem, consider selecting a lower microstep resolution. At 50,000 steps/rev, the 16 million step limit is exceeded after 320 revolutions. At 2000 steps/rev, 8000 revs can be made before exceeding the limit.

FEED & SET OUTPUT

The *FEED & SET OUTPUT* instruction (Figures 47, 48, 49) allows combination of a Feed to Length instruction with a Set Output instruction, making the program shorter. Feed to Length and Set Output are frequently used in combination to make the MSC signal another device when it finishes a move.


Feed to Length & Set Output
Move Motor Fixed Distance & Set Output
Cancel
OK
Move Parameters
Output Parameters
Analysis

Distance: 0.0 inch

☐ Get distance from PIT
Name: Dist1

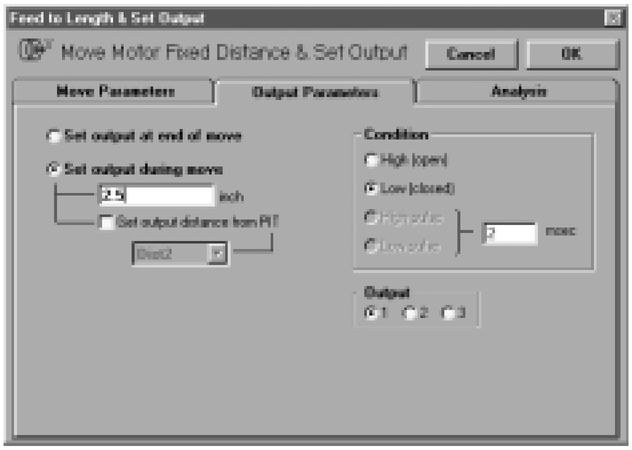
Direction: ● CW ○ CCW

Speed: 0.01250 inch/sec

☐ Get speed from PIT
Name: Speed1

Accel: 50.0 inch/sec/sec
Decel: 50.0 inch/sec/sec

Figure 47 - MOVE PARAMETER DIALOG BOX


Feed to Length & Set Output
Move Motor Fixed Distance & Set Output
Cancel
OK
Move Parameters
Output Parameters
Analysis

○ Set output at end of move
● Set output during move
2.5 inch
□ Set output distance from PIT
Dist2

Condition
○ High (open)
● Low (closed)
○ High pulse
○ Low pulse
2 msec

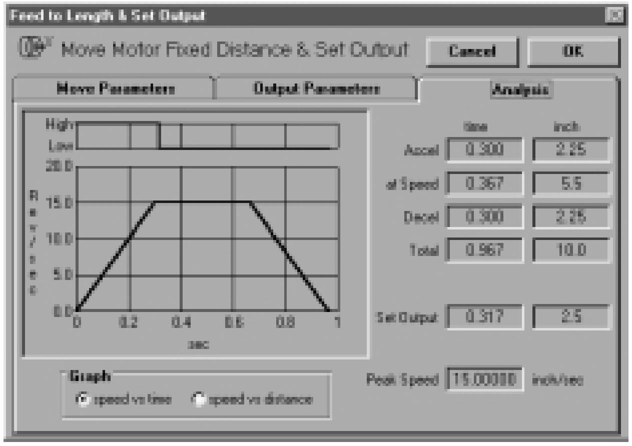
Output
● 1 ○ 2 ○ 3

Figure 48 - OUTPUT PARAMETER DIALOG BOX

Also the Feed & Set Output feature can improve manufacturing throughput. The MSC may be advancing a part which will then be processed by another device; for example feeding material to be cut off by a blade. If the blade requires some time to approach the material, it helps to trigger the blade before the material is fully advanced. That way, more pieces in the same amount of time.

Feed & Set Output allows an output to be set high or low at any point during the move. The Output Parameter Dialog Box allows flexible setting of the output timing and condition.

The analysis display (Figure 49) shows the motion profile and the transition of the output in either speed vs time or spread vs distance format.



	time	inch
Accel	0.300	2.25
at Speed	0.367	5.5
Decel	0.300	2.25
Total	0.967	10.0
Set Output	0.317	2.5

Peak Speed 15.00000 inch/sec

Graph
● speed vs time ○ speed vs distance

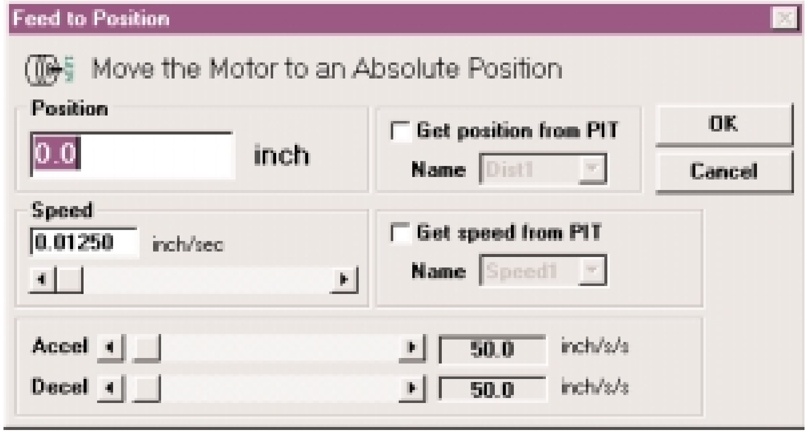
Figure 49 - ANALYSIS DISPLAY

FEED TO POSITION

This instruction moves the motor and load from wherever they are to an absolute position. For example, if the load is at the 4 inch position and the program executes a Feed to Position 6 inches, the motor will move two inches clockwise. If the load was at the 10 inch position and a Feed to Position 6 inches was commanded, the move would be 4 inches counter clockwise.

Feed to Position requires the usual move parameters: speed, accel & decel rates. Like other move instructions, speed can be recalled from a PIT variable, allowing it to be entered by the operator on the PIT panel.

Position can be a positive or negative number, Position can be entered on the PIT, but note that the PIT has no minus (-) key, so a negative number cannot be entered on the PIT. Avoid using negative absolute positions by using the Set Position instruction.



Feed to Position

Move the Motor to an Absolute Position

Position: 0.0 inch

☐ Get position from PIT
Name: Dist1

Speed: 0.01250 inch/sec

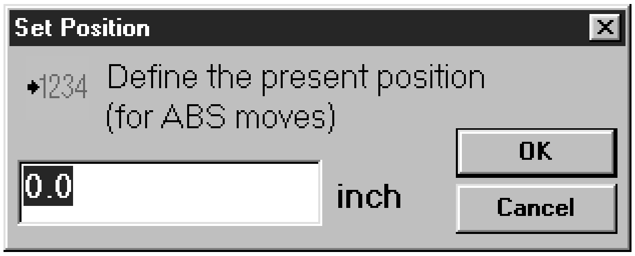
☐ Get speed from PIT
Name: Speed1

Accel: 50.0 inch/s/s
Decel: 50.0 inch/s/s

Figure 50 - FEED TO POSITION DIALOG BOX

SET ABS POSITION

This instruction (Figure 51) allows the present motor position to be defined as any absolute position. The Seek Home instruction automatically clears the absolute position counter when it's finished, defining the home position as 0. However, it may be preferable to define the home sensor as the 8 inch position, or 90 degrees, or whatever. Simply put a Set Position instruction after the Seek Home instruction, or anywhere else in your program where appropriate to define the absolute position.



Set Position

Define the present position (for ABS moves)

0.0 inch

Figure 51 - SET POSITION DIALOG BOX

If the PIT is to be used for entering Absolute Positions, choose the position setting to allow PIT entries to be positive values.

SAVE ABS POSITION

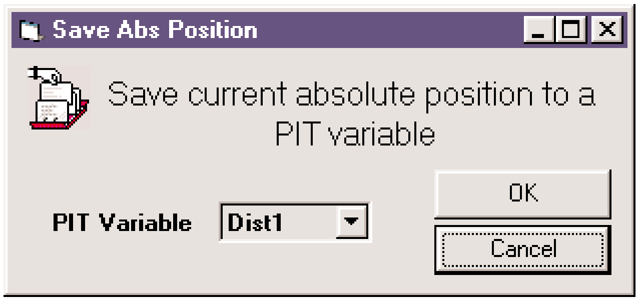
This instruction (Figure 52) allows the present absolute position to be saved to a PIT variable. This is useful if an operator is to visually position the load, and then be able to return to that position later in the program.

For a material handling application, a program could be created using the Wait Input command to allow the operator to move the load into position. The operator would then press an ENTER button to exit the Wait Input instruction. If the next instruction was Save Abs Position, the load position that the operator carefully obtained would be recorded in nonvolatile memory. Elsewhere in your program a Feed to Position instruction to return the load to the saved position could be used.

The MSC supports up to 8 PIT variables, so as many as eight different positions can be saved.

Even though the positions that the indexer has "learned" are still remembered after shutdown (because they are stored in nonvolatile memory), it is necessary to "home" the system each time it's powered up. Otherwise, the saved absolute positions don't make any sense.

NOTE: You need MSC with firmware version 1.52 or later to use the Save Abs Position instruction.


Save Abs Position

Save current absolute position to a PIT variable

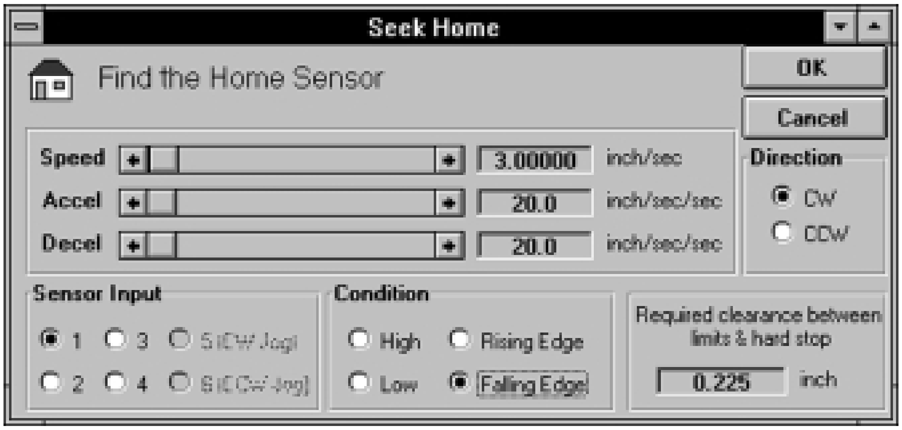
PIT Variable: Dist1

Figure 52 - SAVE ABS POSITION DIALOG BOX

SEEK HOME

This instruction allows the motor to move until a home sensor is found. The home sensor can be wired to any of the general purpose inputs.

Some applications require the motor to start from a certain position each time the power is turned on, but can't guarantee where it was left at the last power down. The solution is to wire a sensor to one of the MSC inputs and place a *SEEK HOME* command at or near the beginning of the program.



Speed 3.00000 inch/sec
Accel 20.0 inch/sec/sec
Decel 20.0 inch/sec/sec

Direction: ● CW ○ CCW

Sensor Input: ● 1 ○ 3 ○ 5 (CW Jog) ○ 2 ○ 4 ○ 6 (CCW Jog)

Condition: ○ High ○ Rising Edge ○ Low ● Falling Edge

Required clearance between limits & hard stop 0.225 inch

Figure 53 - SEEK HOME DIALOG BOX

SEEK HOME will ask for many of the same parameters as the other feed programs: *SPEED*, *ACCEL*, *DECEL* and *DIRECTION*. The MSC must also be told which input the sensor is wired to and what input condition to look for. The four input conditions are:

HIGH - move until the specified input reaches a high voltage (no current) state. This is the default state of an input if nothing is connected to it.
LOW - move until the specified input is at a low voltage (current flowing) state.
RISING EDGE - move until the signal goes from low (current flowing) to high (no current). This is similar to the high condition, but the difference is important. With a *SEEK HOME* command to a high input and the load already on the home sensor (causing the input to be high), the load will not move. Instead, choose "rising edge" and the MSC will move the load off and back to the home sensor.

FALLING EDGE - the opposite of rising edge. MSC waits for input voltage to go from high, to low (no current to current flowing). If the input is already low at the start of the home instruction, it is ignored. It must go high, then go low to be recognized.

The MSC begins a *SEEK HOME* command by moving the motor in the direction specified. If the home sensor is found, the motor decelerates to a stop, then backs up to the position at which it found the switch edge. If a limit is encountered before the home sensor is found, the MSC reverses the direction of motion, then proceeds to the home sensor. That way it always ends up at the sensor no matter which side of the home sensor the load is started on.

For more precise homing and to be sure the home position is on the same edge of the home sensor a double home move is required, with a short move in between to go off the sensor. This allows a slow speed move with very small decel distance for the final homing. The first home instruction can be faster.

Tol-O-Matic Application Engineers can provide recommendations.

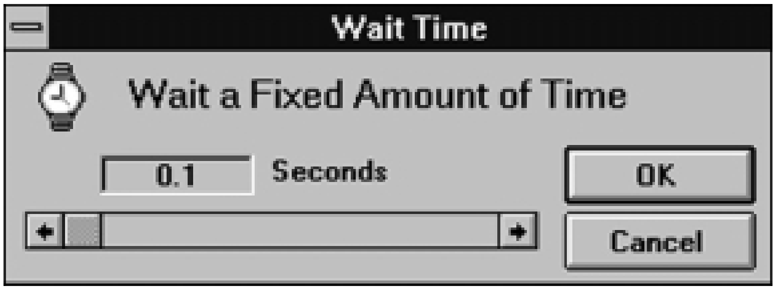
Always use a Hall Effect switch for the home sensor.

There is a box in the lower right-hand corner of the *SEEK HOME* dialog box. This tells how many steps the MSC needs to decelerate to a stop. The *REQUIRED CLEARANCE* box tells how much distance must be allowed between the limit sensors and any hard stop, based on the speed and decel rate that is set. If enough clearance is not allowed, the load may crash into something as it decelerates past a limit while seeking home.

The higher the speed, the longer it will take to stop. If the decel rate is increased, then the motor can stop in fewer steps.

WAIT TIME

This is the simplest instruction. Simply enter an amount of time, and the MSC will pause for that time before proceeding to the next line in the program. The range is 0.01 to 300 seconds.

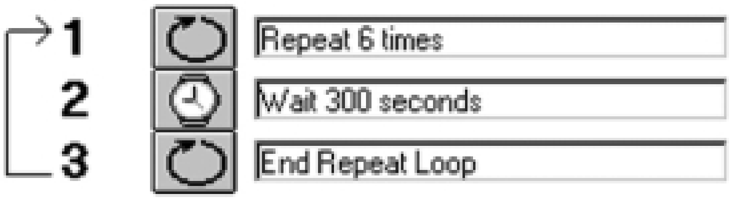

Wait Time

Wait a Fixed Amount of Time

0.1 Seconds

Figure 54 - WAIT TIME DIALOG BOX

The *WAIT TIME* instruction can be made to last longer (than 300 seconds) by placing a repeat loop around it. First factor the 3 minute delay into two parts. The most delay in one *WAIT TIME* instruction is 300 seconds. Therefore, six times 300 seconds results in 1800 seconds or 30 minutes. The program is shown in Figure 55.


1 Repeat 6 times

2 Wait 300 seconds

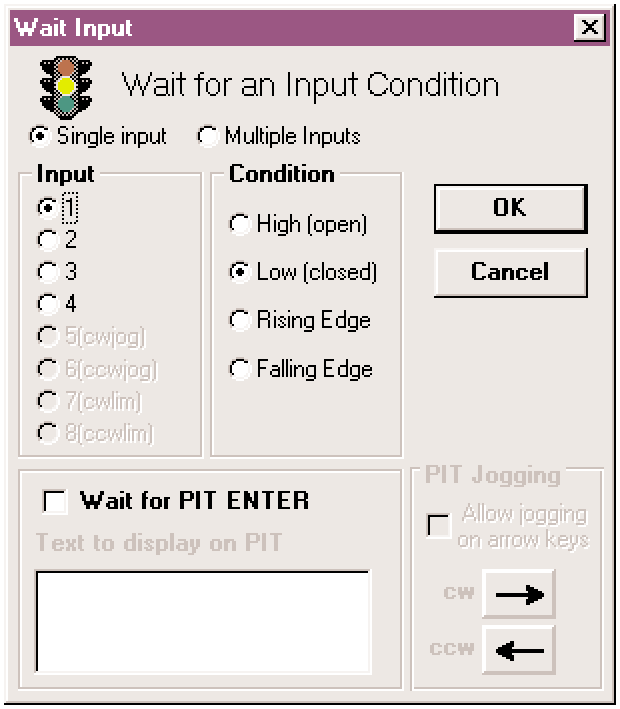
3 End Repeat Loop

Figure 55 - WAIT TIME INSTRUCTION

Loops can be nested to provide very long delays.

WAIT INPUT

Rarely does a motion controller operate completely on its own with no input from the outside world. The *WAIT INPUT* command is used to cause the MSC to wait before it starts a motion.



Wait Input

Wait for an Input Condition

● Single input ○ Multiple Inputs

Input

● 1
○ 2
○ 3
○ 4
○ 5(cwjog)
○ 6(ccwjog)
○ 7(cwlim)
○ 8(ccwlim)

Condition

○ High (open)
● Low (closed)
○ Rising Edge
○ Falling Edge

□ Wait for PIT ENTER

Text to display on PIT

PIT Jogging

□ Allow jogging on arrow keys

Figure 56 - WAIT INPUT DIALOG BOX

The *WAIT INPUT* instruction has two parameters (see Figure 56). The first is to specify the input. The second parameter is what kind of voltage condition to expect. The choices are:

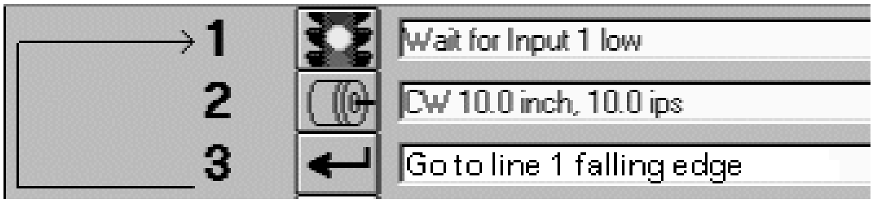
HIGH - Wait until the specified input is at a high voltage (no current) state. This is the state an input will be in if nothing is connected to it, so be careful using this condition. If a wire comes loose, it could cause an undesired motion.

LOW - Wait until specified input is at a low voltage (current flowing) state. This is the most popular configuration. A momentary contact switch (normally open type) connected between an input and ground creates this condition when the button is pressed.

RISING EDGE - Wait until the signal goes from low (current flowing) to high (no current). This is similar to the high condition, but the difference is important. For example: a signal into the MSC is one that will go high when motion is to occur. However, the signal remains high after the motion is complete, going low sometime later. Choosing high as the input condition will cause the MSC to complete the motion and start again because the input signal is still

high when it finishes the first move. If *RISING EDGE* is chosen, the MSC will wait for the input voltage to go low, then high (current flowing to no current) before moving.
FALLING EDGE - The opposite of *RISING EDGE*. MSC waits for input voltage to go high, then low (no current to current flowing).

Figure 57 illustrates a simple program that can be executed with the switch wired between *INPUT 1* and *GND*. Pressing the button will cause the motor to move 1000 steps.



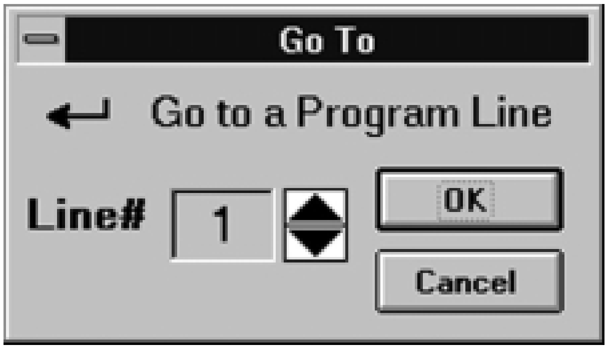
```
1  Wait for Input 1 low
2  CW 10.0 inch, 10.0 ips
3  Go to line 1 falling edge
```

Figure 57

With the optional Panel-mount Interface (PIT), the *WAIT INPUT* instruction can be used to display a message on the PIT screen and wait until the operator presses the *ENTER* button on the PIT keypad. Simply check the box marked "Wait for PIT ENTER" and type the message in the box marked "Text to display on PIT."

GO TO

The *GO TO* instruction is used to make the indexer-drive jump to another line in the program. To jump back to the beginning, place a *GO TO* instruction at the end of the program.



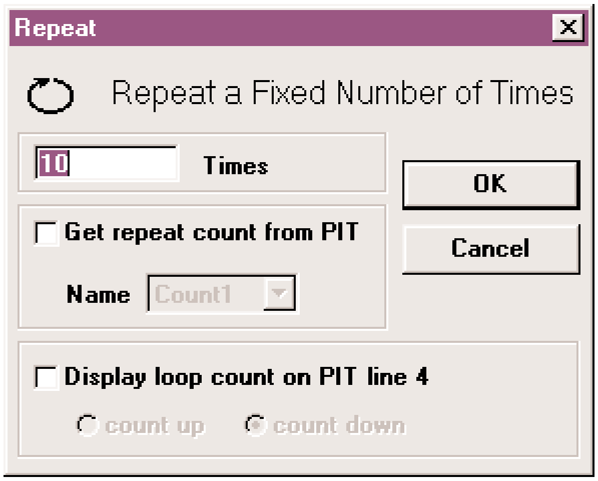
Line# 1

Figure 58 - GO TO DIALOG BOX

There is only one parameter to enter in a *GO TO* instruction: the line number to jump to. Click on the *SPIN* button to increase or decrease the line number.

REPEAT/END REPEAT

Sometimes the same thing is required several times. If that number is known in advance, Repeat loops allow the instructions inside the loop to be repeated up to 65535 times.



10 Times

□ Get repeat count from PIT

Name Count1

□ Display loop count on PIT line 4

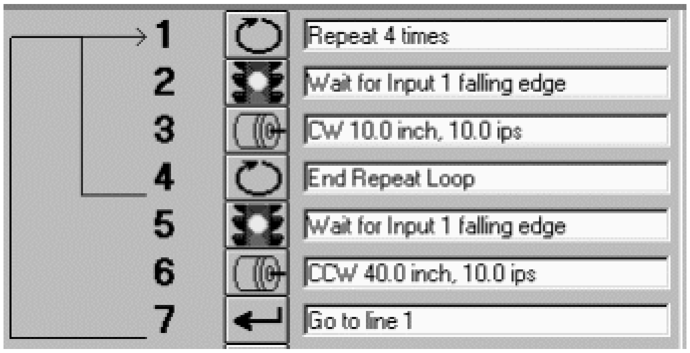
○ count up ● count down

Figure 59 - REPEAT/END REPEAT DIALOG BOX

Consider a system dispensing fluids into an array of containers. There are five rows and five columns of containers, each 100 steps away from the next. Each trigger command is to cause a move to the next position. After the fifth container is full, return to the first, 400 steps back.

A MSC controlling the X axis, or motion between columns, would be programmed as follows:

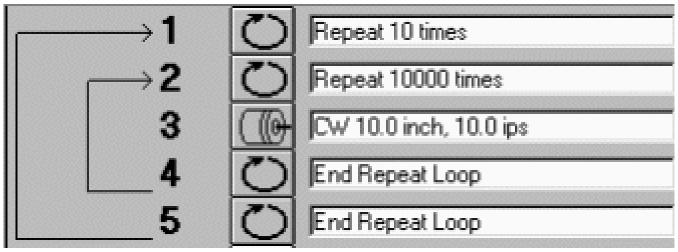
Refer to Figure 60. The program begins on line 1, entering the repeat loop. The next four times, the MSC will wait for the voltage at Input 1 to fall, then move clockwise 100 steps, taking the dispenser to the next container. After the fourth time, the MSC drops out of the loop into line 5. This time when Input 1 falls, the motor is moved 400 steps counterclockwise, returning to the original position.



```
1  Repeat 4 times
2  Wait for Input 1 falling edge
3  CW 10.0 inch, 10.0 ips
4  End Repeat Loop
5  Wait for Input 1 falling edge
6  CCW 40.0 inch, 10.0 ips
7  Go to line 1
```

Figure 60

Sometimes, it may be necessary to repeat something more than 65,535 times. For example, material is to be fed into a cut off knife and 100,000 pieces are to be created. The best solution is to set up 2 loops, one inside the other. The total number of cycles will be the number of repeats in the two loops multiplied together. 100,000 is 10,000 x 10, so one loop could be set for 10 and the other for 10,000 as shown in Figure 61.



1 Repeat 10 times
2 Repeat 10000 times
3 CW 10.0 inch, 10.0 ips
4 End Repeat Loop
5 End Repeat Loop

Figure 61

The *REPEAT* instruction can also use data gathered and stored by an *PIT PROMPT* instruction as the loop count. Check the box marked “Get repeat count from PIT” and select a variable from the list. For example, put an *PIT PROMPT* in the program to ask for the number of parts to be processed and save that data as *COUNT1*. Then set up the *REPEAT* instruction to get the repeat count from the PIT variable *COUNT1*.

The PIT can also display the loop count. Check Display on the dialog box and select Count up or Count down.

NOTE: If an input instruction is used to exit a Repeat Loop a Reset Repeat Loop instruction should be used to reset the loop. Otherwise the loop resumes where it left off.

RESET REPEAT LOOP

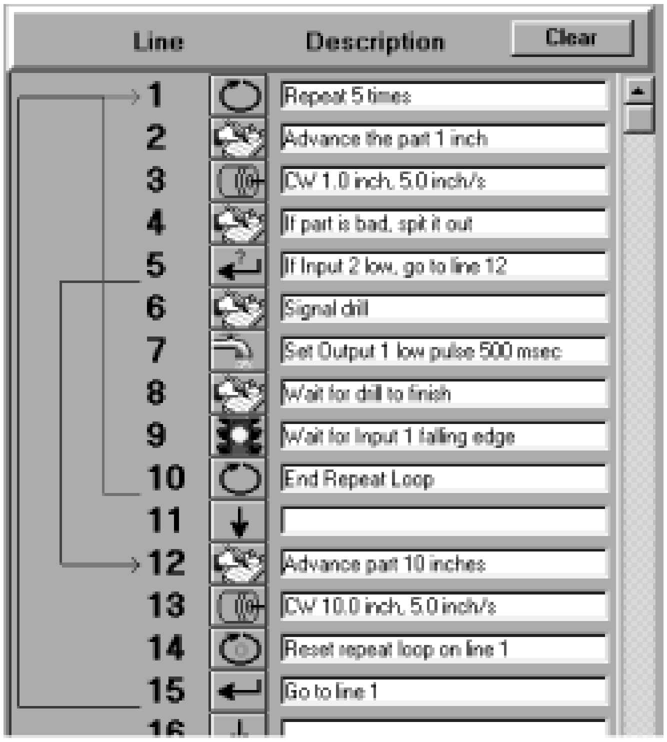
Forces a Repeat Loop to reset its counter if it’s been terminated by an If Input or Feed to Sensor with Safety Distance instruction.

Sometimes it is necessary to leave a repeat loop before it is completed. If, for example, a repeat loop is set up to fill 100 bottles with fluid. If the reservoir runs dry, the loop is exited by putting an ***IF INPUT*** instruction inside the loop, triggered by a fluid sensor. The If Input could branch outside the loop, to a PIT Prompt telling the machine operator to refill the reservoir.

Suppose that 60 bottles have been filled, with 40 remaining. For the loop to “pick up where it left off”, simply branch back to the beginning of the loop (to the Repeat instruction) after the operator finishes re-filling, and the loop will automatically fill the remaining 40 bottles.

On the other hand, consider a program to drill holes in parts, and each part gets five holes. The step motor is used to advance the part by 1 inch for each hole. So, there is a repeat loop with a count of 5. Along comes a bad part, detected after the 3rd hole is drilled. The loop is exited with 2 counts remaining.

If for the next part the program re-enters the loop again without reset, the next part will only get two holes drilled into it. What is wanted is for the loop to reset itself. For this, use the Reset Repeat Loop instruction as shown below.



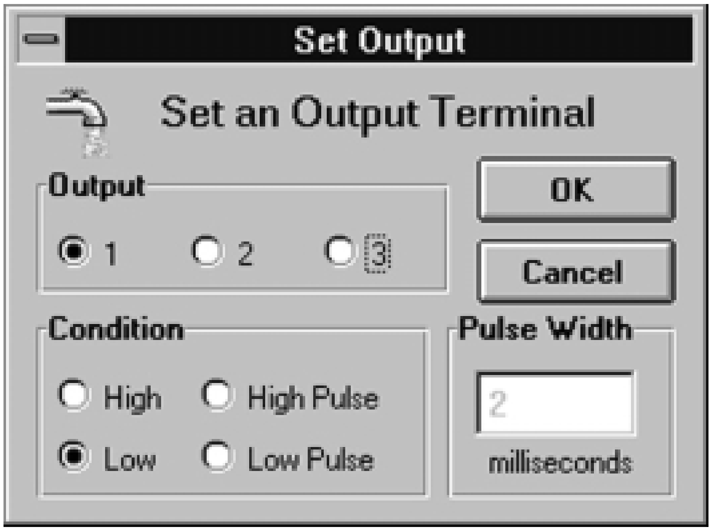
Line	Description
1	Repeat 5 times
2	Advance the part 1 inch
3	CW 1.0 inch, 5.0 inch/s
4	If part is bad, spit it out
5	If Input 2 low, go to line 12
6	Signal drill
7	Set Output 1 low pulse 500 msec
8	Wait for drill to finish
9	Wait for Input 1 falling edge
10	End Repeat Loop
11	
12	Advance part 10 inches
13	CW 10.0 inch, 5.0 inch/s
14	Reset repeat loop on line 1
15	Go to line 1
16	

Figure 62

SET OUTPUT

This command allow the choice of one of the three outputs and to put a voltage signal on it. For a detailed description of the circuitry and connections, see “Wiring Inputs" and "Wiring Outputs” sections of this manual.

There are four choices of output conditions (see Figure 63).



Set an Output Terminal

Output

● 1 ○ 2 ○ 3

Condition

○ High ○ High Pulse

● Low ○ Low Pulse

Pulse Width

2

milliseconds

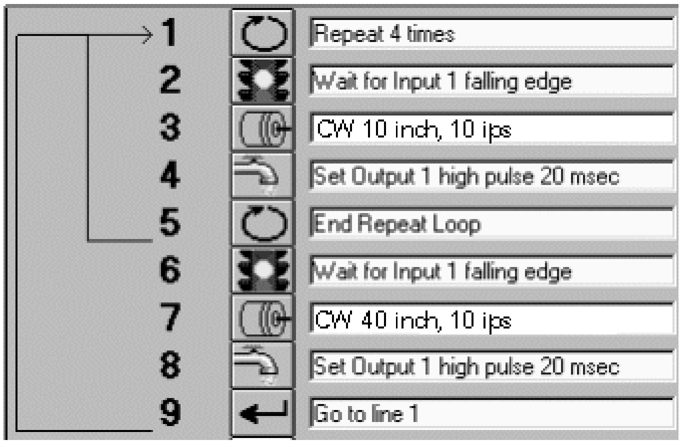
Figure 63

HIGH - Makes the photo transistor open. In circuits where the “-” output pin is grounded, and the “+” pin is pulled up, this causes a high voltage to appear on the “+” pin.
LOW - Makes the photo transistor close. In circuits where the “-” output pin is grounded, this causes a low voltage to appear on the “+” pin.
HIGH PULSE - Makes the photo transistor open for a specified amount of time (2 to 500 milliseconds)
LOW PULSE - Makes the photo transistor close for a specified amount of time (2 to 500 milliseconds)

At power-up, the MSC sets all 3 programmable outputs high (open circuit).

As an example of the appropriate use of the *SET OUTPUT* instruction, consider a system dispensing fluids into an array of containers. Each time the MSC moves to a new position, it should tell the dispenser it has arrived. This can be accomplished with a high pulse, but the choice would depend on the kind of signal the dispenser wants to see in order to be activated.

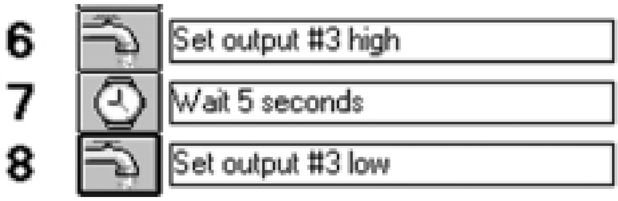
There are five rows and five columns of containers, each 100 steps away from the next. Each trigger command is to cause a move to the next position. After the fifth container is full, return to the first, 400 steps back. After adding a *SET OUTPUT* instruction after each *FEED TO LENGTH,* the program is as shown in Figure 64.



1 Repeat 4 times
2 Wait for Input 1 falling edge
3 CW 10 inch, 10 ips
4 Set Output 1 high pulse 20 msec
5 End Repeat Loop
6 Wait for Input 1 falling edge
7 CW 40 inch, 10 ips
8 Set Output 1 high pulse 20 msec
9 Go to line 1

Figure 64

There may be occasions when a long pulse is required. This can be done by combining two Set Output commands with a Wait Time. The instructions shown in Figure 65 will produce a high pulse of 5 seconds on Output 3.



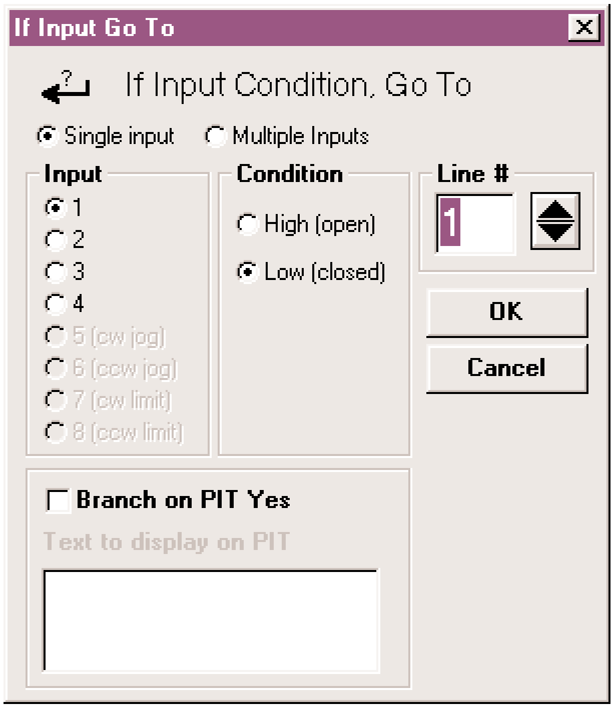
6 Set output #3 high
7 Wait 5 seconds
8 Set output #3 low

Figure 65

IF INPUT GO TO

This instruction allows the MSC to make decisions based on input signals.

Refer to Figure 66. Choose an input terminal for the instruction to check. The MSC must also be told what signal condition to look for. Finally, the line number that the instruction will jump to must be set if the input condition occurs.



If Input Go To

If Input Condition, Go To

● Single input ○ Multiple Inputs

Input: ● 1 ○ 2 ○ 3 ○ 4 ○ 5 (cw jog) ○ 6 (ccw jog) ○ 7 (cw limit) ○ 8 (ccw limit)

Condition: ○ High (open) ● Low (closed)

Line #: 1

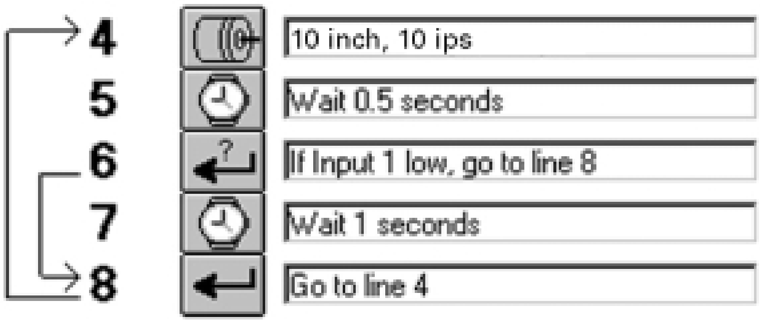
□ Branch on PIT Yes

Text to display on PIT

Figure 66 - IF INPUT DIALOG BOX

The *IF INPUT* instruction is included for three reasons:

1. It allows skipping part of the program based on an external condition. For example, the MSC's task is to feed parts. Normally the MSC waits only a half second before feeding the next part, because that's how long it takes for the saw to cut the part. However, parts are sometimes made of a different material that takes longer to cut (aluminum vs. steel, maybe). On the days steel parts are run, it would be advantageous to be able to flip a switch and change the delay between parts to 1.5 seconds. Figure 67 shows the program required to accomplish this.

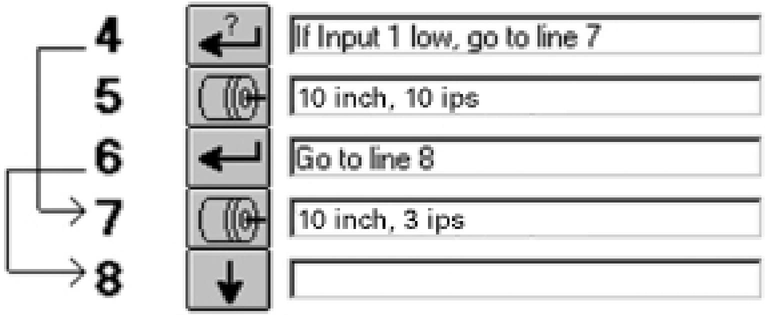


```
4  10 inch, 10 ips
5  Wait 0.5 seconds
6  If Input 1 low, go to line 8
7  Wait 1 seconds
8  Go to line 4
```

Figure 67

The program feeds a part during Step 4. Step 5, then makes it wait a half second. If the switch connected to *INPUT 1* is closed (low voltage signal state), the program jumps to Step 8, skipping the extra 1 second of delay. If the switch is open, the delay occurs. The switch's open circuit position can then be marked as "Steel" and the closed position as "Aluminum."

2. It allows a parameter to be changed such as distance or speed based on an input. Consider the example above. How will the cut-off saw know how fast to go when the switch is set for "Aluminum" or "Steel?" The same switch that controls the movement of the saw could be wired to the MSC. This time, the program is written as shown in Figure 68.

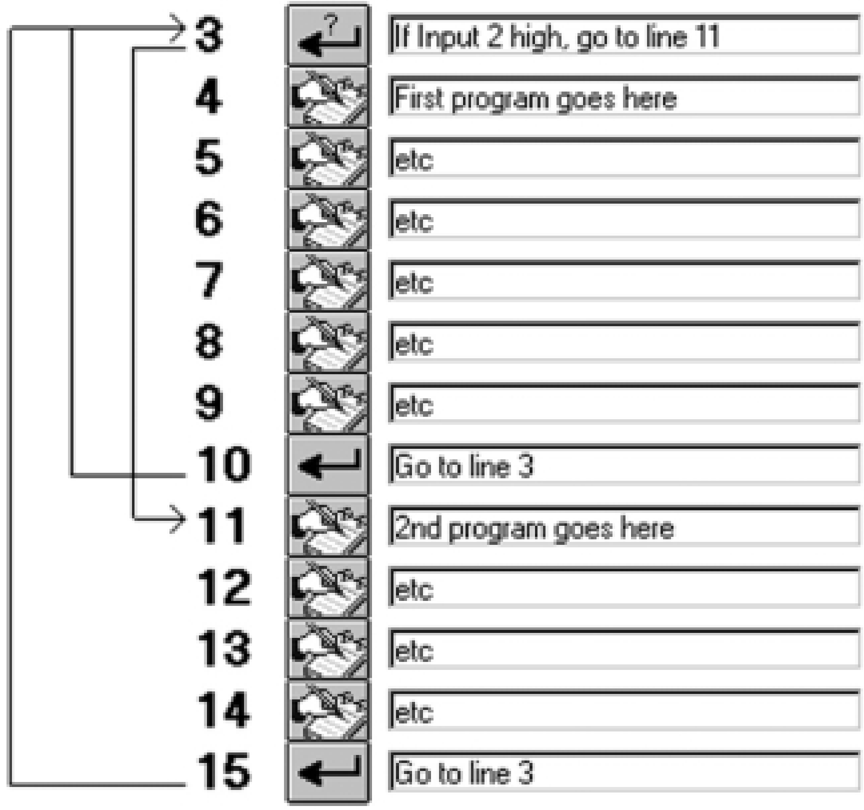


```
4  If Input 1 low, go to line 7
5  10 inch, 10 ips
6  Go to line 8
7  10 inch, 3 ips
8
```

Figure 68

When the MSC gets to Step 4, it will look at the “Steel/Aluminum” switch. If the signal is low (Aluminum), it jumps to the *FEED TO LENGTH* program at Step 7, which moves the saw at 3 inches/sec. If the switch is high, the program does not jump, but instead executes the *FEED TO LENGTH* at Step 5, which feeds at 1.0 inch/sec. The MSC then jumps past the second feed because of the *GO TO* instruction in Step 5. **NOTE:** To prevent moving the saw twice, do not forget the *GO TO*.

3. It allows the use of multiple programs within the 100 line program space. It may be that the system is to do is two completely different things depending on an input. If each of these tasks requires 4 instructions, the program should look like Figure 69.



3 If Input 2 high, go to line 11
4 First program goes here
5 etc
6 etc
7 etc
8 etc
9 etc
10 Go to line 3
11 2nd program goes here
12 etc
13 etc
14 etc
15 Go to line 3

Figure 69

Depending on the state of Input 2, the program will either execute lines 4–9 or lines 11–15. Either way, the program ultimately returns to line 3 to check the condition of the switch again.

The optional Panel-mount Interface (PIT) can be used as the decision making input of an *IF INPUT* instruction. Simply check the box marked “Branch on PIT YES” and type in the message the operator is to see. If the operator presses the *YES* button, the drive will jump to the line specified in the line number box. If the operator presses *NO,* the program moves on to the next line.

Notes:

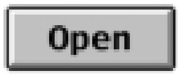
Program Documentation 10

Program Documentation

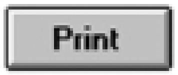
SAVE, OPEN, PRINT & QUIT

In addition to exchanging programs with the MSC, the programming software can also save & load programs using the hard drive, and can print hard copies of programs using the printer.

SAVE - allows saving a program to the hard drive. A file dialog box asks for a name for the program. Enter up to 8 characters, not including the suffix ".SI5". If a suffix is not added, the software will automatically add ".SI5" to the file name. The characters in the filename must conform to the usual DOS/Windows 3.1 rules. The safest approach is to use only letters and numbers in the filename, and to avoid special characters like "?" or "\".



OPEN - provides a dialog box showing all the ".SI5" files on the drive. Click to select one, then click OK to load it. .



PRINT - makes a hard copy of the program on any printer attached to the computer and installed in Windows.

QUIT - exits the MSC programming software and returns to the Windows Program Manager.

Notes:

Technical Specifications 11

Specifications

Programmable Inputs	8, bidirectional, optically isolated, 5 - 24 Vdc
Programmable Outputs	3, optically isolated, sourcing or sinking
Input Voltage (AC Line)	110 or 220 VAC, 50 - 60 Hz (switch selected)
Input Current (AC Line)	0.2A max at 110VAC
Input Fuses (AC Line, L1 and L2/N)	0.1A slow blow fuse
5V output	±5%, 100 mA max, self resetting fuse, not isolated from internal circuitry
24V output	±5%, 100mA max, self resetting fuse, isolated from internal circuitry
Input impedance - IN 1 - 4, CW Jog, CCW Jog	2200 Ohms
Input impedance - limits	2200 Ohms
Output current (Out1 - 3)	100 mA max
Output voltage (Out1 - 3)	24 Vdc max
Motion update rate	12,800 Hz
Output voltage, logic high (STEP & DIR)	2.5V min, 3.4V typ, with 20 mA load
Output voltage, logic low (STEP & DIR)	0.5V max, 0.3V typ, with 20 mA load
Output frequency (STEP)	50 Hz to 2.54 MHz
Output duty cycle (STEP)	50%
Weight	TBD
Ambient temperature range	0 - 50 °C

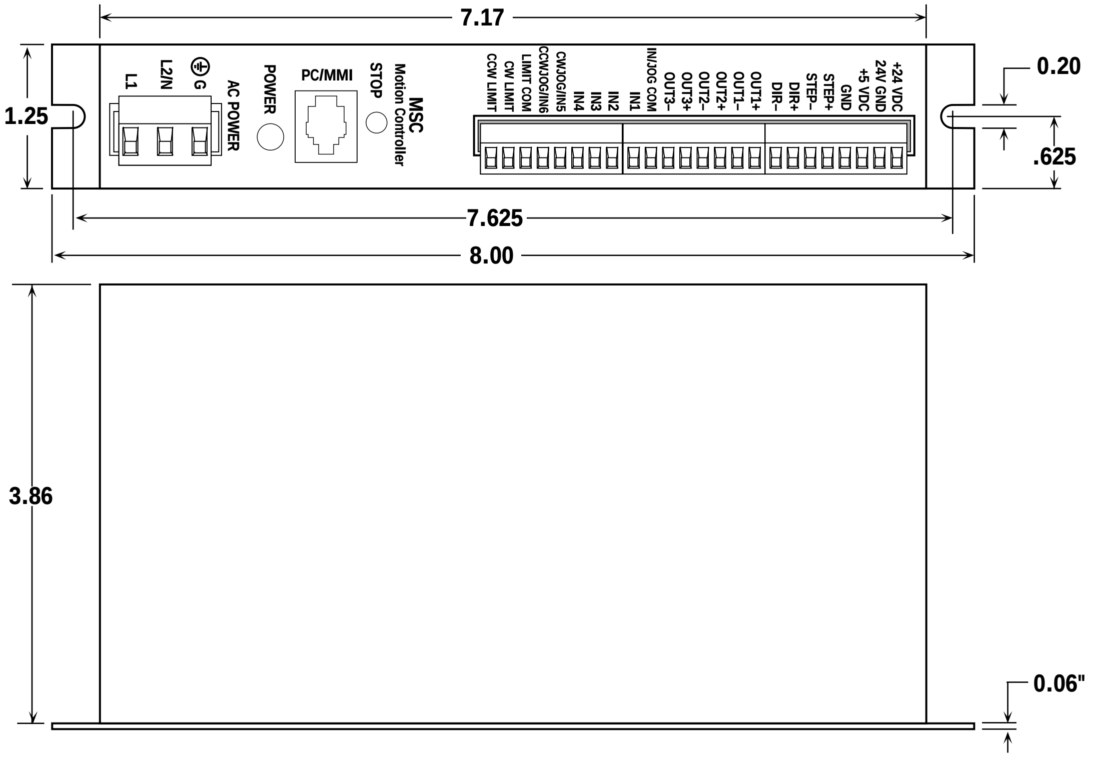
Use and Care

- Never allow metal particles near the drive.
- Never put the drive where it can get wet.
- Never block the fins of the heat sink or the vent holes.
- Never use the drive in a space where there is no air flow, or where the ambient temperature exceeds 50 °C.

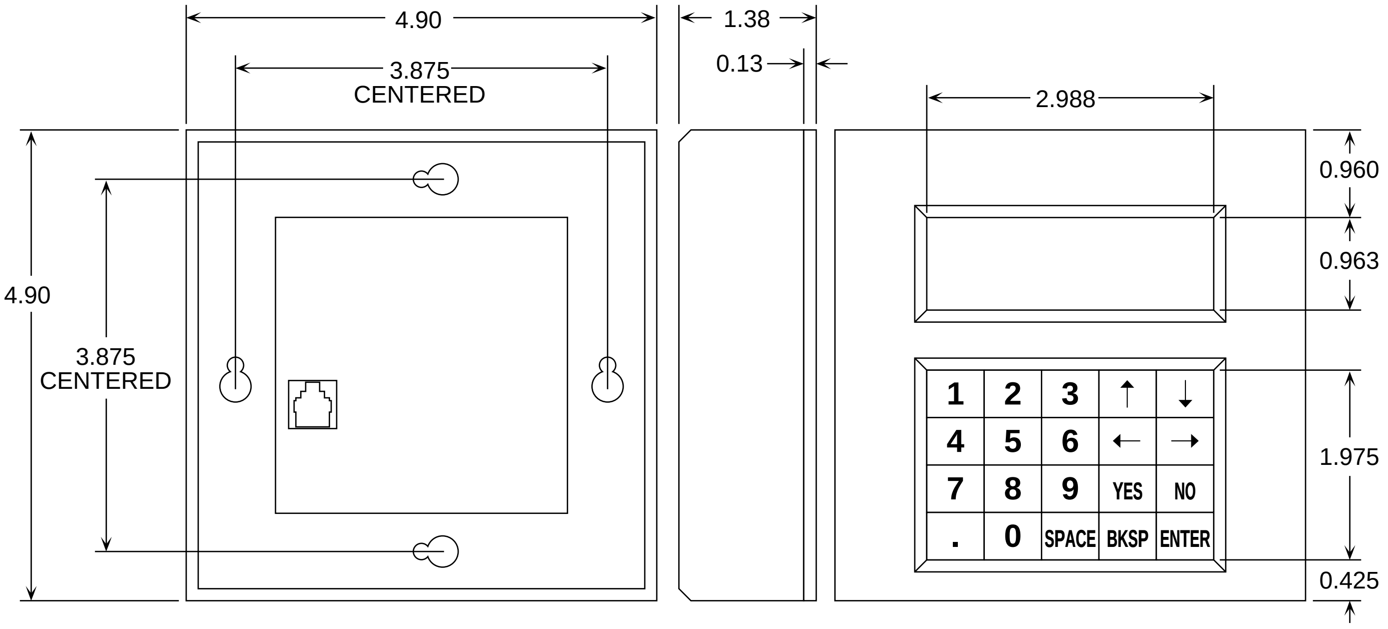
Notes:

Dimensions

MSC CONTROLLER *Figure 63*



OPTIONAL PIT *Figure 63*



Notes:

Mounting Information 13

Mounting the MSC

The MSC has two 0.2-inch-wide mounting slots in the back panel to accommodate #10 screws (not included). The MSC is lightweight (less than 4 lbs.) and is not a significant heat source. Observe the following mounting precautions:

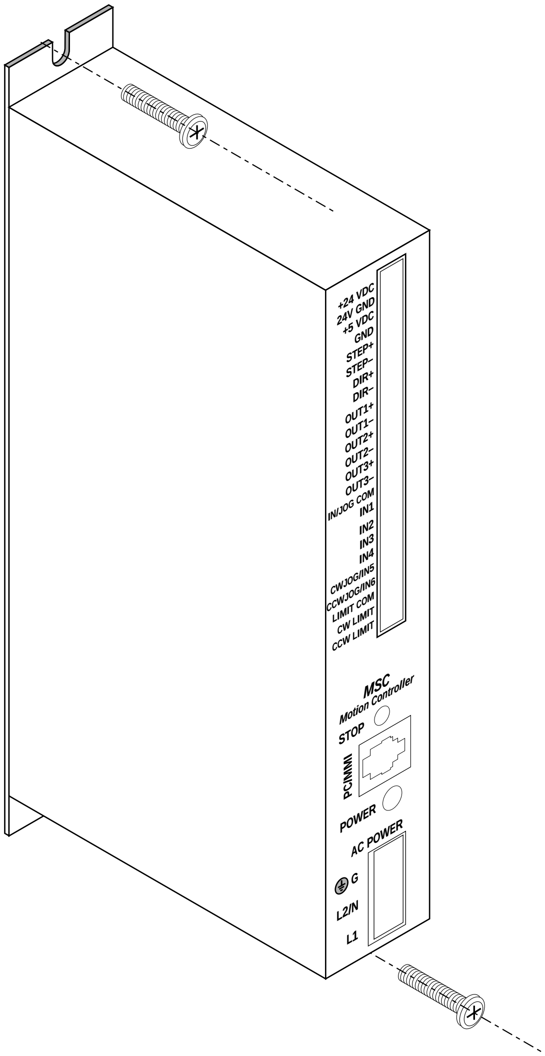


Figure 64 - MOUNTING THE MSC

⚠ **Caution!** Observe the following mounting precautions:

- ***Mount in a clean dry environment.***
- ***Do not mount above or adjacent to heat or emi-producing devices.***
- ***Observe the operating temperature limits***

Mounting the Optional PIT

There are two ways to mount the PIT in the application. In either case, connect the PIT to the MSC with the programming cable. **NOTE:** The adapter plug is not needed. The PIT has the same telephone style connector as the MSC.

Depending on how the PIT and cable are mounted, it may be difficult to remove the cable from the back of the PIT. If this is the case and it is necessary to reprogram the MSC, any telephone line cord can be used as a programming cable. Be careful not to lose the adapter plug that connects the telephone cord to the *COM* port of the computer. The adapter is a only available from the factory.

FLUSH MOUNTING

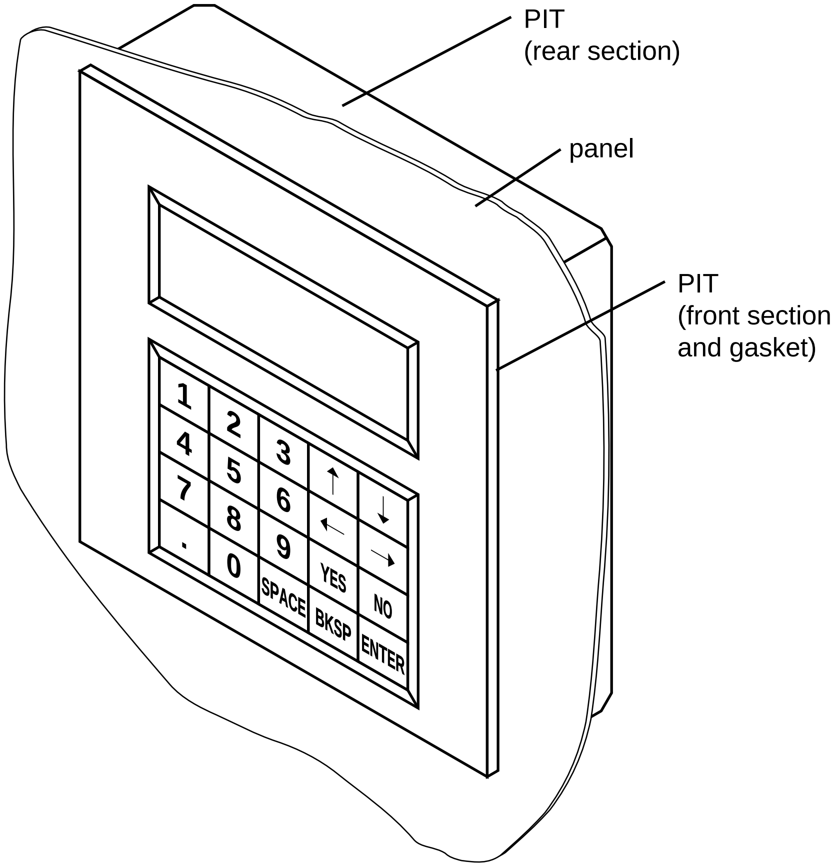


Figure 65 - FLUSH MOUNTING THE PIT

Refer to Figure 65. After removing the PIT from the shipping carton, notice that it has two parts. The first is a fairly thin section that contains the keypad, display and some circuit boards. The other part is thicker and contains the telephone jack and a cable that connects to the thin part.

When flush mounting the PIT in a panel, only the thin section will stick out from the panel - the large portion mounts behind the panel. It is necessary to cut a precise section from the panel. A cardboard template is included in the box for this purpose.

If the PIT is to be dust proof and watertight, the black rubber gasket must be placed between the thin part of the PIT and the panel. Assemble the two halves using the eight small screws.

SURFACE MOUNTING

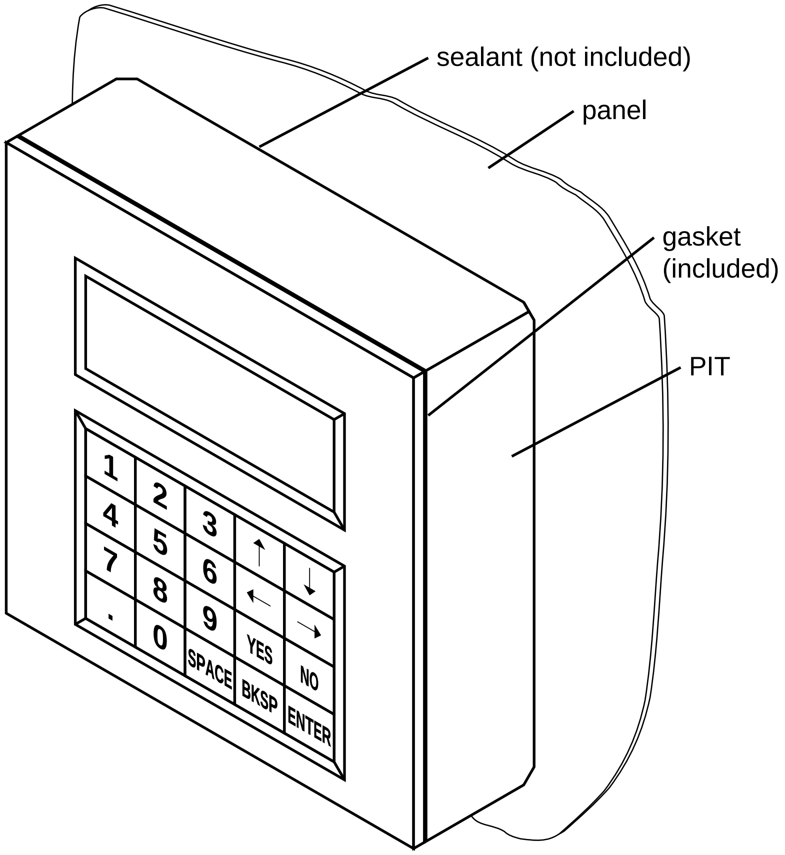


Figure 66 - SURFACE MOUNTING THE PIT

An easier way to mount the PIT is to bolt the two halves together ahead of time, using the eight small screws. If the PIT is to be dust proof and watertight, put the black rubber gasket between the two halves before screwing them together.

Next, cut a hole in the panel for the cable that runs between the PIT and the MSC. The hole must be at least 5/8” in diameter for the connector to fit through. There must be two holes that line up with the big mounting holes in the PIT. Figure 63 shows the location of the big mounting holes. When mounting the PIT to the panel, some kind of sealant will be required to keep dust and liquid out. Silicone or latex caulking is appropriate, or make a gasket from a sheet of compliant material such as rubber or RTV.

TOL-O-MATIC, INC.
3800 County Road 116
Hamel, MN 55340
763.478.8000 Telephone
763.478.8080 Fax
http://www.tolomatic.com

Axidyne®