ELECTRIC LINEAR MOTION PRODUCTS

MSCLDC Closed Loop Dc Positioning System

User Manual

TOL-O-MATIC, INC
Excellence in Motion®

Contents

Axidyne Closed Loop DC Positioning System

Introduction

System Overview

Connections, Indicators, Potentiometers

System Wiring

Wiring Inputs

Wiring Outputs

Tol-O-Motion MSCLDC Programming Software

PIT Interface

Programming Tools

Programming Instructions

Technical Specifications

System Installation

Dc Motors

Introduction

About This Manual...

The Axidyne Closed Loop DC Controller/Drive (MSCLDC) may be combined with any of the Tol-O-Matic brush dc motors to provide flexible solutions for linear motion control applications that require motion profiling and positioning. The motors are chosen to match the Axidyne actuator product family. With simple operator inputs the MSCLDC will function as a stand-alone motion controller/drive.

This manual provides the information necessary to configure, install, and program these Axidyne system components for the selected application.

If you have difficulty configuring, installing, or programming your system, please contact your local distributor for help, or call Tol-O-Matic at 1-800-328-2174.

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 when 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.

Overview

Typical electronic linear motion control systems consist of the following elements:

Motor: Provides the torque and speed necessary for an actuator to meet application requirements.

Drive: Converts the signals received from the controller or PLC to actually move the motor. In addition, the drive must convert the local power source (typically 115 V.a.c., 60 Hz) to the power required by the motor. The power ratings (Watts) of the motor and the drive must match the peak and RMS requirements of the application.

Controller: Features I/O connections to receive inputs from a programmable logic controller (PLC) or other operator interface and convert them to output signals to the drive module to properly control the motor and to achieve the required motion profile(s).

Operator Interface: An optional device used by the system operator to program or signal the controller remotely.

The performance of an electric linear actuator system is determined by the type of control system used with the actuator (i.e. dc open loop or brush servo, stepper, or ac brushless servo). In general, dc systems represent a low-cost, mature technology easily applied to meet basic linear motion needs.

MSCLDC Controller/Drive System

The MSCLDC closed-loop dc control comprises a programmable controller/indexer module together with a brushed dc servo drive module in an enclosed configuration. When implemented with a dc brush servo motor/encoder this system provides programmable proportional and derivative closed loop control in a position controlling mode. An optional operator interface (PIT) is available.

Tol-O-Matic offers a range of dc brush servo motors with torque speed characteristics to match the Axidyne family of actuators.

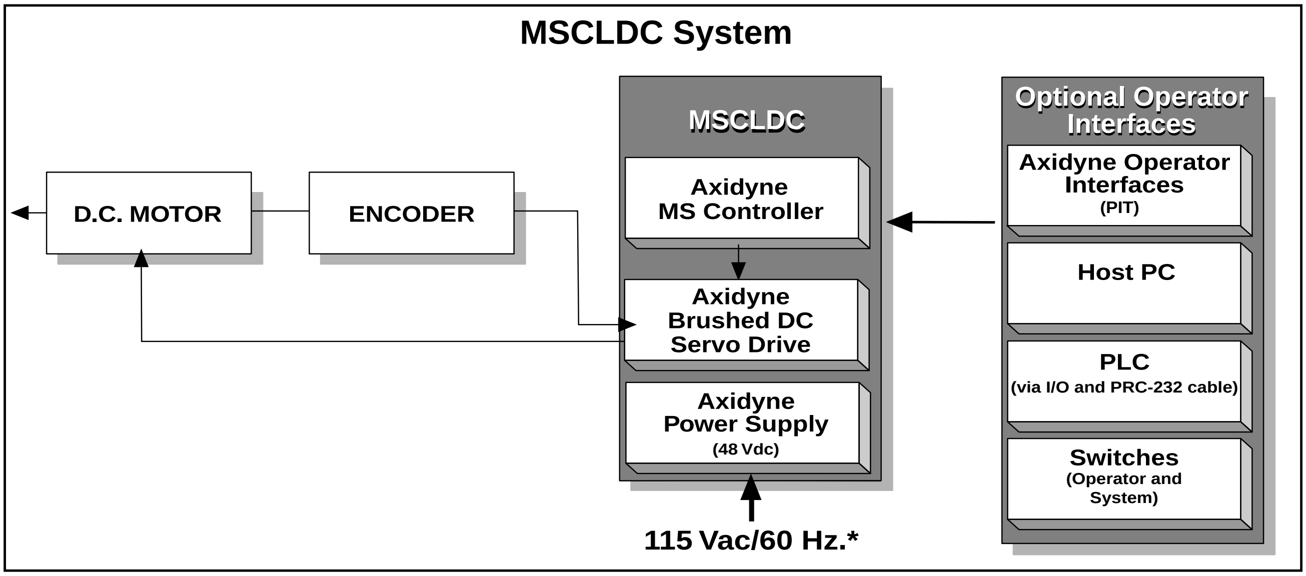

MSCLDC System
D.C. MOTOR
ENCODER
MSCLDC
Axidyne
MS Controller
Axidyne
Brushed DC
Servo Drive
Axidyne
Power Supply
(48 Vdc)
115 Vac/60 Hz.*
Optional Operator
Interfaces
Axidyne Operator
Interfaces
(PIT)
Host PC
PLC
(via I/O and PRC-232 cable)
Switches
(Operator and
System)

Figure 1

MSCLDC Drive Module (DMCLDC)

The MSCLDC Drive module (DMCLDC) consists of a brushed dc drive and a 48 Vdc power supply. It has inputs for a pulse train and direction signal to be provided by the stepper controller module.

Tol-O-Matic's brushed dc drive utilizes a P.D. (proportional and derivative) feedback loop for closing the loop around the motor. The brushed dc drive module has three main functions: 1) convert 115V/60 Hz. supply power to the voltage and current level the motor requires to produce the desired speed and torque; 2) respond appropriately to the stepper control pulse train and an encoder feedback input to ensure that the desired motion profile is achieved; and 3) provide position holding torque. Using the latest pulse-width-modulation (PWM) drive technology, Axidyne brushed dc drives provide smoother, quieter motor operation at low speed, prolonged motor brush life, reduced heat build-up and reliable repetitive solid-state reversing.

Brushed DC Drive

Features

- *10 Amp continuous, 15 Amp peak power supply rating.*
- *Position feedback accomplished via a two channel (A & B) differential incremental encoder.*
- *Proportional + Derivative (PD) feedback loop for closing the loop around the motor.*
- *Final position error of +/- 2 encoder counts (typical), with repeatability typically +/- 1 encoder count.*
- *Fault status output via a solid state relay, Form "C" output*
- *Position error (+/- 127 encoder counts) and current trip faults.*
- *Current trip point potentiometer for setting torque fault limit.*
- *No motor holding torque while in a fault condition.*
- *Fault condition indication via 2 L.E.D's to indicate current trip and position error faults for field diagnostics.*
- *Opto-isolated external I/O power (Vdc I/O).*
- *Potentiometer for Damping and Gain adjustment.*
- *Dedicated reset input for clearing fault condition.*
- *Five catalog dc brush servo motors in three frame sizes, with encoders.*

MSCLDC Controller Module

The MSCLDC controller (module ms) provides easy programmability of the speed and direction pulse train input to the closed loop dc drive module. The controller provides automatic homing commands, distance (steps) or sensor based moves, and programmable acceleration and deceleration rates to maximize performance. Waits, dwells and jumps are all programmable.

Features

- ***1,000 step per revolution resolution***
- ***Windows® based software for easy set up and PC programming via RS232 (RJ11) cable***
- ***Powerful, flexible, common language commands***
- ***Optional PIT (Panel-mount Interface) for insertion of variables***
- ***Two dedicated limit switch inputs***
- ***CW and CCW jog inputs***
- ***Four general purpose opto-isolated inputs***
- ***Three general purpose opto-isolated outputs***

Notes:

Overview

This section is intended to provide a comprehensive overview of the MSCLDC controller/drive inputs, outputs, indicators, connectors and adjustments. Details on system wiring and adjustments follow.

Ac Power Input

115 Vac/Neutral/Ground terminals for ac power.

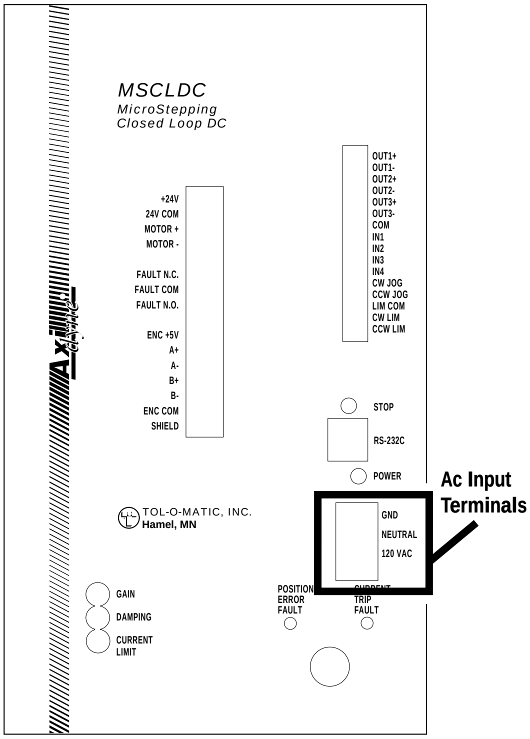


Figure 2 - AC INPUT LOCATION

Input and Output Connections

MOTOR AND ENCODER

The motor red and black leads are connected to Motor+ and Motor- respectively.

Wire Color Code	*Encoder Terminal*
Red	*+5V*
Black	*GND*
White	*A+*
Yellow	*A-*
Green	*B+*
Blue	*B-*

POWER SOURCE FOR INPUTS

+24V, 24 COM are terminals for an internal 24Vdc power supply for the opto-isolated inputs. This supply can be used when inputs are from switches, relays or other non-active devices.

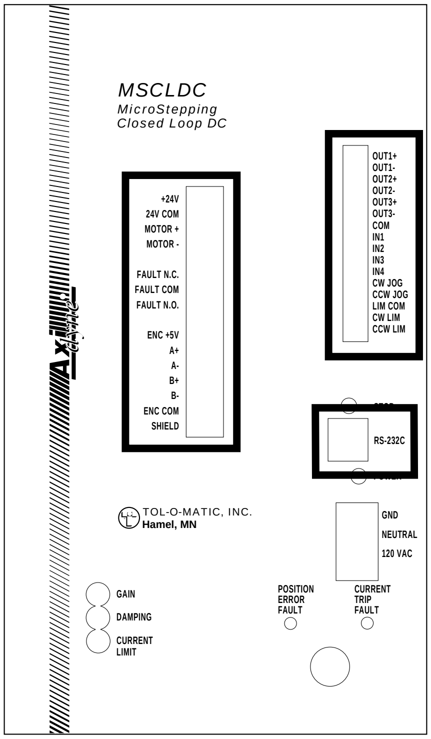


Figure 3 - INPUT AND OUTPUT LOCATIONS

LOGIC INPUTS 1, 2, 3, 4 AND COM

These are optically isolated inputs capable of sourcing or sinking 5-24V. They allow external events to control the flow of an internal program.

JOG CW, CCW

These are dedicated inputs for jogging the motor at a predesignated speed set in software. If not required for jog these inputs can be used as general purpose inputs additional to Inputs 1 through 4.

LIMITS CW LIM, CCW LIM, LIM COM

These optically isolated inputs are used in conjunction with motion range limit switches to prevent damage to the actuator system.

RS 232C

This is a telephone-style jack for connection of the RS232C line from the programming computer or the PIT (Panel-mount Interface) panel.

LOGIC OUTPUTS 1, 2, 3 (+ AND -)

These are optically isolated programmable outputs used for providing external signals of program status.

FAULT N.C., COM, N.O.

A form C solid state relay provides N.O. and N.C. contacts for remote indication of position error or current trips fault. Reset requires removing and reapplying ac input power.

Fault Indicators

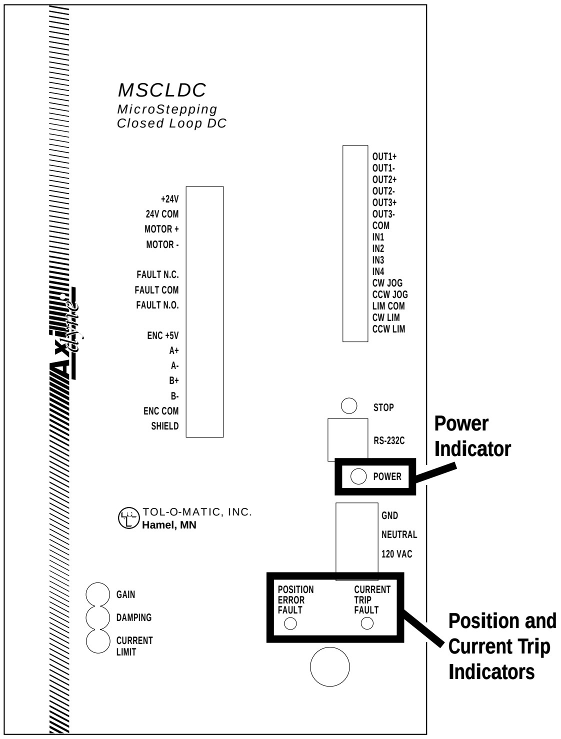


Figure 4 - INDICATOR LOCATIONS

POSITION ERROR LED

The position error detection circuit shuts off (faults) the drive if the position error exceeds the preset limit. The PD (Proportional & Derivative) control loop in the closed loop dc drive calculates error by comparing the commanded position of the motor with its actual position (provided via encoder feedback). If that error is greater than 127 encoder counts, then the position error will be active and will fault the drive, and light the LED.

OVERCURRENT TRIP LED

The overcurrent protects against overloading of the motor and actuator system while allowing for high starting torque for systems with high inertial load. The overcurrent detection circuit monitors motor current build-up above a set value at any time during operation.

POWER LED

The red Power LED is on when ac input is present and the controller is operating or ready to operate. The Power LED will flash if the STOP button is pressed and if the CW or CCW limits are activated.

Potentiometers

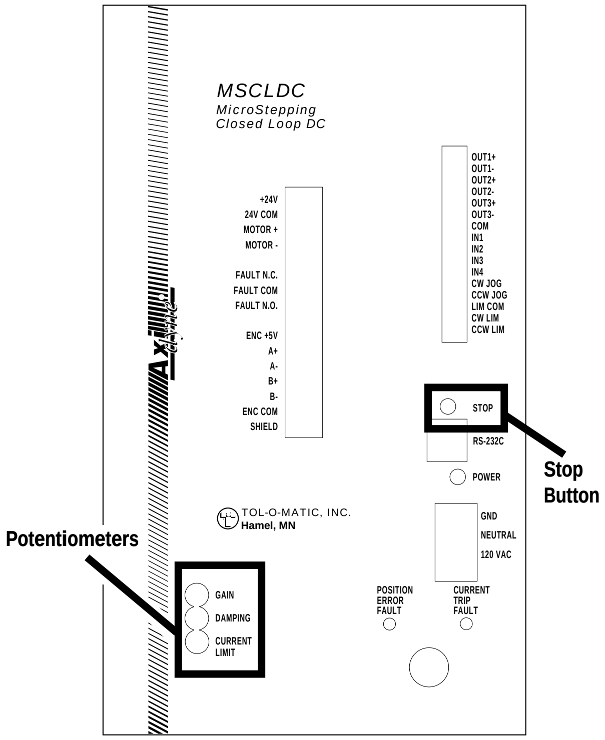


Figure 5 - POTENTIOMETER AND STOP BUTTON LOCATIONS

There are three adjustment potentiometers:

GAIN

Position error adjustment between the commanded and actual position feedback signal. Proportional gain is the response to the current value of the error signal. It is analogous to a spring constant, where the larger the value, the stiffer the spring. 25 turns this potentiometer (CW / CCW increases / decreases the system gain respectively).

DAMPING

System settling time adjustment between the system's natural and operating frequency. This is generally considered the damping of perturbations in position and velocity. It is analogous to a shock absorber in a automobile. 25 turns this potentiometer (CW / CCW increases / decreases the system damping respectively).

CURRENT TRIP

Used to adjust the maximum current allowed before faulting the drive. This allows protection against overloading of the motor and actuator system while allowing high starting torque for systems with high inertial load. The overcurrent detection circuit monitors motor current build-up above a set value at any time during operation. 25 turns this potentiometer (CW / CCW increases / decreases the current trip level respectively). This pot should be set after the system is tuned and programmed.

See page 26 for information about setting these potentiometers.

Stop Button

There is a button marked “STOP” on the front panel of the MSCLDC. This button can be used to interrupt motion at any time. After pressing the *STOP* button, the motor will stop. The front panel *POWER* LED will then flash until the MSCLDC is reset by removing and restoring the power. If the MSCLDC is connected to a computer running the Windows Programming Software, the software will alert the user on screen to the condition, and provide the option to reset the drive from the computer.

Grounding

In general, all electrical components and enclosures must be connected to earth ground through a grounding electrode conductor to provide a low impedance path for ground fault or noise-induced currents. The MSCLDC enclosure features internal grounding for operator safety.

A single-point grounding setup is recommended, and all earth ground connections must be continuous and permanent. Prepare all other components and mounting surfaces prior to installation so that good electrical contact is made between the component enclosure and the mounting surface. Remove the paint from equipment surfaces where the ground contact will be bolted to a panel, and use star washers to ensure solid bare metal contact.

Wiring AC Power

To connect the MSCLDC to an ac power supply, consult the wiring diagram in Figure 6.

☠ ***WARNING: All ac power must be disconnected prior to installation of wiring. Failure to observe safe working practices when installing or servicing this equipment can expose you to dangerous voltages.***

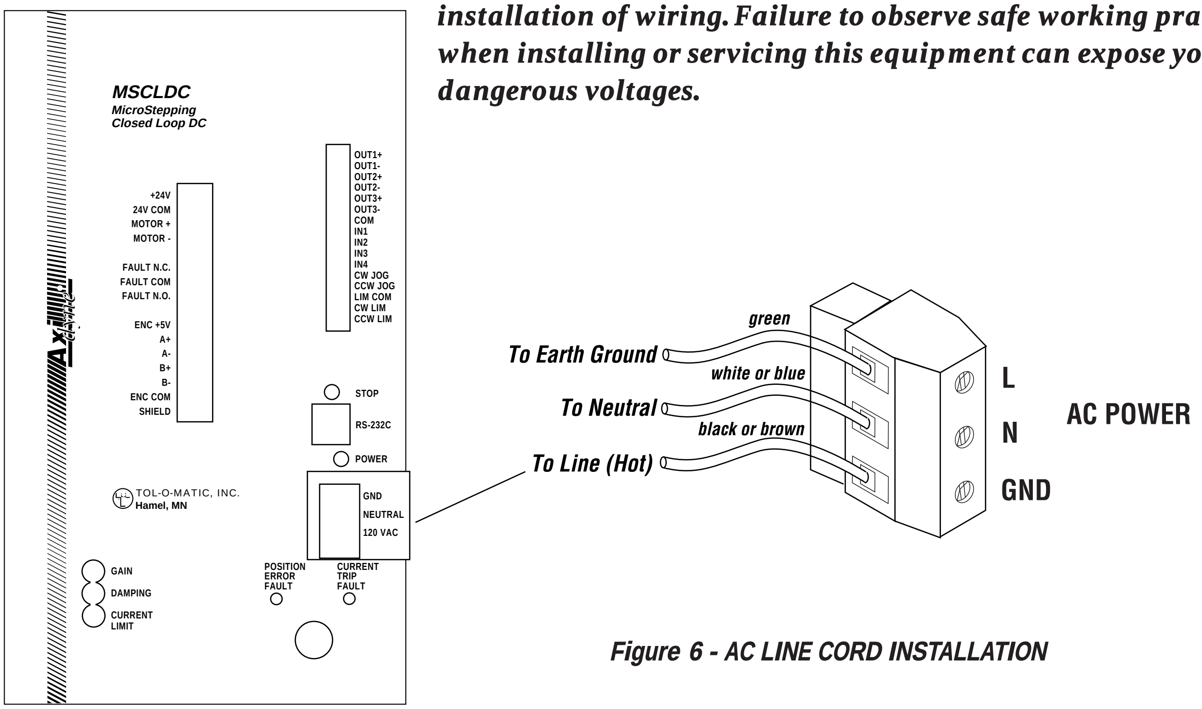


Figure 6 - AC LINE CORD INSTALLATION

Encoder/Motor Connection

ENCODER

There are six wires connecting the Encoder to the MSCLDC Module. Terminal connections are for +5 Vdc to power the encoder, encoder GND, and the other four are connected to the A and B channels, and serve as the encoder feedback to the closed loop dc drive module. Use of shielded cable is strongly recommended with the shield connected at one end only.

Wire the Tol-O-Matic encoder to the module as follows:

Wire Color Code	***Encoder Terminal***
Red	*+5V*
Black	*GND*
White	*A+*
Yellow	*A-*
Green	*B+*
Blue	*B-*
Orange	*I+ (not used)*
Brown	*I- (not used)*

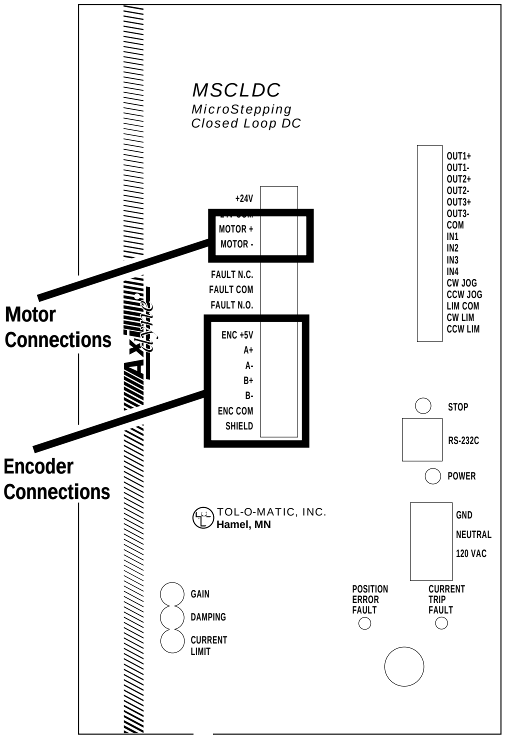


Figure 7 - MOTOR AND ENCODER CONNECTIONS

MOTOR

Connect the Red and Black motor leads to terminals 2 and 3 (Motor + and Motor -) respectively.

Fault Relay

Fault N.C., Fault COM, Fault N.O. are the terminals of the Form C contact of a solid state relay which changes state at error or overcurrent faults and can be used to provide remote indication of drive shut down.

Reset requires removal and restoration of ac power.

Computer Connection

TO CONNECT TO THE COMPUTER FOR PROGRAMMING

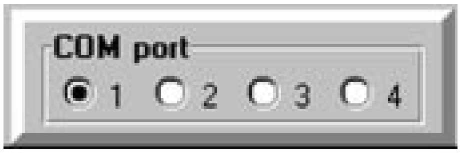
Locate the MSCLDC within 6 feet of the computer.

The MSCLDC was shipped with a 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 COM1 serial port of the PC. Secure the adapter with the screws on the sides. If the COM1 port on the PC is already allocated, the COM2 port may be used. On some PCs, 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.

The MSCLDC was also shipped with a 7 foot telephone line cord. Plug one end into the adapter just attached to the PC, and the other end into the RS232 jack on your MSCLDC.

⚠ Never connect the MSCLDC to a telephone circuit. It uses the same connectors and cords as telephones and modems, but the voltages are not compatible.

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 connection.



COM port

● 1 ○ 2 ○ 3 ○ 4

Figure 8

Programming note: Always open and run the programming software on the PC before applying power to the MSCLDC.

Wiring Inputs

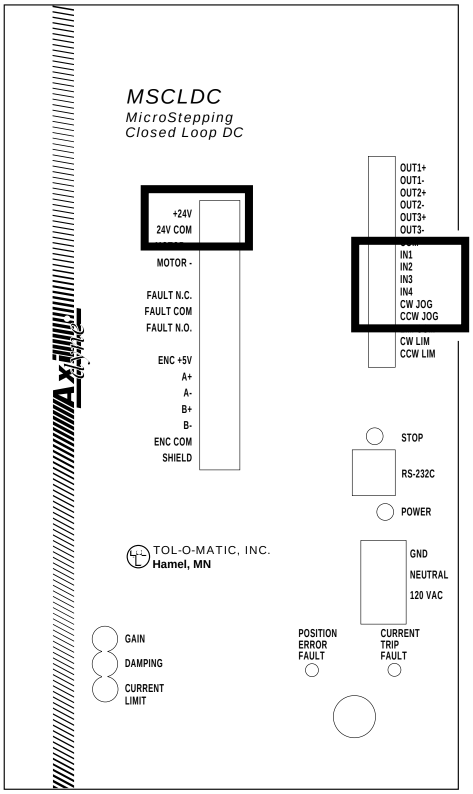


Figure 9 - WIRING INPUT LOCATIONS

The maximum voltage that can be applied to an input terminal is 24 volts DC.

The MSCLDC 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 a PLC, power should be available from the PLC or TTL power supply. If using relays or mechanical switches, the MSCLDC built-in 24 volt power supply can be used.

NOTE: If current is flowing into or out of an MSCLDC 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.

Figures 11 through 14 show how to connect MSCLDC inputs to various devices.

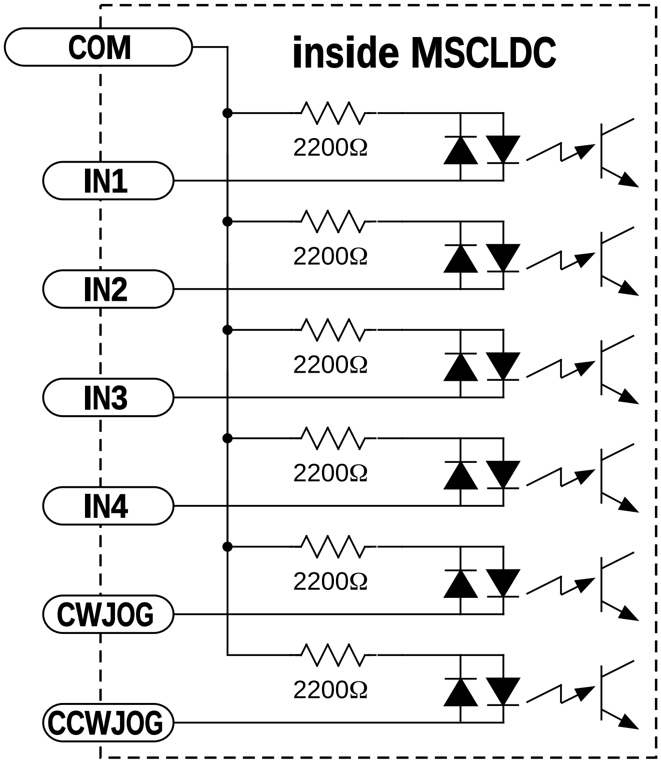


Figure 10 - SCHEMATIC DIAGRAM OF MSCLDC INPUT CIRCUITS

CONNECTING AN INPUT TO A SWITCH OR RELAY

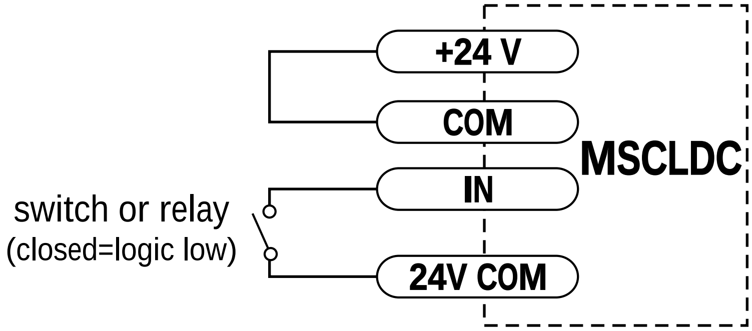


Figure 11

Use normally open momentary contact switch to trigger MSCLDC using *Wait Input* instruction, or use single throw switch for parameter selection using If Input instruction. Use normally open momentary switch for jogging. This connection is appropriate for Tol-O-Matic Reed switches.

CONNECTING AN PNP TYPE PROXIMITY SENSOR TO AN INPUT

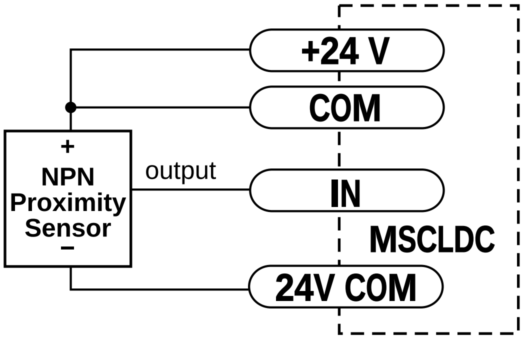


Figure 12

When prox sensor activates, input will go low. This connection is appropriate for Tol-O-Matic Hall-effect sourcing switches.

CONNECTING A PNP PROXIMITY SENSOR TO AN INPUT

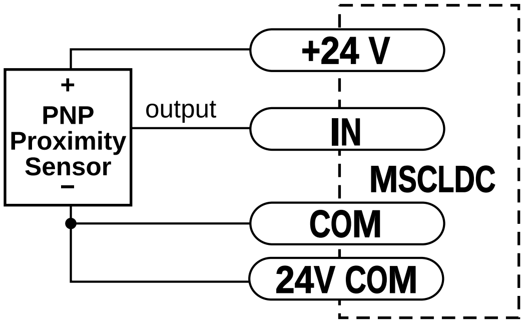


Figure 13

When prox sensor activates, input will go low. This connection is appropriate for Tol-O-Matic Hall-effect sinking switches.

JOG INPUTS

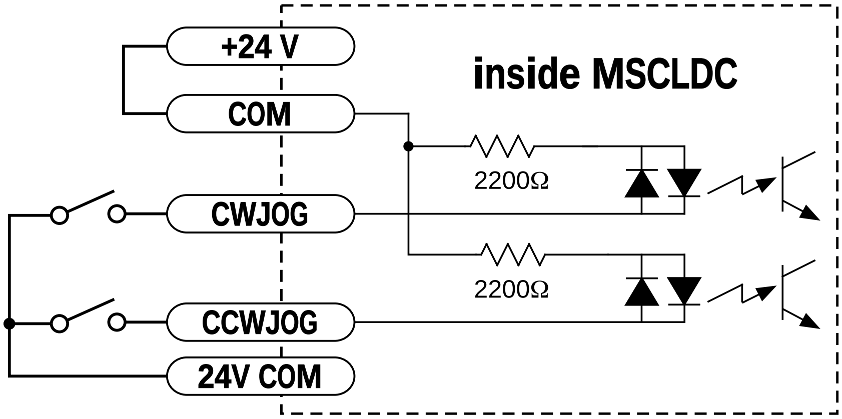


Figure 14

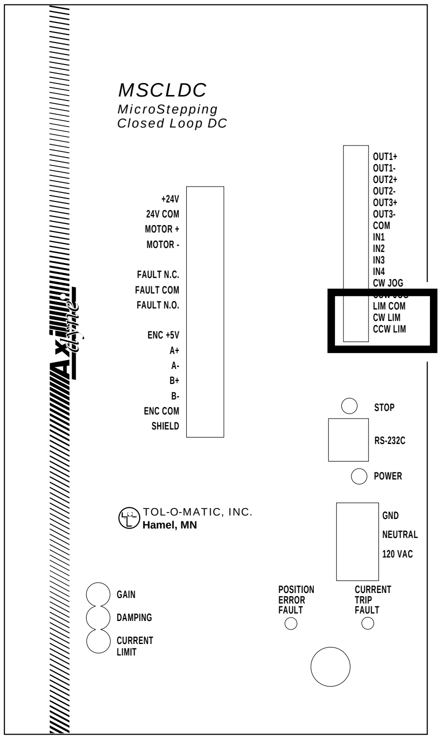
Two of the MSCLDC input terminals are provided for jogging the motor. The inputs are labeled "JOG CW" and "JOG CCW". Taking one of the inputs low commands the drive to move the motor at a pre-designated speed until the contact is opened. A relay or mechanical switch can be used to activate the jog inputs. 5 volt circuitry can also be used. Figure 14 is a schematic diagram of the input circuit.

If using a switch or relay, wire one end to the JOG input and the other to the 24V COM terminal. Then connect the COM and the +24V terminals.

If the MSCLDC is connected to a PC with the programming software running, the jog will function under two conditions:

- if the program is not executing,
- if the program is executing a *Wait Input* command.

Limit Switches



The MSCLDC 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 MSCLDC is limited. This is useful if a program error could cause damage to your system by exceeding the mechanical operating range.

The limits are optically isolated. This allows choice of a voltage for the limit circuits of 5 to 24 volts dc. It also allows use of long wires on limit sensors that may be far from the MSCLDC with less risk of introducing noise to the MSCLDC. The schematic diagram of the limit input circuit is shown in Figure 15.

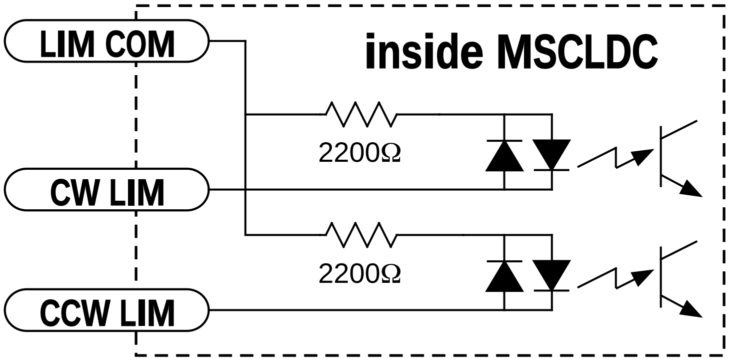


Figure 15

WIRING A LIMIT SWITCH

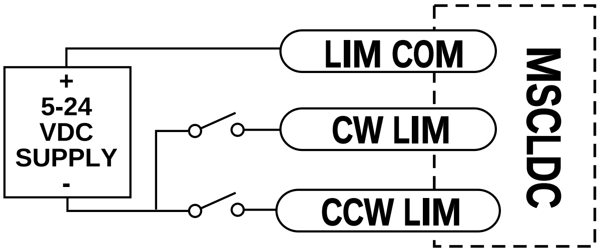
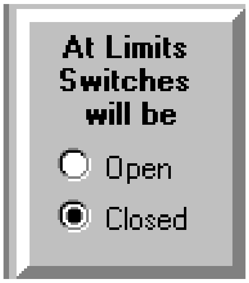


Figure 16

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

Programming for Normally Open or Normally Closed

The main window of the MSCLDC programming software contains a panel for selecting the type of limit switches or sensors.



At Limits Switches will be

○ Open

● Closed

Figure 17

If the limit switch closes 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 open when a limit is reached, select the option button "open." This type of switch is frequently called *normally closed*.

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 like Figure 18:

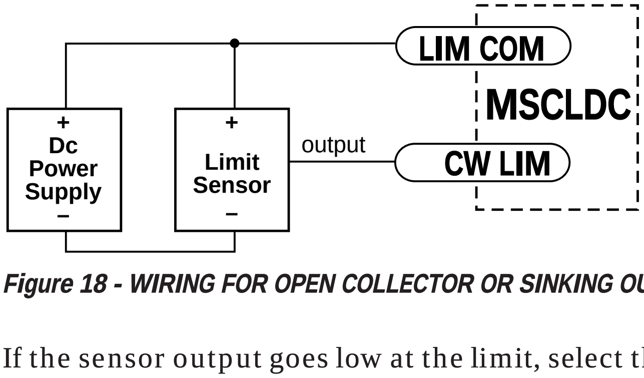


Figure 18 - WIRING FOR OPEN COLLECTOR OR SINKING OUTPUT FROM SENSOR

If the sensor output goes low 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 19:

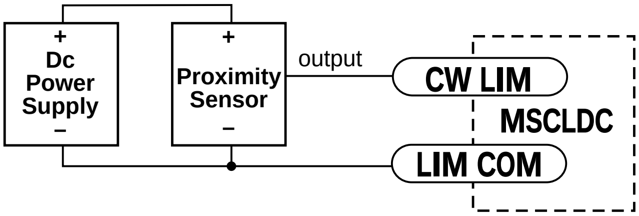


Figure 19 - WIRING FOR SOURCING OUTPUT FROM SENSOR

If the sensor output goes high at the limit, choose the program option “closed.” If the output is low at the limit, select “open.”

MSCLDC 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 MSCLDC will immediately stop the motor, with no deceleration. The red Power LED on the front panel will flash, and no further motion is possible. Ac power must be removed from the drive to reset this condition. If the drive is connected to a PC, the programming software will alert the user to this condition. Reset can be from the PC (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.

Wiring Outputs

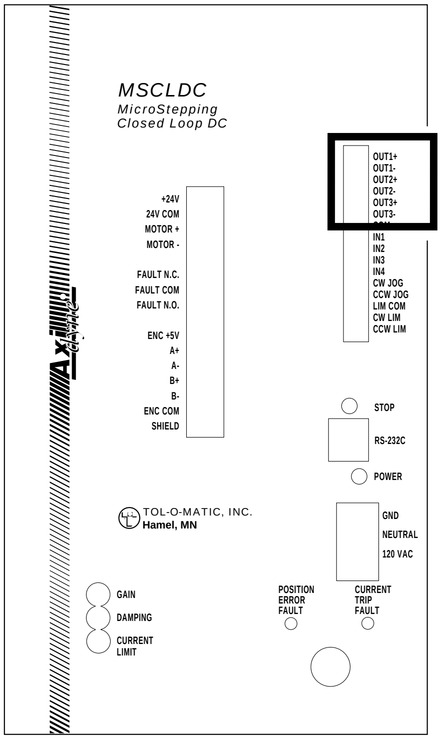


Figure 21 - OUTPUT LOCATIONS

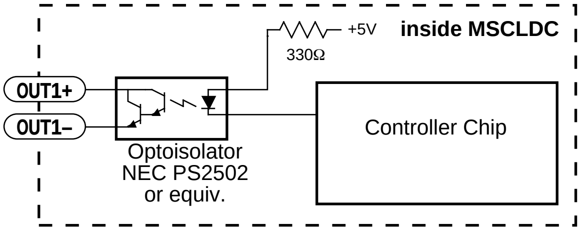


Figure 20

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

All three MSCLDC outputs are optically isolated. That means that there is no electrical connection between the indexer-drive and the output terminals. The signal is transmitted to the output as light. What the load “sees” is a transistor (NPN type) that closes, or conducts current, when the output is “low.” When the output is “high,” the transistor is open.

At power-up, the MSCLDC sets all three programmable outputs to high (open circuit).

Since there is no electrical connection to the MSCLDC, current and voltage must be provided by a PLC or a power supply. The current must be limited to less than 100mA so that the output transistor is not damaged. Normally a resister is used for this, but some loads (such as PLC inputs) limit the current automatically.

Figure 22 shows how to connect an MSCLDC output to an optically isolated PLC input.

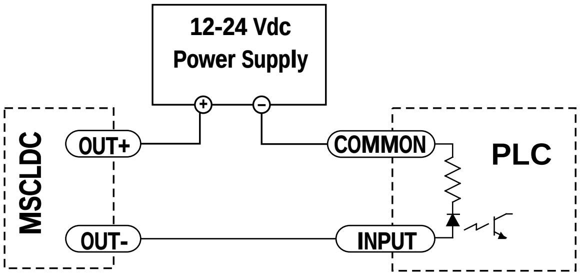


Figure 22

Setting the Potentiometers

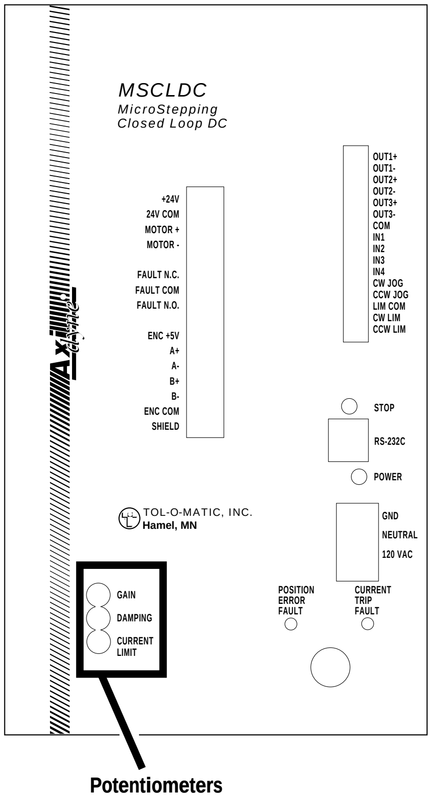


Figure 23 - POTENTIOMETER LOCATIONS

Figure 23 shows the potentiometers on the controller/drive module. All three are 25 turn pots.

The ranges and factory settings of the three potentiometers are:

	Range	*Factory Setting*
Gain	*0.35 to 1.75 Volts/Encoder Count*	*Full CCW (0.35 V/Count)*
Damping	*0 to 25 turns*	*10 turns back from full CW*
Current Trip	*2 Amps to 15 Amps*	*2 Amps*

GAIN POTENTIOMETER

The 25 turn gain pot can be set at maximum (fully CW) for most conditions. The gain may need to be reduced under heavy load conditions.

DAMPING POTENTIOMETER

The 25 turn damping pot is best adjusted to maximum (fully CW), then turned back 5 to 10 turns CCW. This gives the best compromise between good velocity regulation and overly aggressive velocity regulation. Less damping will result in degraded velocity loop performance and may result in instability. More damping may result in instability at low speeds.

OVERCURRENT LIMIT POTENTIOMETER

After the system is tuned and programmed, the current trip should be set. This is done by adjusting the 25 turn current trip pot CCW until the drive current trips while running at maximum load. Then turn the pot 2 to 4 turns CW to allow operating margin.

Tol-O-Motion MSCLDC Programming Software

Overview

The MSCLDC 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 MSCLDC's functions are controlled by the software. The MSCLDC 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 MSCLDC 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 or Windows 95
- At least 4 MB memory (8 MB will make the software run much faster, especially on 386 and 486 systems.)
- 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 MSCLDC 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 MSCLDC ***after*** the Programming software is running on the computer.

Using the Software to Input System Settings

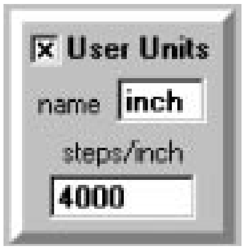
SETTING THE USER UNITS

The main programming window (see Figure 27) allows selection of User Units. This selection is used in conjunction with the encoder-based preset resolution of 1000 steps/revolution to set a scaling factor which allows all move requirements to be entered (whether in programming or via the PIT) in the user's preferred units. If the User Units are not selected, all programming for distance and speed will be in steps and revs/sec.

To make the selection click the cursor on the box for User Units name (Figure 24) and enter a one to four digit description (mm, inch, feet, ????). Using the preset value of 1000 steps/revolution, calculate the steps/unit (using screw pitch or belt wheel diameter and gear ratio) and enter in the user unit box. Click the small user units box off (no **X**) then on (**X**).

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

NOTE: All max speeds will be subject to the 50 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.



☒ User Units

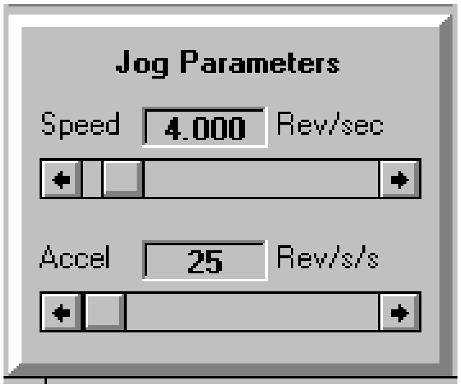
name: inch

steps/inch: 4000

Figure 24 - USER UNITS PANEL

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 25).



Jog Parameters

Speed 4.000 Rev/sec

Accel 25 Rev/s/s

Figure 25 - 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-Matic Software for Programming Motion

OVERVIEW

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

Five 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. *SEEK HOME* searches for a home sensor, “bouncing off” the limits if necessary to find it.

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 outputs a signal to other equipment that a particular place in the program has been reached.

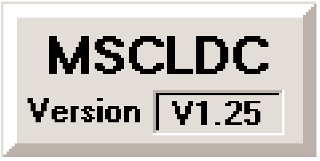
Using the *PIT PROMPT* instruction with the optional Panel-mount Interface (PIT), (or operator panel), the operator can enter distances, speeds and repeat loop counts on a keypad. The drive can also display messages for the operator, pause the program until the operator presses the *ENTER* button, or 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 13 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 MSCLDC. 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 MSCLDC. Software in a chip is usually called ***firmware.*** It is the MSCLDC firmware that runs the drive and executes the program. The drive firmware version is displayed near the top of the screen when the MSCLDC is connected to the computer and turned on (see Figure 26).



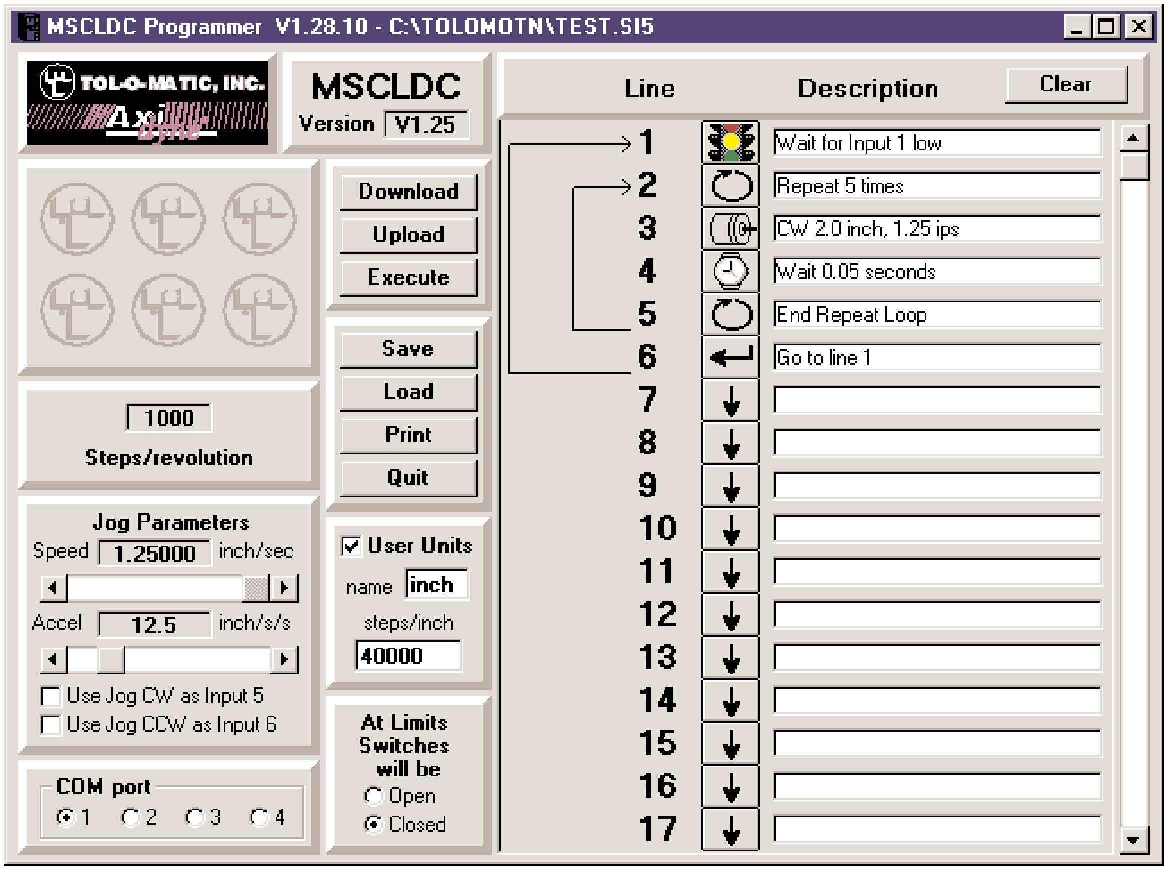
MSCLDC

Version V1.25

Figure 26 - FIRMWARE VERSION DISPLAY

Entering The Program

To activate the MSCLDC 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 27.



MSCLDC Programmer V1.28.10 - C:\TOLOMOTN\TEST.SI5

MSCLDC Version V1.25

1000 Steps/revolution

Jog Parameters

Speed 1.25000 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

☑ User Units

name inch

steps/inch 40000

At Limits Switches will be

○ Open

● Closed

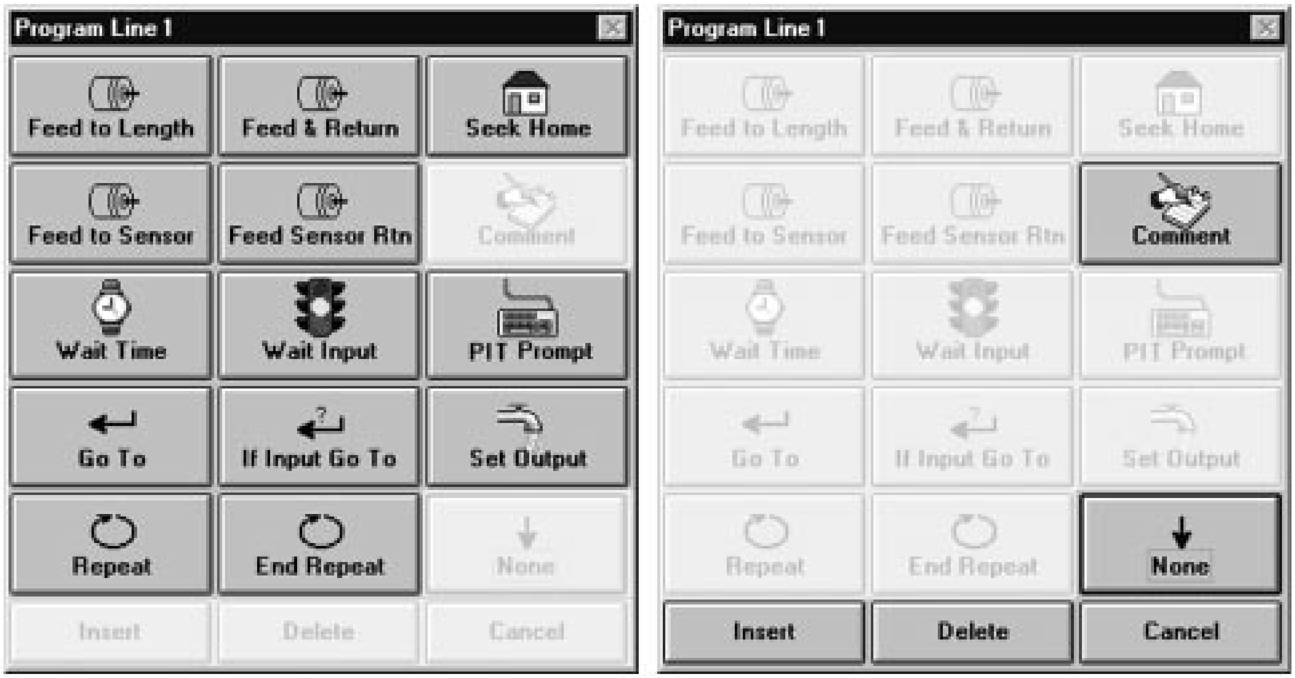
Line	Description
1	Wait for Input 1 low
2	Repeat 5 times
3	CW 2.0 inch, 1.25 ips
4	Wait 0.05 seconds
5	End Repeat Loop
6	Go to line 1
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	

Figure 27 - MAIN PROGRAMMING WINDOW

If an MSCLDC 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 MSCLDC firmware that is in the drive. Program parameter setup is described on page 39.

INSTRUCTIONS & TOOLS

Refer to Figure 28. The programming software provides 13 easy-to-use high-level instructions for building programs. In addition, there are five tools.


INSTRUCTION SET

TOOL SET

Figure 28 - INSTRUCTION/TOOL MENU

	NAME	PAGE REF.	NAME	PAGE REF.
Instructions	Feed to Length	51	Feed & Return	53
	Seek Home	56	Feed to Sensor	54
	Feed to Sensor & Return	55	Wait Time	58
	Wait Input	59	PIT Prompt	49
	Go To	61	If Input Go To	66
	Set Output	64	Repeat	62
	End Repeat	62		

	NAME	PAGE REF.	NAME	PAGE REF.
Tools	Comment	48	None	48
	Insert	47	Delete	47
	Cancel	48		

Most programs will begin with the instruction *WAIT INPUT* or a PIT prompt. This ensures that when power is applied to the MSCLDC, 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 MSCLDC encounters a "None" program line, it simply moves on to the next line, as the icon implies with a downward pointing arrow. After the MSCLDC 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 29), displaying the programming instructions and tools.

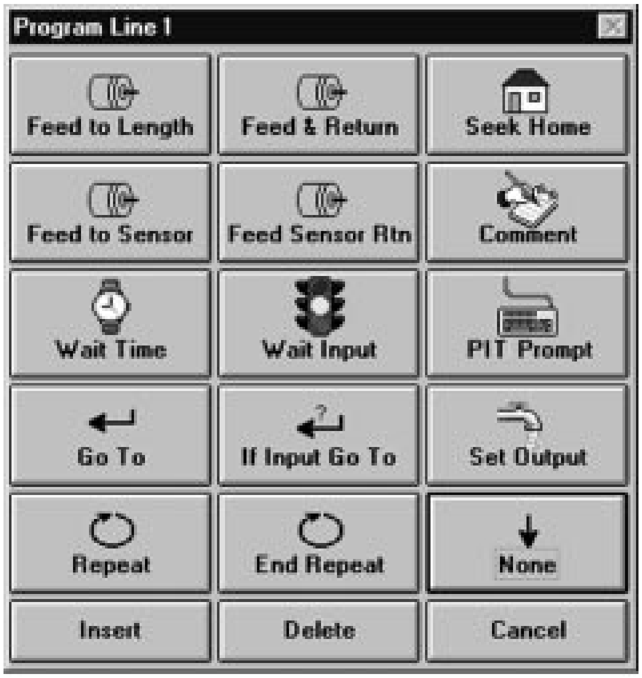


Figure 29 - "PROGRAM LINE" DIALOG BOX

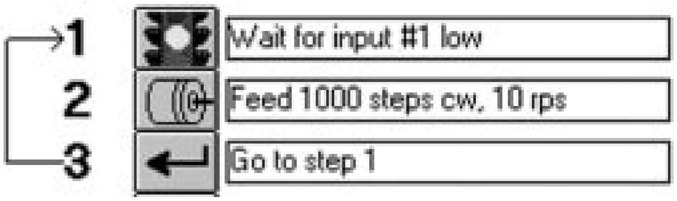
Click on the button marked "Wait Input". The *WAIT INPUT* dialog box will appear. Click on the option button marked "Low" (in the "Condition" group), then click *OK*.

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 "25000," then slide the speed bar to "10 rev/sec." Click *OK*.

The second program line should now show the motor icon and the caption “CW 2000 steps, 10 rps.”

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 30.



1 Wait for input #1 low

2 Feed 1000 steps cw, 10 rps

3 Go to step 1

Figure 30

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.

DOWNLOAD, UPLOAD AND EXECUTE

The MSCLDC 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.

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 MSCLDC 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.

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

EXECUTE - tells the MSCLDC to begin running the program currently in its internal memory, starting on line 1. After pressing the *EXECUTE* button, a box will appear with a STOP button in the middle of it.

STOP - interrupts the indexer-drive at any point in the program. This feature is useful when the drive starts doing things not intended for it to do. **NOTE:** Remember that the MSCLDC 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 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.

Ensure the motor/drive parameters and limits are properly set.. Then, 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 MSCLDC 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.

Press the *EXECUTE* button. The *EXECUTE* dialog box will appear (see Figure 31).

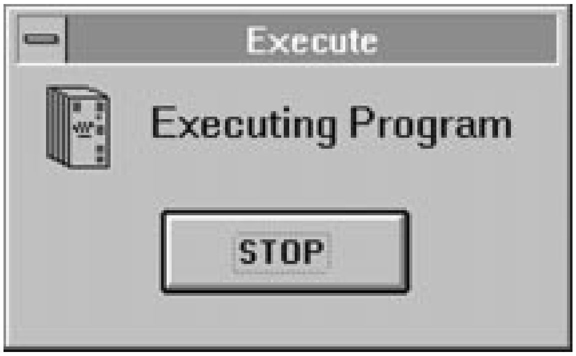


Figure 31 - EXECUTE DIALOG BOX

Each time the switch is closed the motor should move one revolution. (If the switch remains closed, the program will continually repeat.)

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

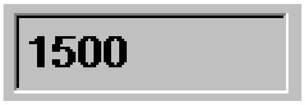
COPYING INSTRUCTIONS

There may be occasions 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 the instruction icon to be copied, and drag the icon onto another one elsewhere in the program. This is referred to as ***drag and drop.***

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 MSCLDC 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 32. 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.

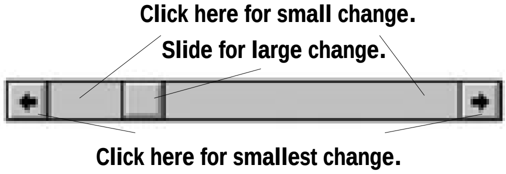


Figure 32 - PRECISE SCROLL BAR ADJUSTMENT

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 33.

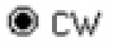


Figure 33 - OPTION BUTTON

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 34), 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.

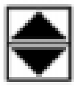


Figure 34 - SPIN BUTTON

PIT Interface

Overview

The MSCLDC 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 (Panel Mount User Interface)

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 six 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.

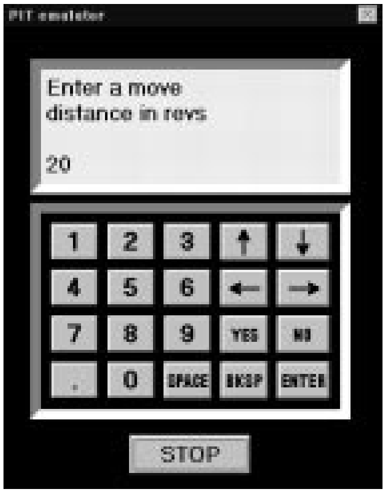
3. ***Prompt the user to make a decision.*** *Example:* Offer the user an option such as changing set up parameters, that can be responded to by pressing the *YES* or *NO* keys.

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.

To run a program from the computer using the PIT, press the *EXECUTE* button. If the program in the drive contains any PIT instructions, a different *EXECUTE* dialog box will appear on the screen. The PIT *EXECUTE* dialog box looks and acts like the real PIT (see Figure 35). It will display messages, and clicking on the buttons will enter data. Like the other *EXECUTE* dialog box, there is a *STOP* button that can interrupt the program at any time.


Enter a move
distance in revs

20

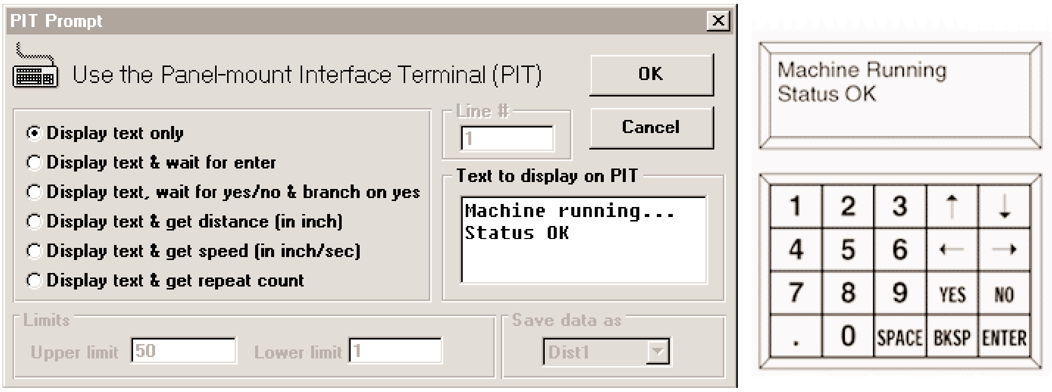
Figure 35 - ON-SCREEN PIT EMULATION

NOTE: The *real* PIT does not have a *STOP* button.

The emulated PIT saves the expense of a second RS232 port on the MSCLDC. It also allows evaluation of the PIT before purchase.

TO DISPLAY A MESSAGE ON THE PIT (SEE FIG. 36):

1. Click on a program line icon.
2. Select the *PIT PROMPT* instruction.
3. Type “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)

● Display text only
○ Display text & wait for enter
○ Display text, wait for yes/no & branch on yes
○ Display text & get distance (in inch)
○ Display text & get speed (in inch/sec)
○ Display text & get repeat count

Line #: 1

Text to display on PIT:
```
Machine running...
Status OK
```

Limits — Upper limit: 50 Lower limit: 1

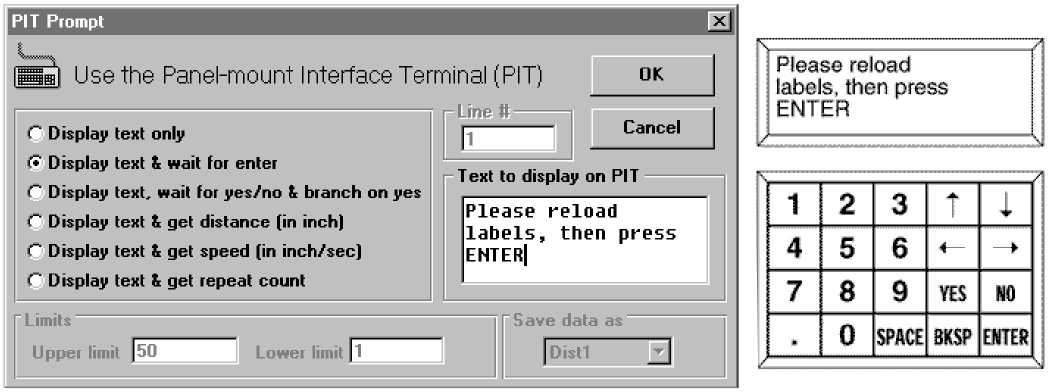
Save data as: Dist1

Machine Running
Status OK

Figure 36 - DISPLAYING A MESSAGE ON PIT

TO PAUSE UNTIL USER PRESSES ENTER (SEE FIG. 37):

1. Select the *PIT PROMPT* instruction.
2. Type “Please reload labels, then press ENTER” in the text box.
3. Select the *DISPLAY TEXT & WAIT FOR ENTER* option button.
4. 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 (in inch)
○ Display text & get speed (in inch/sec)
○ Display text & get repeat count

Line #: 1

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

Limits — Upper limit: 50 Lower limit: 1

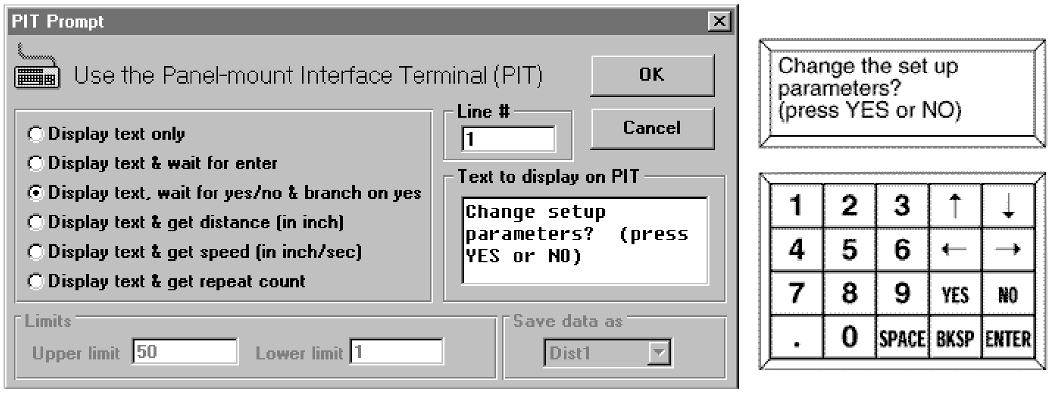
Save data as: Dist1

Please reload
labels, then press
ENTER

Figure 37 - PAUSING UNTIL USER PRESSES ENTER

TO LET USER MAKE A DECISION (PIT BRANCHING) (SEE FIG. 38):

1. Put an *PIT PROMPT* instruction on line 1.
2. Type “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 “12” in the line # box.
5. Click *OK.*
6. Enter parameter setting instructions, starting on *PROGRAM LINE 12*.
7. Place a *GO TO LINE 2* instruction at the end of the parameter-setting instructions.



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 (in inch)
○ Display text & get speed (in inch/sec)
○ Display text & get repeat count

Line #: 1

Text to display on PIT:

```
Change setup
parameters?  (press
YES or NO)
```

Limits: Upper limit 50 Lower limit 1

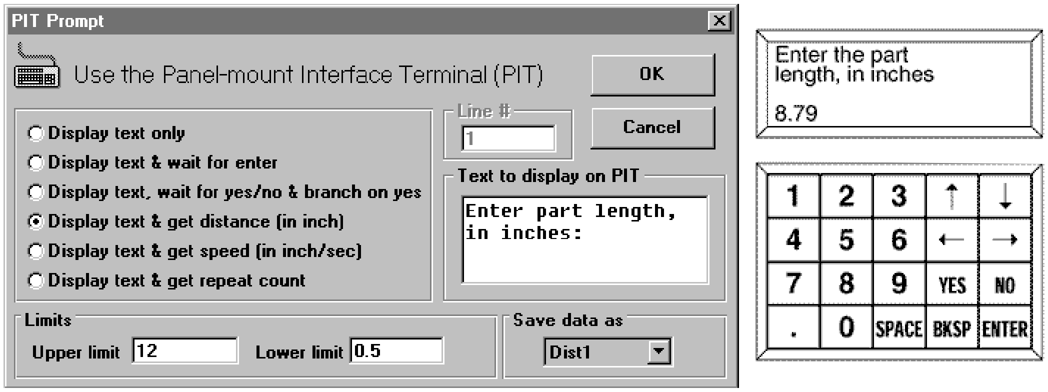
Save data as: Dist1

Change the set up
parameters?
(press YES or NO)

Figure 38 - PIT BRANCHING

TO ASK THE USER FOR A MOVE DISTANCE (SEE FIG. 39):

1. Select an *PIT PROMPT* instruction.
2. Type “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 an 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.



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 (in inch)
○ Display text & get speed (in inch/sec)
○ Display text & get repeat count

Line #: 1

Text to display on PIT:

```
Enter part length,
in inches:
```

Limits — Upper limit: 12 Lower limit: 0.5

Save data as: Dist1

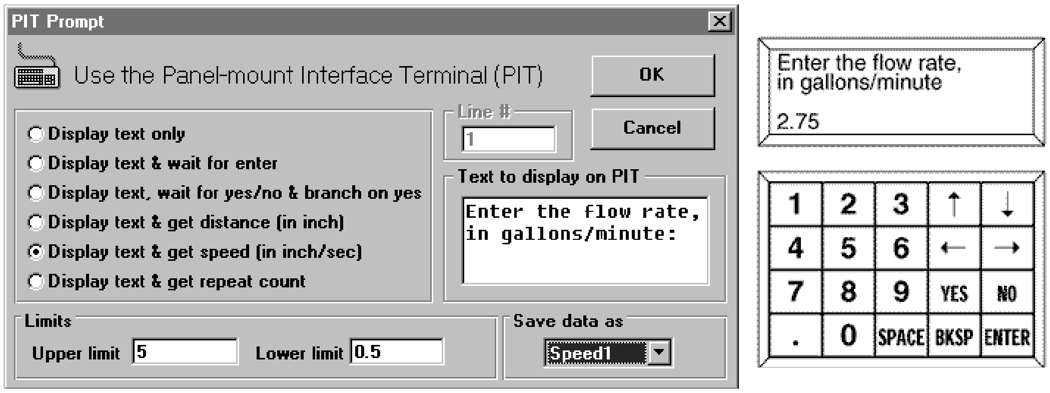
Enter the part
length, in inches

8.79

Figure 39 - PROMPT USER FOR MOVE DISTANCE

TO GET A SPEED FROM THE USER (SEE FIG.40):

1. Select an *PIT PROMPT* instruction.
2. Type “Enter the flow rate, in gallons/minute” 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 flow rates between 1 and 5 gal/min).
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..



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 (in inch)
● Display text & get speed (in inch/sec)
○ Display text & get repeat count

Line #: 1

Text to display on PIT:

```
Enter the flow rate,
in gallons/minute:
```

Limits — Upper limit: 5 Lower limit: 0.5

Save data as: Speed1

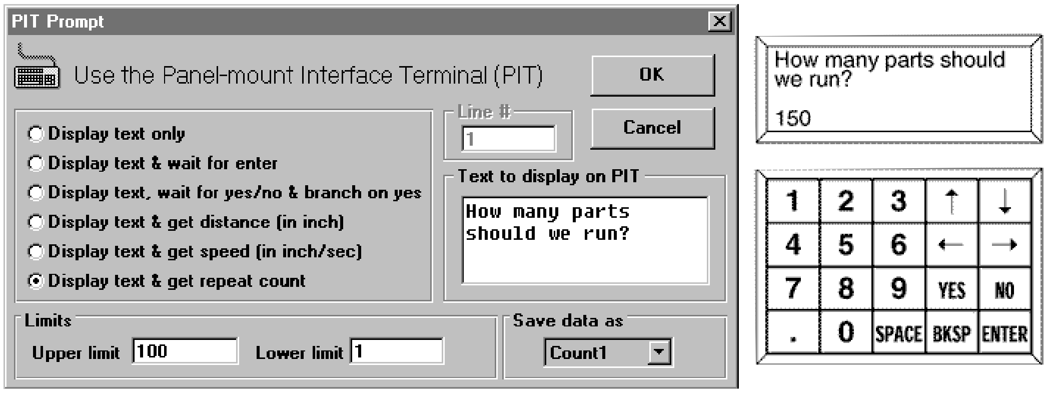
Enter the flow rate,
in gallons/minute

2.75

Figure 40 - PROMPT USER FOR SPEED

TO GET A REPEAT COUNT FROM THE USER (SEE FIG. 41):

1. Select an *PIT PROMPT* instruction.
2. Type “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.



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 (in inch)
○ Display text & get speed (in inch/sec)
● Display text & get repeat count

Line #: 1

Text to display on PIT:
```
How many parts
should we run?
```

Limits: Upper limit 100 Lower limit 1

Save data as: Count1

How many parts should we run?
150

1	2	3	↑	↓
4	5	6	←	→
7	8	9	YES	NO
.	0	SPACE	BKSP	ENTER

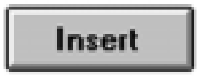
Figure 41 - PROMPT USER FOR REPEAT COUNT

Overview

There are five programming tools which appear in the *PROGRAM LINE* dialog box (page 34, Figure 28). 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” instead of choosing one of the 14 instructions. A warning will appear that all the program lines after the one chosen will be “pushed down” to make room, and that the last line (100) will be lost.
3. Click *OK* to restore the main window. The instructions are re-ordered, and the line needed for the new program line is available.
4. 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. Select *DELETE*. Another dialog will appear to confirm or cancel the instruction.
3. Click *OK* 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. (If an error has been made, repeat steps 1 and 2, above.)

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

NONE

This tool leaves a blank program line for future use without the need to Insert later.

CANCEL

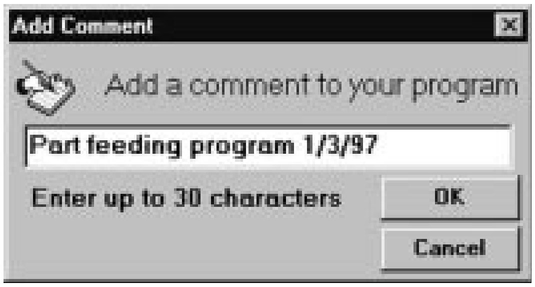
This tool exits from programming without saving or downloading the program.

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 MSCLDC executes a program, it simply skips over the comments.

Refer to Figure 42. 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

Part feeding program 1/3/97

Enter up to 30 characters

Figure 42 - COMMENT DIALOG BOX

There is a limit to the number of comments the program can have. The MSCLDC 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 and then confirming OK will erase all program stored and displayed in the active Main Programming window lines 1-100. CLEAR does not clear program stored in a connected MSCLDC.

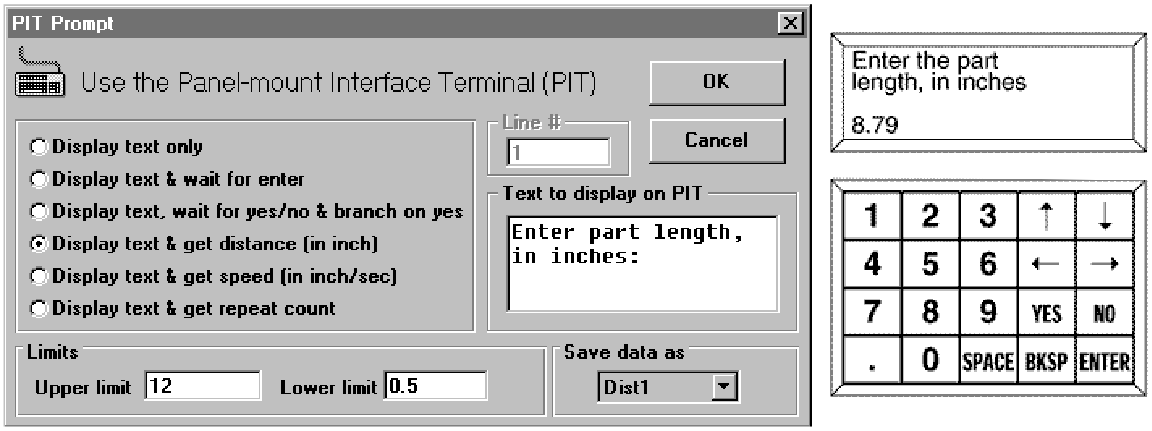
Overview

There are 13 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 (Man Machine 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 43). 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.



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 (in inch)
○ Display text & get speed (in inch/sec)
○ Display text & get repeat count

Line #: 1

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

Limits: Upper limit 12 Lower limit 0.5

Save data as: Dist1

Enter the part
length, in inches
8.79

Figure 43 - 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 instruction will not accept any value bigger than 100 or smaller than 1.

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.)

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 and User Units were not selected on the Main Programming Window,the data is not scaled and the user must enter the distance in steps, or steps/sec.

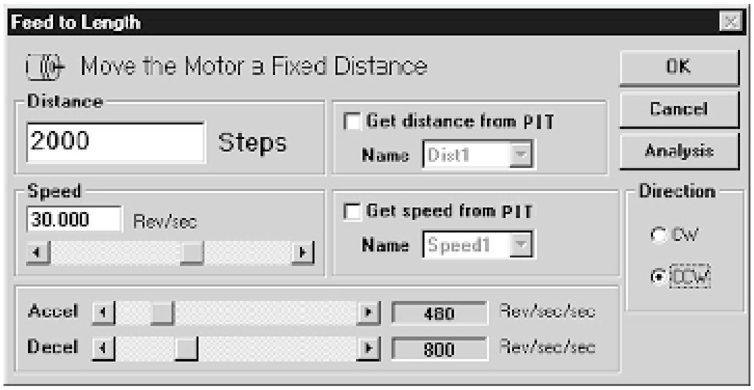
To pause the program until the user presses the *ENTER* key on the PIT, choose the option marked “Display text & wait for enter.”

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 moves. This is the instruction to use to move the motor a fixed number of steps. 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 to appear (see Figure 44). This is where the parameters for the move are entered..


Feed to Length

Move the Motor a Fixed Distance

Distance: 2000 Steps

□ Get distance from PIT — Name: Dist1

Speed: 30.000 Rev/sec

□ Get speed from PIT — Name: Speed1

Accel: 480 Rev/sec/sec

Decel: 800 Rev/sec/sec

Direction: ○ CW ● CCW

Figure 44 - FEED TO LENGTH DIALOG BOX

DISTANCE - the number of motor steps to move. The maximum number 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 MSCLDC stop and ask the user for an entry. An *PIT PROMPT* instruction is needed elsewhere in the program for that.

SPEED - the maximum speed the motor is to go, in revolutions per second. 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.

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. The MSCLDC has an acceleration range of 1 to 3000 revs/sec/sec.

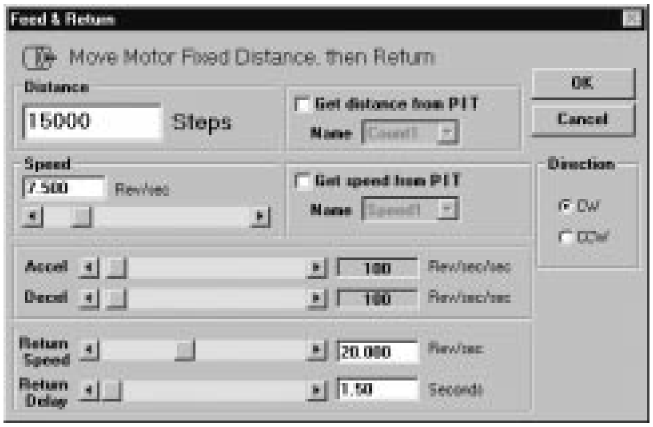
DECEL - this is the rate at which the drive decelerates to a stop at the end of the move. The range is the same as for acceleration. Because friction encourages a motor to stop, decel can often be set higher than accel.

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 vs. time graph of the move along with a calculation of the move duration and the times spent accelerating and decelerating.

FEED & RETURN

The *FEED & RETURN* instruction 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).



Distance: 15000 Steps
☐ Get distance from PIT — Name: Count1
Speed: 7.500 Rev/sec
☐ Get speed from PIT — Name: Speed1
Direction: ● CW ○ CCW
Accel: 100 Rev/sec/sec
Decel: 100 Rev/sec/sec
Return Speed: 20.000 Rev/sec
Return Delay: 1.50 Seconds

Figure 45 - 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 pages 51 - 52.

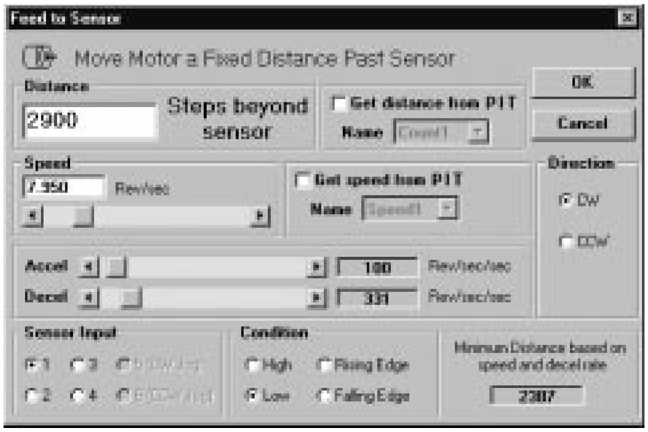
RETURN SPEED must also be set. The range is .025 to 50 revolutions per second. In the case of the cut-off knife, set to feed slowly as the knife is cutting, then retract quickly. Thus, the return speed would be higher than the forward speed.

RETURN DELAY determines how long the MSCLDC 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 speed 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 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 MSCLDC inputs to stop motion.



Feed to Sensor

Move Motor a Fixed Distance Past Sensor

Distance: 2900 Steps beyond sensor

□ Get distance from PIT
Name: Count1

Speed: 7.950 Rev/sec

□ Get speed from PIT
Name: Speed1

Direction: ● CW ○ CCW

Accel: 100 Rev/sec/sec

Decel: 331 Rev/sec/sec

Sensor Input: ● 1 ○ 2 ○ 3 ○ 4

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

Minimum Distance based on speed and decel rate: 2387

Figure 46 - 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 MSCLDC 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 MSCLDC 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 state. This is the default state of an input if nothing is connected to it.

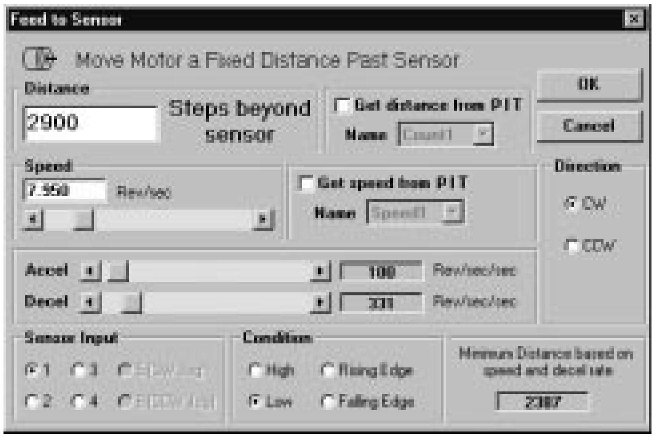
LOW - move until specified input is at a low voltage state.

RISING EDGE - move until the signal goes from low to high. This is similar to the high condition, but the difference is important. If a sensor is wired to the MSCLDC, it 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 MSCLDC will complete the motion, then refuse to start again because the input signal is still high. If a rising edge is chosen, the MSCLDC would proceed with the input voltage high and stop when the sensor signal goes from low to high again.

FALLING EDGE - the opposite of rising edge. MSCLDC waits for input voltage to go high, then low.

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

Move Motor a Fixed Distance Past Sensor

Distance: 2900 Steps beyond sensor

□ Get distance from PIT — Name: Count1

Speed: 7.950 Rev/sec

□ Get speed from PIT — Name: Speed1

Direction: ● CW ○ CCW

Accel: 100 Rev/sec/sec

Decel: 331 Rev/sec/sec

Sensor Input: ● 1 ○ 2 ○ 3 ○ 4

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

Minimum Distance based on speed and decel rate: 2387

Figure 47 - 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 MSCLDC 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 MSCLDC.

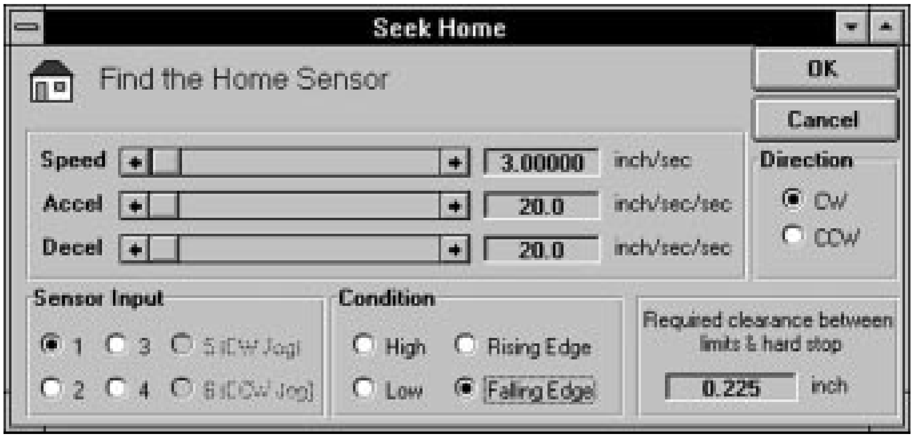
The indexer-drive feeds each time it is triggered until the knife trips the sensor, then returns to the starting point.

NOTE: The maximum distance for any program is 16,000,000 steps, the longest distance the MSCLDC can track. If a move of more than 16 million steps is made before reaching the sensor, the MSCLDC 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.

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 MSCLDC inputs and place a *SEEK HOME* command at or near the beginning of the program.



Find the Home Sensor

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 48 - 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 MSCLDC 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 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 state.
RISING EDGE - move until the signal goes from low to high. 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 MSCLDC will move the load to the edge of the home sensor.

NOTE: For the load to be at the exact same position after each Seek Home command, choose *RISING EDGE* or *FALLING EDGE*.

FALLING EDGE - the opposite of rising edge. MSCLDC waits for input voltage to go high, then low.

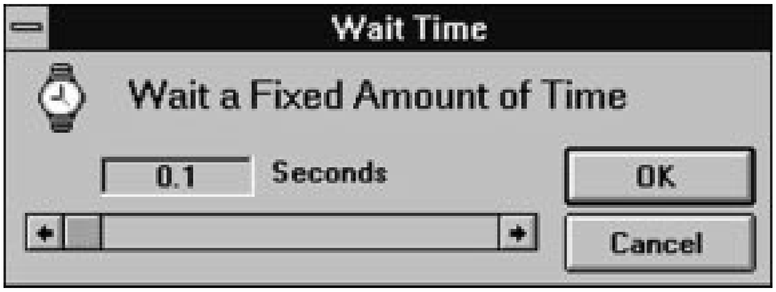
The MSCLDC begins a *SEEK HOME* command by moving the motor in the direction specified. If the home sensor is found, the motor decelerates to a stop, then backs up to the sensor. If a limit is encountered before the home sensor is found, the MSCLDC reverses the direction of motion, moves to the opposite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

There is a box in the lower right-hand corner of the *SEEK HOME* dialog box. This tells how many steps the MSCLDC 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 MSCLDC will pause for that time before proceeding to the next line in the program The range is 0.01 to 300 seconds.



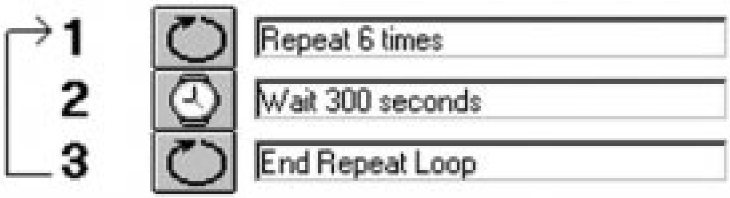
Wait Time

Wait a Fixed Amount of Time

0.1 Seconds

Figure 49 - 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 49.



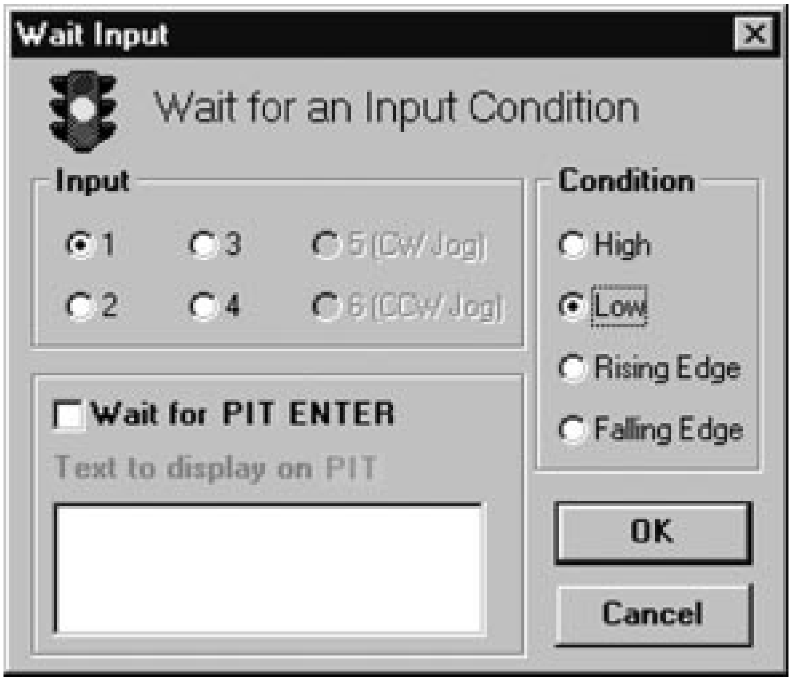
1 Repeat 6 times
2 Wait 300 seconds
3 End Repeat Loop

Figure 50 - 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 MSCLDC to wait before it starts a motion.



Wait Input

Wait for an Input Condition

Input

● 1 ○ 3 ○ 5 (CW/Jog)

○ 2 ○ 4 ○ 6 (CCW/Jog)

Condition

○ High

● Low

○ Rising Edge

○ Falling Edge

□ Wait for PIT ENTER

Text to display on PIT

Figure 51 - WAIT INPUT DIALOG BOX

The *WAIT INPUT* instruction has two parameters (see Figure 51). The first is to specify the input. The second parameter is what kind of voltage condition to expect. The choices are:

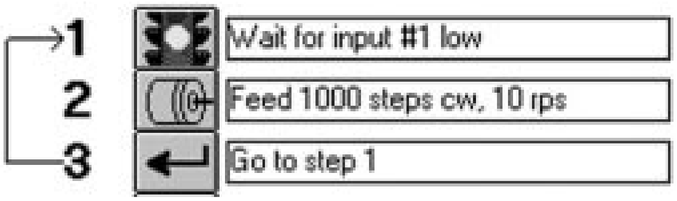
HIGH - Wait until the specified input is at a high voltage state. This is the state an input will be in if nothing is connected to it, so be careful using this condition. If a wire comes loose, it could cause an undesired motion.

LOW - Wait until specified input is at a low voltage 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 to high. This is similar to the high condition, but the difference is important. For example, a signal into the MSCLDC 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 MSCLDC 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 MSCLDC will wait for the input voltage to go low, then high before moving.

FALLING EDGE **-** The opposite of *RISING EDGE.* MSCLDC waits for input voltage to go high, then low.

Figure 52 illustrates a simple program that can be executed with the momentary contact switch wired between *INPUT 1* and *GND.* Pressing the button will cause the motor to move 1000 steps.



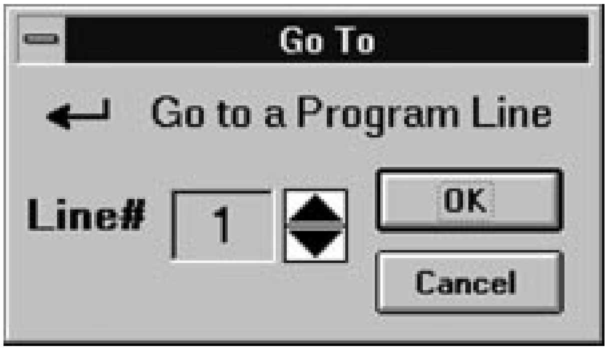
1 Wait for input #1 low
2 Feed 1000 steps cw, 10 rps
3 Go to step 1

Figure 52

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.



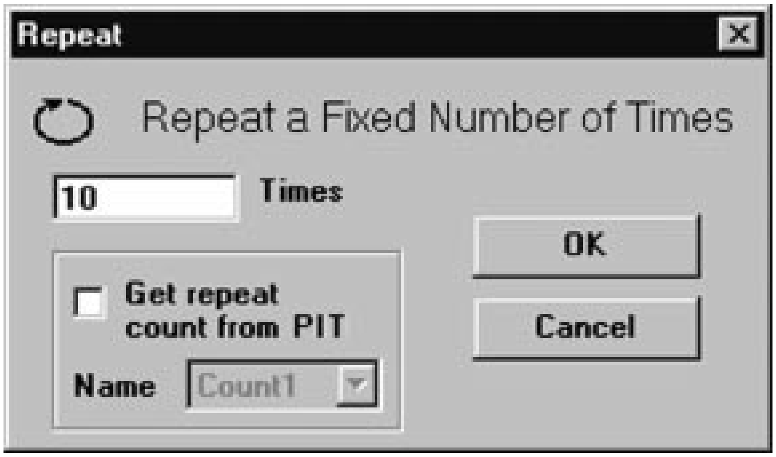
Line# 1

Figure 53 - 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.



Repeat

Repeat a Fixed Number of Times

10 Times

□ Get repeat count from PIT

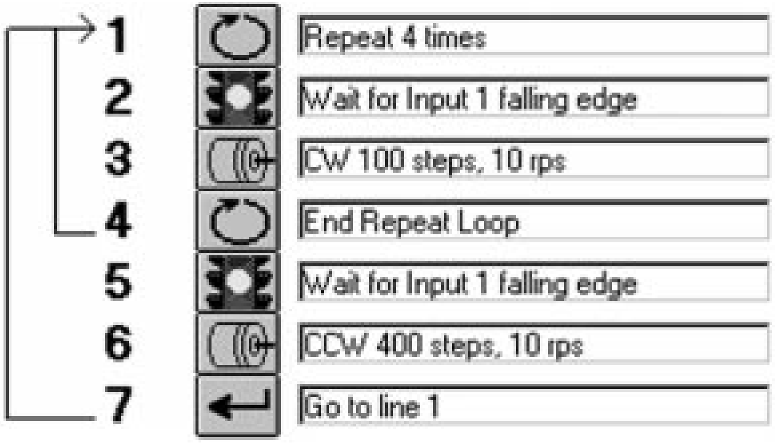
Name Count1

Figure 54 - 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.

An MSCLDC controlling the X axis, or motion between columns, would be programmed as follows:

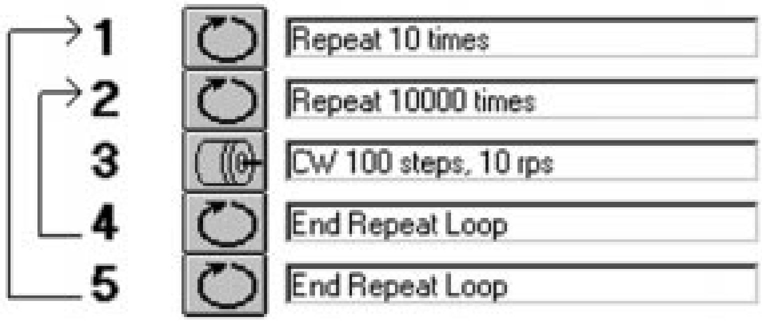
Refer to Figure 55. The program begins on line 1, entering the repeat loop. The next four times, the MSCLDC will wait for the voltage at Input 1 to fall, then move clockwise 100 steps, taking the dispenser to the next container. After the fourth time, the MSCLDC drops out of the loop into line 5. This 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 100 steps, 10 rps
4  End Repeat Loop
5  Wait for Input 1 falling edge
6  CCW 400 steps, 10 rps
7  Go to line 1
```

Figure 55

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 56.



```
1  Repeat 10 times
2  Repeat 10000 times
3  CW 100 steps, 10 rps
4  End Repeat Loop
5  End Repeat Loop
```

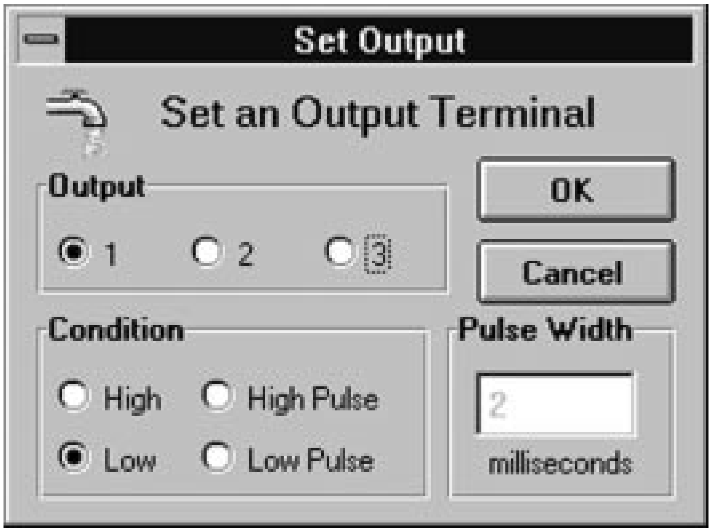
Figure 56

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*.

SET OUTPUT

This command allows 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 57).



Set Output

Set an Output Terminal

Output: ● 1 ○ 2 ○ 3

Condition: ○ High ○ High Pulse ● Low ○ Low Pulse

Pulse Width: 2 milliseconds

Figure 57

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.

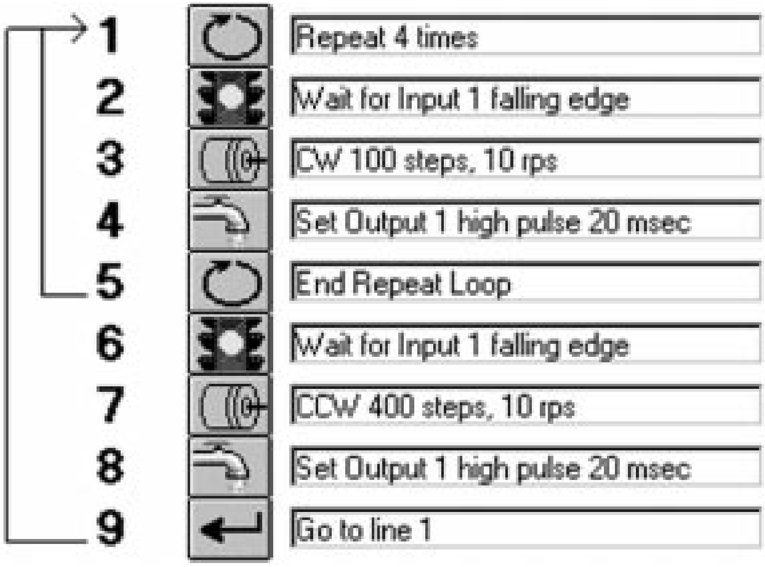
HIGH PULSE - Makes the photo transistor open for a specified amount of time (2 to 500 milliseconds).

LOW PULSE - Makes the photo transistor close for a specified amount of time (2 to 500 milliseconds).

At power-up, the MSCLDC 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 MSCLDC 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.

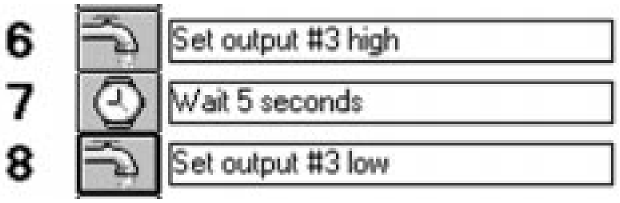
There are five rows and five columns of containers, each 100 steps away from the next. Each trigger command is to cause a move to the next position. After the fifth container is full, return to the first, 400 steps back. After adding a *SET OUTPUT* instruction after each *FEED TO LENGTH*, the program is as shown in Figure 58.



1 Repeat 4 times
2 Wait for Input 1 falling edge
3 CW 100 steps, 10 rps
4 Set Output 1 high pulse 20 msec
5 End Repeat Loop
6 Wait for Input 1 falling edge
7 CCW 400 steps, 10 rps
8 Set Output 1 high pulse 20 msec
9 Go to line 1

Figure 58

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 59 will produce a high pulse of 5 seconds on Output 3.



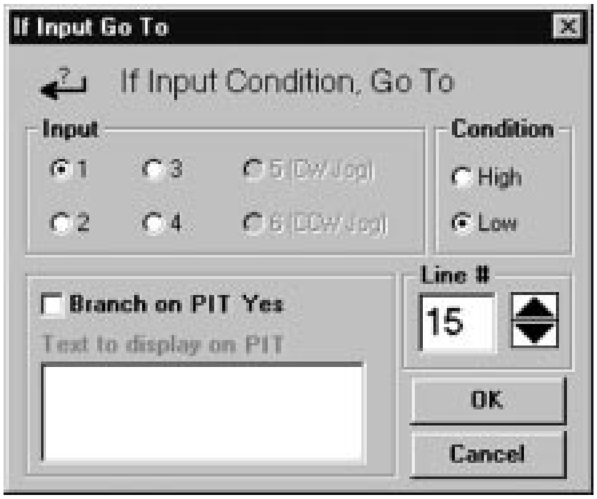
6 Set output #3 high
7 Wait 5 seconds
8 Set output #3 low

Figure 59

IF INPUT GO TO

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

Refer to Figure 60. Choose an input terminal for the instruction to check. The MSCLDC 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

Input

● 1 ○ 3 ○ 5 (CW Jog)

○ 2 ○ 4 ○ 6 (CCW Jog)

Condition

○ High

● Low

□ Branch on PIT Yes

Text to display on PIT

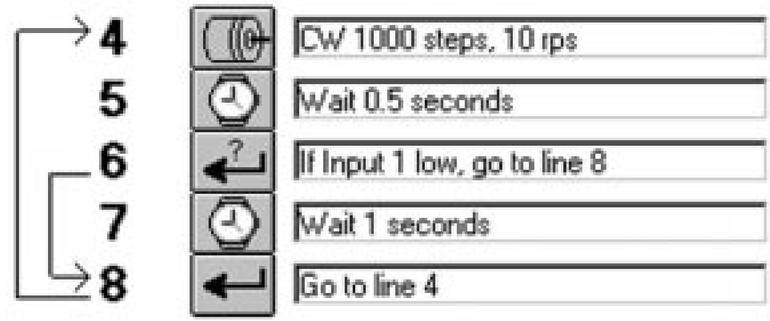
Line #

15

Figure 60 - 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 MSCLDC’s task is to feed parts. Normally the MSCLDC 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 61 shows the program required to accomplish this.



4 CW 1000 steps, 10 rps

5 Wait 0.5 seconds

6 If Input 1 low, go to line 8

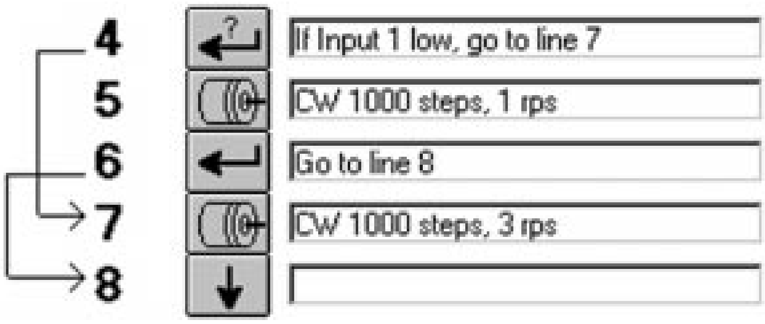
7 Wait 1 seconds

8 Go to line 4

Figure 61

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 MSCLDC. This time, the program is written as shown in Figure 62.



4 If Input 1 low, go to line 7

5 CW 1000 steps, 1 rps

6 Go to line 8

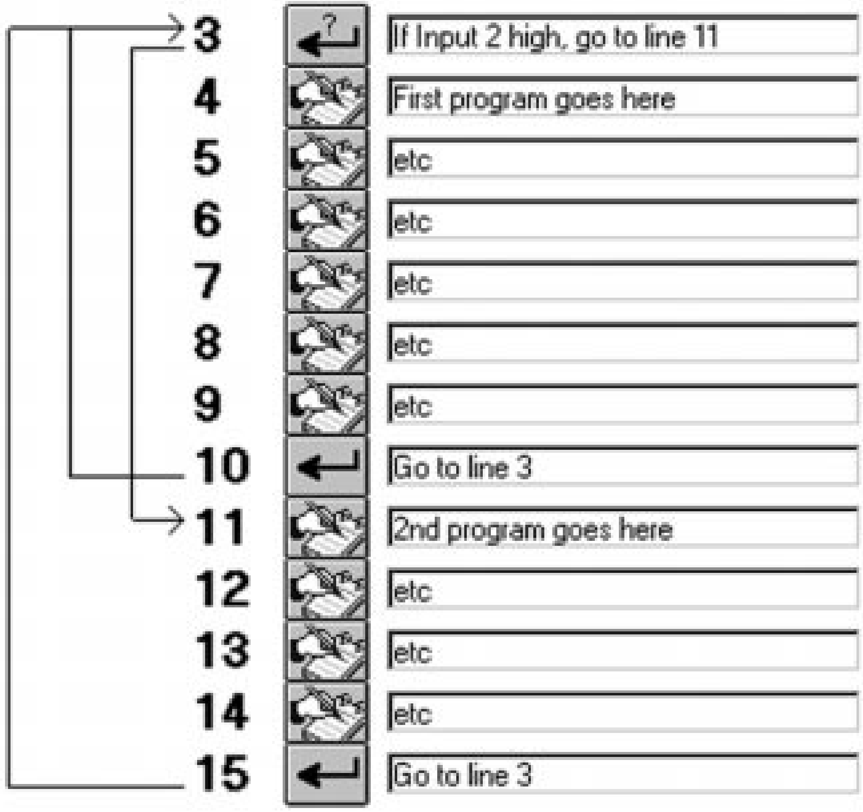
7 CW 1000 steps, 3 rps

8

Figure 62

When the MSCLDC 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 revolutions per second. 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 rev/sec. The MSCLDC 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 63.



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 63

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.

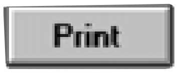
The optional Panel-mount Interface (PIT) as the decision making input of an *IF INPUT* instruction. Simply check the box marked “Branch on PIT YES” and type in the message the operator is to see. If the operator presses the *YES* button, the drive will jump to the line specified in the line number box.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

SAVE, LOAD, PRINT AND QUIT

In addition to exchanging programs with the MSCLDC, 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 file name must conform to the usual DOS/Windows 3.1 rules. The safest approach is to use only letters and numbers in the file name, and to avoid special characters like "?" or "\".

LOAD - provides a dialog box showing all the ".SI5" files on the drive. Click to select one, then click OK to load it. Several example programs are installed with the programming software. It's a good idea to load some of the examples and look at them: they may help with new applications.



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

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

Notes:

Technical Specifications

MSCLDC Controller/Drive System Specifications

Power: Input: Voltage 85-132 Vac; Frequency 60Hz; Current: 7Amp (max)
Output: Voltage 48 Vdc[1]; Continuous current 10Amp Peak current[2] 15Amp

Performance: System resolution: 1,000 steps/revolution (500 line encoder)
Position range: 16,000,000 steps
Velocity range: 0.025 to 50 revolutions/sec. (larger motors limit max velocity)
Accel/Decel range: 1 to 3,000 rev/sec/sec
Performance range may be limited by motor selection and application.
For motor performance see torque/speed charts on pages 81-83.

Inputs: Two dedicated Limit switch inputs optically isolated 5-24Vdc
CW/CCW jog inputs, optically isolated 5-24 Vdc
Four general purpose programmable inputs optically isolated 5-24 Vdc
Encoder 6 terminals (+5Vdc, A+, A-, B+ B-, GND)

Potentiometer Inputs: Current Trip: adjustment of torque (current) fault limit trip, 200ms delay
Gain: Sets scaling of motor torque to position error
Damping: Sets response to dynamic system disturbance

Outputs: Three programmable optically isolated outputs 24Vdc 100 mA max (external power supply is required)

Fault Output: solid state relay output for position or current trip remote indicator

Fault LED Outputs: Two LEDs indicating current trip or position error

Environmental
Temperature: 0 - 50° C
Humidity: 10 -95% non-condensing

[1] 48Vdc is no load output

[2] 1 second operating at peak current

MSCLDC DIMENSIONS

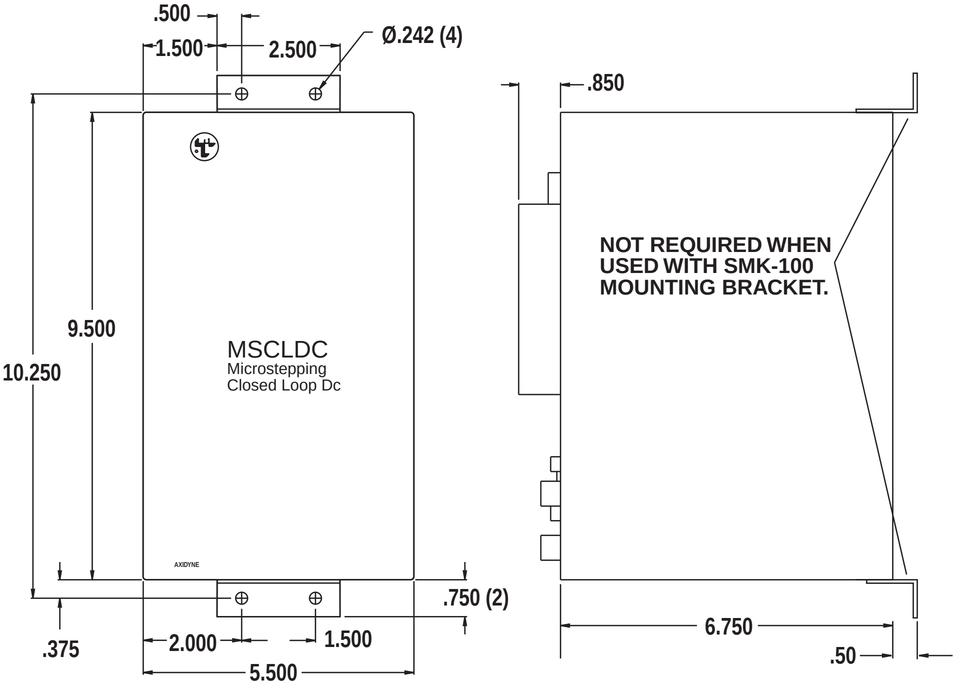


Figure 64

OPTIONAL PIT

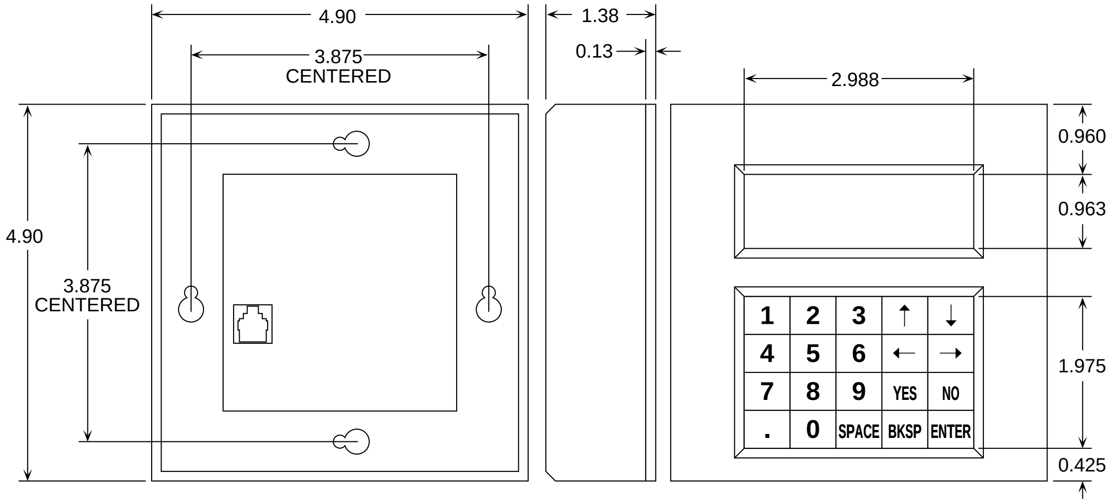


Figure 65

Environmental Concerns

The MSCLDC controller module is designed to operate in an industrial environment; however, severe atmospheric contamination, electrical noise, or temperature extremes can affect system performance. To avoid performance problems, operate the system within the following environmental guidelines:
Operating Temperature: 0˚-50˚C (32˚122˚F)
Humidity: 10-95 percent, non-condensing

Safety Considerations

When installing any motion control system, safety should be a primary concern. All Axidyne hardware should be installed to conform with local and national electrical safety codes. Failure to observe safe working practices when installing or servicing this equipment can expose you to dangerous voltages.

When mounting system components, care should be taken not to place heat producing devices underneath or near the unit.

Mounting the MSCLDC

The MSCLDC can be mounted with top and bottom angle brackets provided.

Alternately Tol-O-Matic offers an optional mounting bracket kit (PN3600-9006) for the MSCLDC that can be used to mount the closed loop dc positioning module. Dimensions for the mounting bracket are shown in Figure 66.

Mounting Bracket Kit

A	7.00
B	0.75
C	1.00 TYP
D	9.50
E	ø.194 THRU (18)
F	.105 REF
G	7.25
H	6.250
J	4.60
K	2.15
L	1.00
M	.25
N	1.40
P	1.99
Q	2.375
R	2.40
S	2.75
T	1.50
U	1.00
V	2.18
W	2.375
X	2.75
Y	Ø.194 THRU (11)
Z	.105
AA	Ø.136
BB	4.800
CC	4.75
DD	1.250

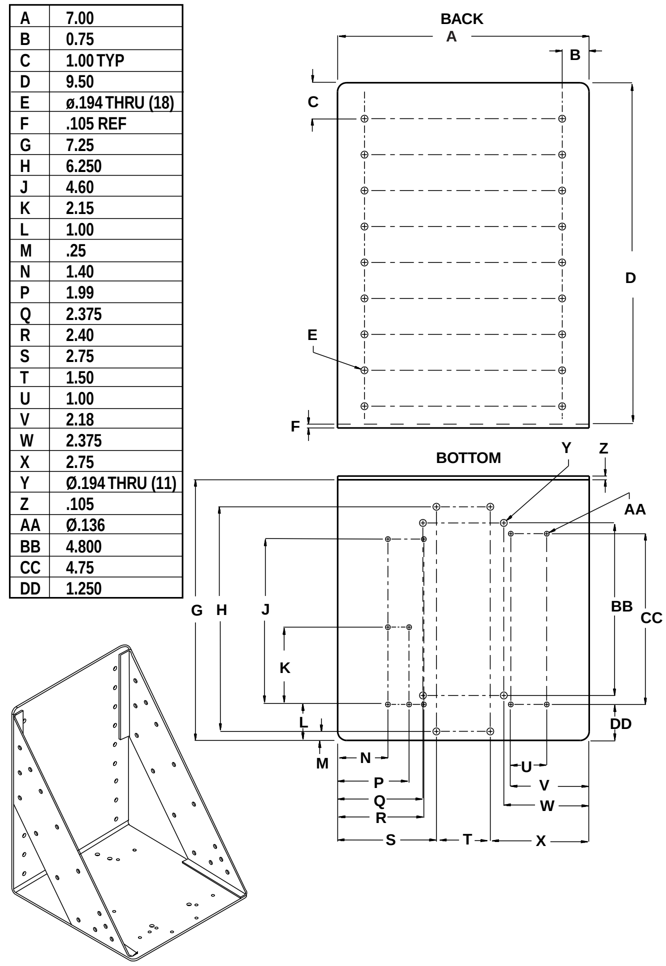


Figure 66

Mounting the Optional PIT

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

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 MSCLDC, 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 only available from the factory.

FLUSH MOUNTING

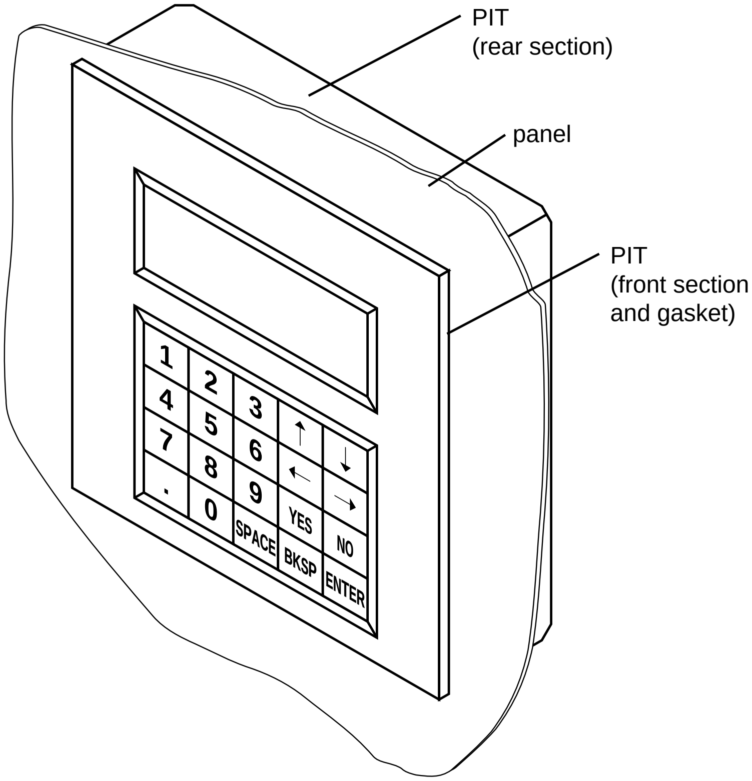


Figure 67 - FLUSH MOUNTING THE PIT

Refer to Figure 67. 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

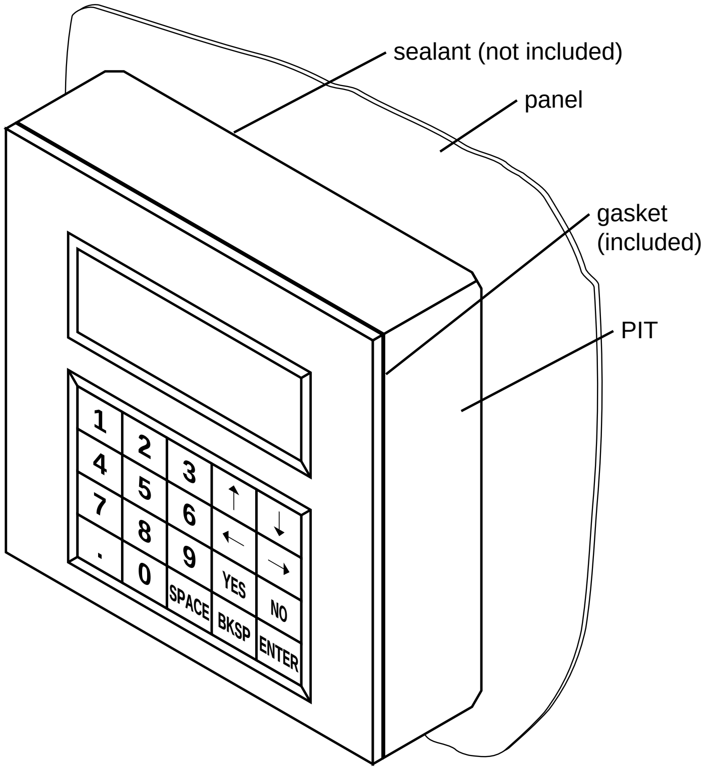


Figure 68 - SURFACE MOUNTING THE PIT

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 MSCLDC. 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 65 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.

Notes:

Overview

Axidyne dc motors are brush-type permanent magnet motors. A selection of motor speed/torque characteristics is available to match Axidyne actuator applications. The motors have permanent magnet poles on the stators and apply continuous power to the rotor through the brushes and commutator (see Figure 69).

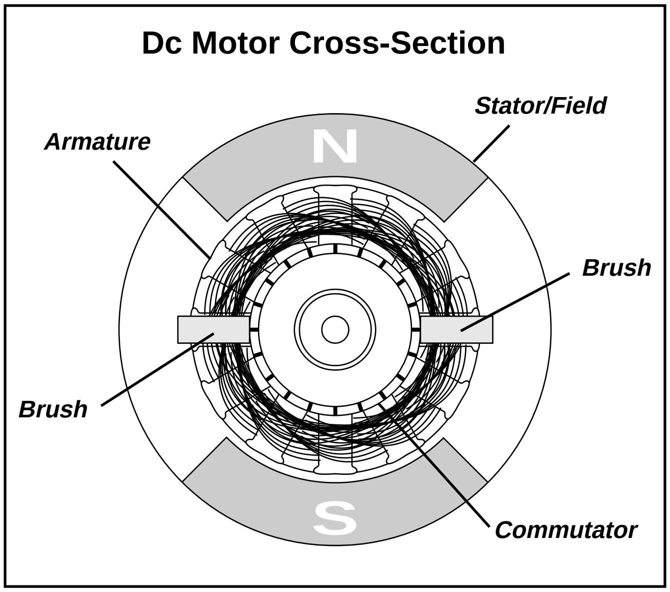


Figure 69

Tol-O-Matic utilizes a two channel (A & B) differential incremental encoder for feedback. The encoder provides 500 counts per revolution, giving a resolution of 1,000 counts per revolution used in the dual mode.

Axidyne Closed Loop Dc Positioning and Drive Modules have a typical final position error of +/- 2 encoder counts. System resolution is a function of screw pitch, belt drive wheel circumference, reductions, and the encoder resolution.

FEATURES

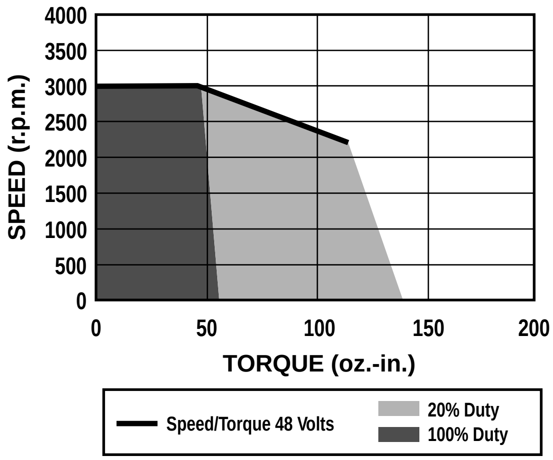
- ***Precision balanced rotors***
- ***ABEC class bearings***
- ***Ferrite magnetics***
- ***Quiet motor designs***
- ***Precision machined dimensions***
- ***Common NEMA mechanical flanges simplify interfacing to standard gearboxes***
- ***Two channel differential encoder (500 count, used in dual (2x) mode).***

Motor Mounting

Axidyne brush dc motors have Nema faces for in-line applications. Mounting kits are used to attach Axidyne motors to Axidyne screw-drive actuators. The kits include standard mounting plates, spacers and fasteners. NOTE: A flexible coupler between the motor shaft and the load is recommended to isolate the motor from vibration and to compensate for possible slight shaft misalignment. Reverse parallel motor mounting is also available on screw actuators, using an enclosed toothed belt drive with 1:1 or 2:1 ratio. Motors may be direct coupled to belt-drive actuators or drive through a 3:1 reduction box. (See Axidyne product catalog no. 3600-4077 for details).

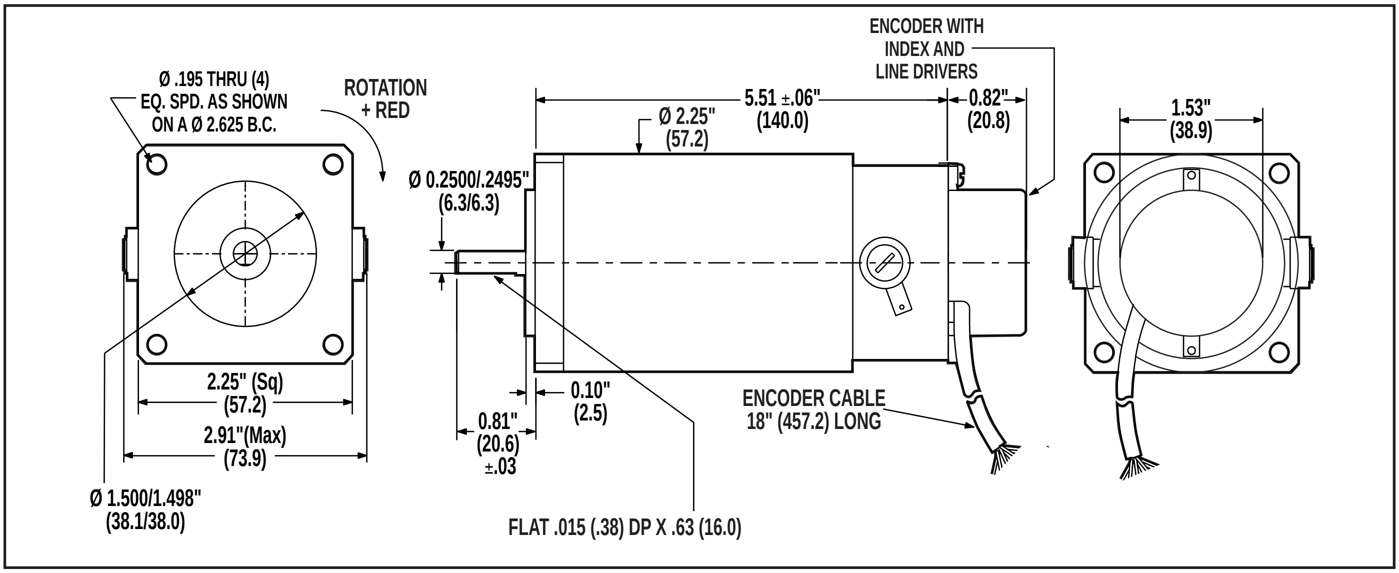
Model MRB-231

Speed-Torque Characteristics

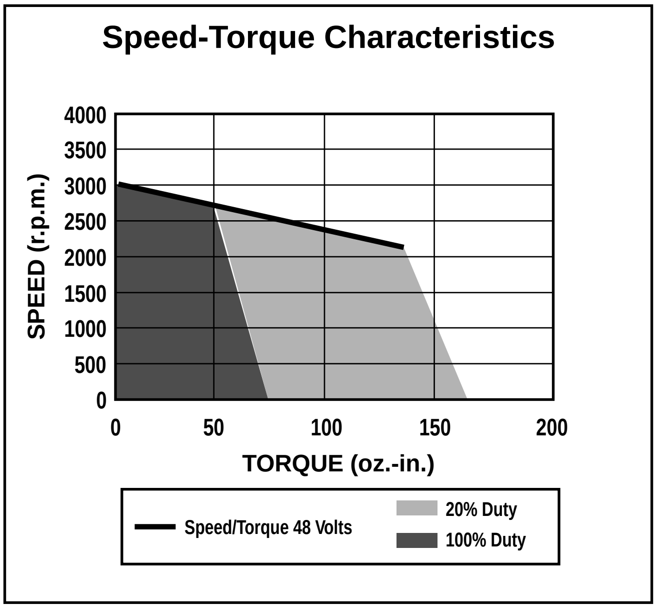


KE:	*12.7 Volts/1000 r.p.m.*
KT:	*17.1 oz.-in./Amp.*
Ra:	*1.7 Ohms*
Rotor Inertia:	*1.92 oz.in.2*
Max. Temp.:	*105° F.*
Weight:	*3.5 lbs.*

Dimensions

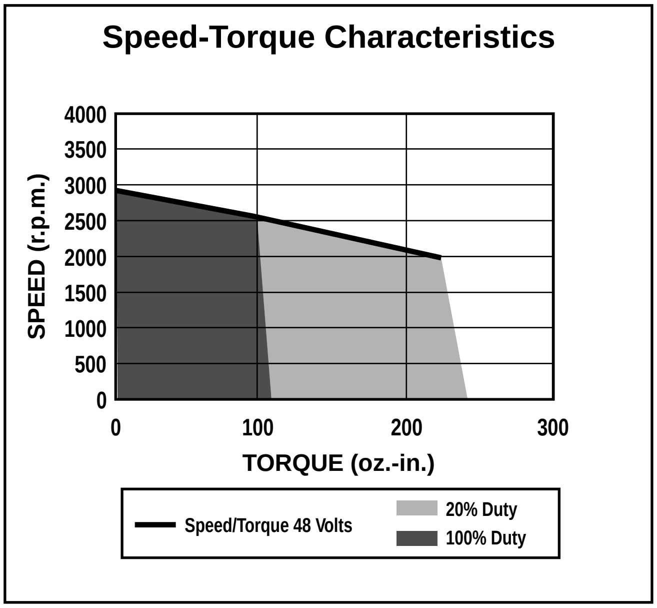


Model MRB-341


Speed-Torque Characteristics

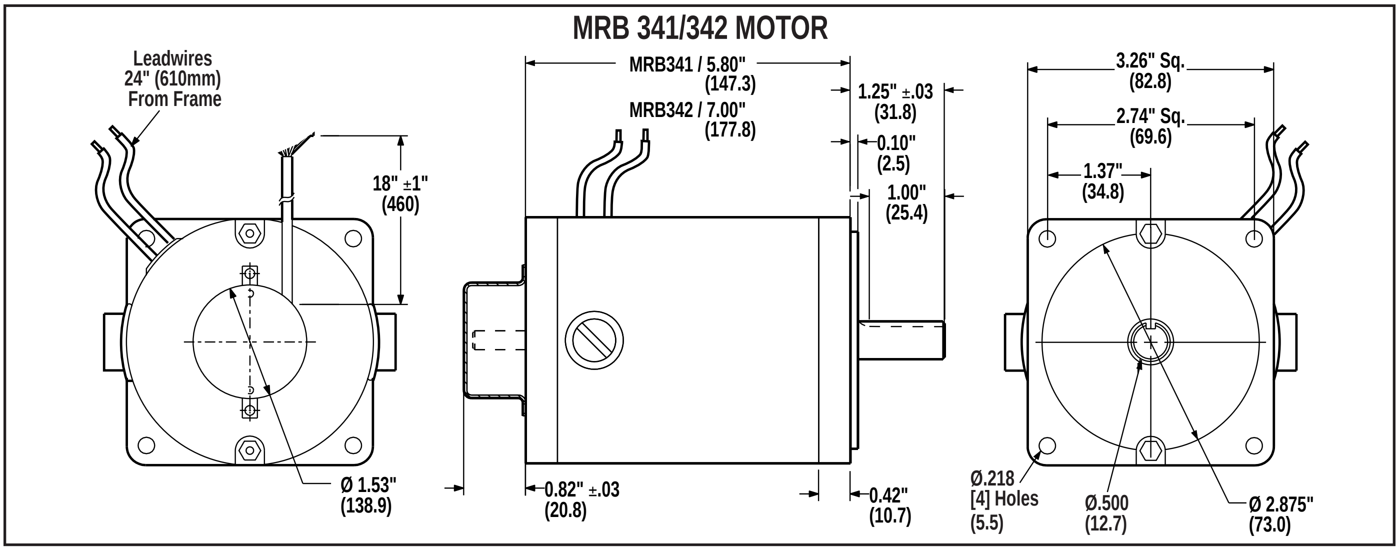
KE:	*15.12 Volts/1000 r.p.m.*
KT:	*19.75 oz.-in./Amp.*
Ra:	*0.97 Ohms*
Rotor Inertia:	*15.36 oz.in.2*
Max. Temp.:	*105° F.*
Weight:	*6.0 lbs.*

Model MRB-342

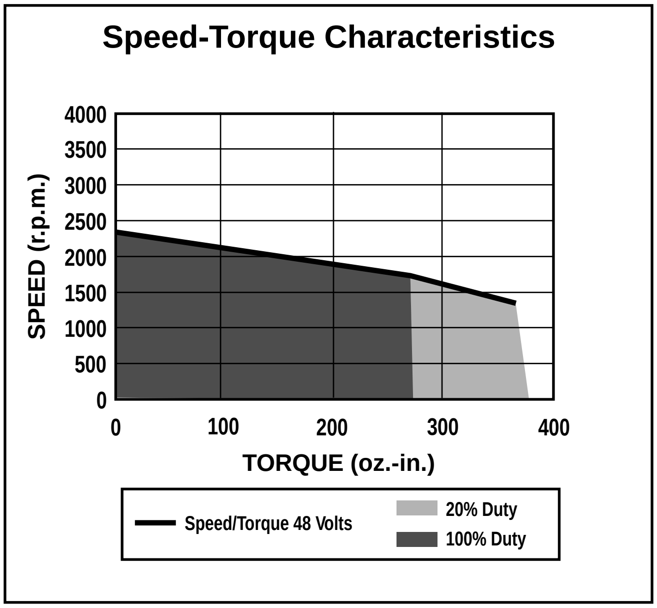

Speed-Torque Characteristics

KE:	*15.83 Volts/1000 r.p.m.*
KT:	*20.35 oz.-in./Amp.*
Ra:	*0.51 Ohms*
Rotor Inertia:	*20.48 oz.in.2*
Max. Temp.:	*105° F.*
Weight:	*8.4 lbs.*

Dimensions

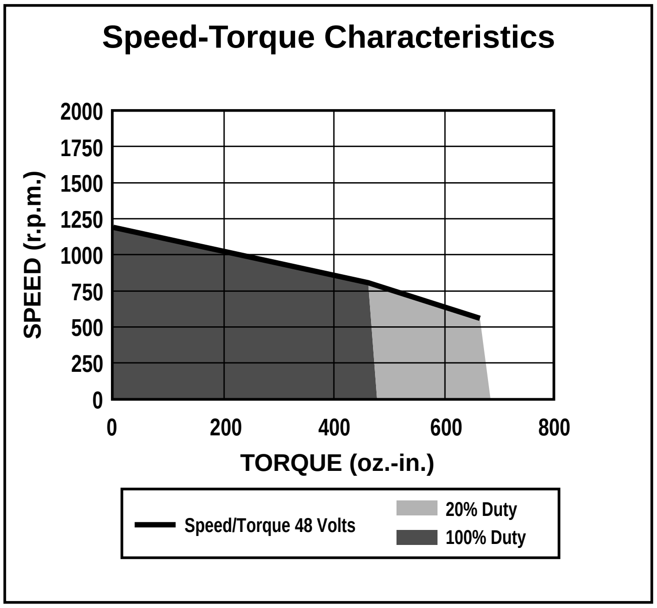

MRB 341/342 MOTOR

Model MRB-401



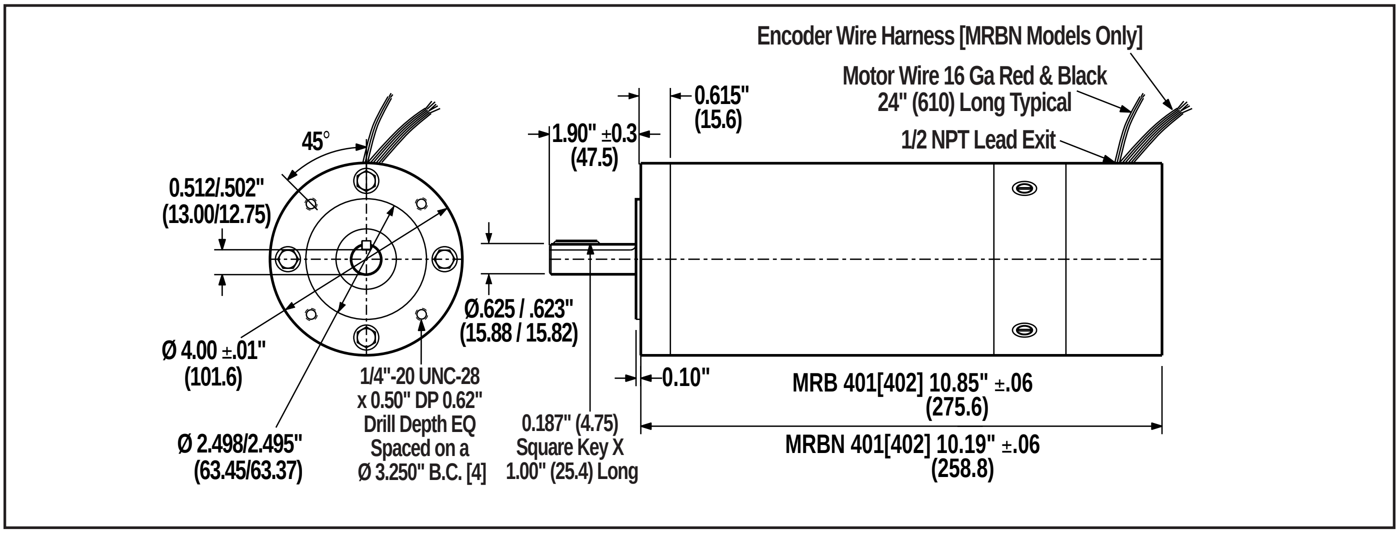
KE:	22.5 Volts/1000 r.p.m.
KT:	30.5 oz.-in./Amp.
Ra:	0.6 Ohms
Rotor Inertia:	100.67 oz.in.2
Max. Temp.:	105° F.
Weight:	17 lbs.

Model MRB-402



KE:	35.8 Volts/1000 r.p.m.
KT:	48.4 oz.-in./Amp.
Ra:	0.87 Ohms
Rotor Inertia:	122.28 oz.in.2
Max. Temp.:	105° F.
Weight:	20 lbs.

Dimensions



Encoder Overview

This is a revolutionary modular type encoder using SMT technology to provide high reliability. Sensors are set up differently under a single LED light source. Installation is simple using a slide/lock mechanism to align center and gap for maximum performance. No adjustments are required and no mechanical rubbing occurs once installed.

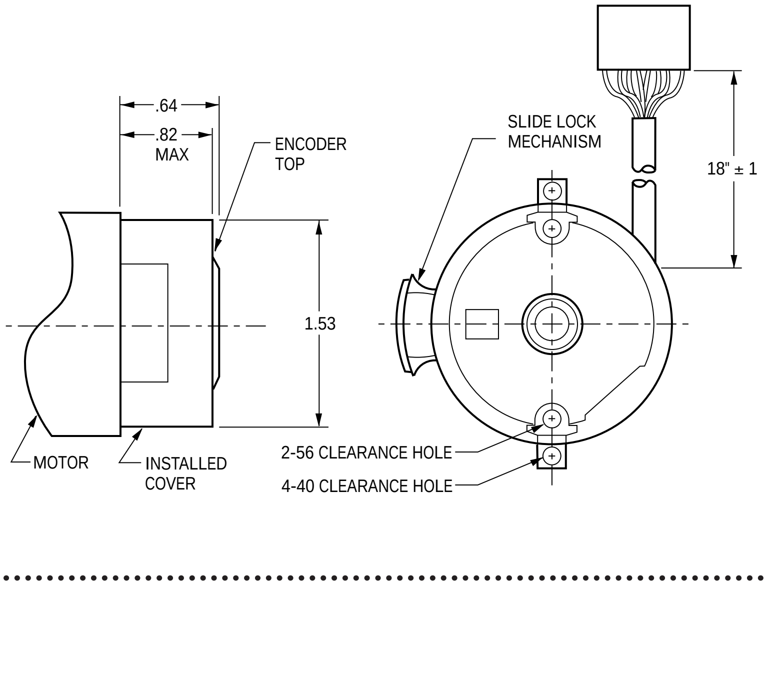
The cover has an integral positive locking system which eliminates the usually required mounting hardware.

NOTE: To extend the encoder leads use a grounded metal junction box. Ensure continuity of the shield which should be single point grounded.

Features

- ***Self aligning***
- ***Self centering***
- ***Self gapping***
- ***Frequency response to 100 KHz (all channels)***
- ***Differential Index***
- ***Positive lock-on cover***

Dimensions



Wire Color Code	***Encoder Terminal***
Red	*+5V*
Black	*GND*
White	*A+*
Yellow	*A-*
Green	*B+*
Brown	*I- (not used)*

Notes:

Symptom	Probable Cause	Corrective Action
Motor runs excessively hot and motor torque seems to diminish. Motor case temperature exceeds 180°F (82,2°C).	1. Excessive loading	1. Reduce surrounding ambient temperature.
	2. High ambient temperature which exceeds motor temperature limit of 105°F (40.5°C).	2. Determine motor/drive torque, speed and duty cycle rating is correct for given application.
	3. Excessive motor losses.	3. Compare apparent thermal resistance (Rth) value with actual thermal resistance limit:

Calculated Rth:

$$Rth = (\theta_a - \theta_A \div P_L)$$

NOTE:
Rth (Apparent) should be less than or equal to Rth (motor specification). See below

$$P_L = I_a \times (v - (0.7 AxK_T \times N)$$

Specified values of Rth and KT as a function of motor model:

MOTOR	Rth=	KT=
MRB231	4.0	17.1
MRB341	2.8	19.75
MRB342	2.1	20.35
MRB401	1.09	27.20
MRB402	0.9	51.36

Symptom	Probable Cause	Corrective Action
Motor starts, travels about a half turn, and faults out.	1. Motor leads wired to drive thermal blocks are reversed.	1. Reverse lead wires
	2. Encoder channels miswired.	2. Check encoder wiring. Verify that it is correct pr installation instructions in chapter 3 of this manual.

Symptom	Probable Cause	Corrective Action
Motor accelerates for a short period of time, and then faults out.	1. The motion profile is not optimum for the motor/drive vs. load configuration.	1. If you are accelerating the load too quickly for the torque capabilities of the motor, decrease the acceleration rate. If you are stalling at or near top speed in a high profile, you probably are running the motor at a speed at which its torque drops off, and you should try reducing that top speed. You may be starting at a step pulse rate which is too high or too low. Too high a starting rate could exceed the drive/motor's torque capabilities.
	2. The actuator or load may be jammed.	2. Remove the motor from the actuator by removing the coupler from the motor shaft and removing the motor flange screws. Try running the motor through its profile without a load. Also, check the actuator input screw shaft for excessive torque.
	3. Current trip potentiometer is set too low.	3. Reset the current trip potentiometer per the instructions given in Chapter 3 of this manual.

Symptom	*Probable Cause*	*Corrective Action*
Motor shaft rotates but actuator carrier does not move	1. Coupling which connects actuator to motor shaft has become loose	1. Tighten jaw style coupling through the access hole located on motor mount spacer.
	2. Main or reduction belt break on belt-drive actuators or reverse parallel screw actuator.	2. Confirm and consult distributor
	3. Two tensioning screws on belt-drive actuator have loosened causing belt to slip around drive wheel.	3. Refer to assembly/ disassembly instructions.

Symptom	*Probable Cause*	*Corrective Action*
Motor starts, travels about a half turn, and faults out.	Damping and/or Gain potentiometers not at optimum settings.	Verify that the Gain potentiometer is set to maximum, per the instructions given in Chapter 3 of this manual. Adjust the Damping potentiometer up or down until the instability lessens.

Symptom	*Probable Cause*	*Corrective Action*
Excessive motor brush wear	1. Check for correct motor brush kit installed.	1. Check brushes and brush holder slots for proper size and or spring tension. Replace with correct brush kit after removing excess carbon buildup from existing brushes.
	2. Application requirements exceeding motor/drive rating.	2. Reduce one or all of these application requirements: Load, speed, accel/decel rate and duty cycle.

Symptom	Probable Cause	Corrective Action
Motor shutters or shakes during some part of move. Current trip fault may occur.	1. Too much gain	Reduce gain. Also adjust damping for best stability. (5-10 from full CW is best in most cases.

TOL-O-MATIC, INC.
3800 County Road 116
Hamel, MN 55340
612.478.8000 Telephone
612.478.8080 Fax
http://www.tolomatic.com

Axidyne®