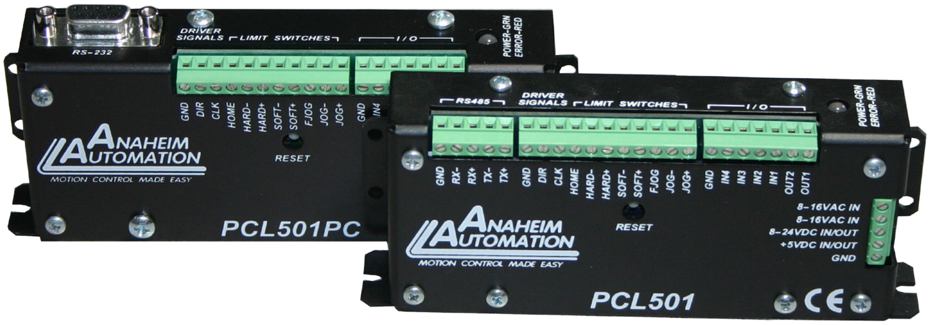
# PCL501 / PCL501PC
# Programmable Step Motor Controller

## User's Guide



ANAHEIM AUTOMATION

A N A H E I M   A U T O M A T I O N

4985 E Landon Drive, Anaheim, CA 92807
e-mail: info@anaheimautomation.com

(714) 992-6990  fax: (714) 992-0471
website: www.anaheimautomation.com

L010121

July 2018

# Table of Contents

# Section 1: Introduction

The PCL501(PC) is an economical single axis step motor controller containing 2 Kbytes of non volatile stored programming space. The user can write the program and then have the PCL501(PC) autostart the program on power up. It provides flexible, independent control of step motors from computers or any machine controller with a serial port. It is also capable of stand-alone operation making it an embedded machine controller. The easy to use Windows software, SMC50WIN, can be used to directly control motion and to program the PCL501(PC). The PCL501(PC) also has the ability for real time functions. A “Direct Mode” is used to directly control motion for Real Time Movements through serial communication. The PCL501(PC) has 20 commands which are easy-to-remember for direct movement of a step motor. The PCL501 communicates via RS485 communication and up to 32 PCL501s can be networked from one communications port on your PC or PLC. The PCL501PC is for single unit communication. It is a RS232 version of the PCL501. The PCL501(PC) has 2 programmable open drain outputs and 4 TTL compatible inputs and can be powered with DC (5-24VDC) or AC (8-16VAC) voltages.

## Description

The PCL501(PC) controller is an open-loop system, meaning that no information is sent back to the PCL501(PC) from the motor to verify the number of steps that were taken. A step motor is essentially a digital device - you give the step motor driver 10 step pulses, and the motor moves 10 steps.

The PCL501 is designed to communicate over a RS485 bidirectional serial data bus.

The PCL501(PC) provides independent programming of acceleration/deceleration, base speed (start up speed), max speed (running speed), and the number of steps to be taken in both relative and absolute positioning modes. On absolute positioning moves, the PCL501(PC) automatically determines the proper direction to go and the number of steps to take. The relative positioning will move a number of steps in the direction that the user defines. The PCL501(PC) has a high level command set including: looping, conditional statements, time delays, and I/O.

Hard, soft, and home limit switch inputs are provided for each axis. These features are generally required in most machine control designs. 4 testable inputs and 2 programmable outputs are provided per axis. These I/O may be used for monitoring and controlling machine operation and/or interaxis coordination. These I/O are accessible independent of the busy state of the axis controls. The 4 inputs are TTL/CMOS compatible. The 2 outputs are current sinking, open drain FETs.

The PCL501(PC) has a built-in programmable reset circuit. Reset is automatic on power-up or by pressing the external reset button. A CD is provided when you purchase the unit. This CD allows you to write and change programs that are to be stored in the PCL501(PC) for autostart use. This CD also allows you to save the programs onto your computer hard drive, and easily retrieve them when needed.

## Technical Support

Everyone needs help on occasion. If you have problems using any of the equipment covered by this manual, please read the manual to see if it will answer the questions you have. Be sure to look in the troubleshooting section located near the back of this manual. If you need assistance beyond what this manual can provide, you can call the factory direct for application assistance. If possible, have this manual in hand. It is often helpful to have the unit connected to a computer with the software installed.

## Ordering Information

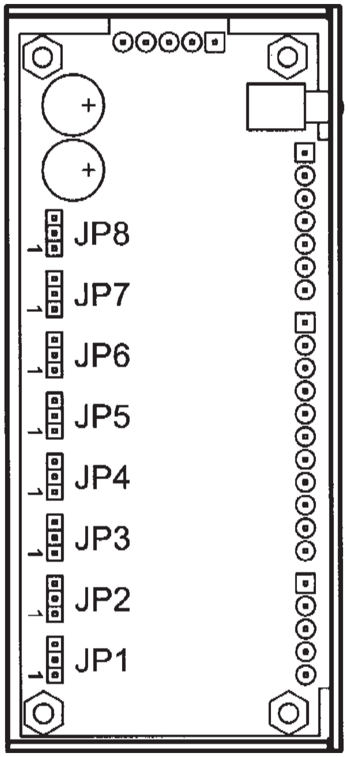
The table below lists a variety of products available from Anaheim Automation. These products include those covered by this manual along with supporting cables and devices. We are continually adding new products to our line, so please consult your nearest authorized Anaheim Automation distributor or representative for information on the latest releases.

| Part Number | Description |
|---|---|
| PCL501 | RS485 compatible controller |
| PCL501PC | RS232 compatible controller (Not multidrop compatible) |
| DPC40501 | Controller/drive pack - Features a 4 amp bipolar drive and power supply |
| 485SD9TB | RS232 to RS485 converter |
| PSAM24V1.2A-5V3.5A | Power supply for PCL501 and PCL501PC (24V@1.2A, 5V@3.5A) |

## Axis Selection

Each PCL501 will be addressed using 5 jumpers (JP4 - JP8) allowing the PC to address up to 32 PCL501s from one port. The jumpers are considered ON (1) when they are in position “1-2” and OFF (0) when they are in position “2-3”. To access the jumpers, remove the inner two screws on the bottom of the front cover and the two screws on the top of the PCL501(PC). Remove the back plate from the cover (The circuit board will still be mounted inside the cover). The jumpers are located on the back of the circuit board. You do not have to remove the board from the cover to access the jumpers. The table below shows how to configure the jumpers for a given axis. JP8 is the LSB (Least Significant Bit) and JP4 is the MSB (Most Significant Bit). The PCL501PC has no axis selection, it’s axis is always 0.

| Axis Selected | Jumper Setting | Axis Selected | Jumper Setting |
|---|---|---|---|
| 0 | 00000 | 16 | 10000 |
| 1 | 00001 | 17 | 10001 |
| 2 | 00010 | 18 | 10010 |
| 3 | 00011 | 19 | 10011 |
| 4 | 00100 | 20 | 10100 |
| 5 | 00101 | 21 | 10101 |
| 6 | 00110 | 22 | 10110 |
| 7 | 00111 | 23 | 10111 |
| 8 | 01000 | 24 | 11000 |
| 9 | 01001 | 25 | 11001 |
| 10 | 01010 | 26 | 11010 |
| 11 | 01011 | 27 | 11011 |
| 12 | 01100 | 28 | 11100 |
| 13 | 01101 | 29 | 11101 |
| 14 | 01110 | 30 | 11110 |
| 15 | 01111 | 31 | 11111 |



## Baud Selection

The baud rate is selected using 3 jumpers (JP1 – JP3). They are selected the same way that the address jumpers are selected. The table below shows how to configure the jumpers for a given baud rate. JP3 is the LSB and JP1 is the MSB.

| Baud Rate | Jumper Setting |
|---|---|
| 1200 | 000 |
| 2400 | 001 |
| 4800 | 010 |
| 9600 | 011 |
| 19200 | 100 |
| 38400 | 101 |
| 57600 | 110 |
| 115200 | 111 |

## Baud Rates

A term used frequently in serial data communications. A “baud” is defined as the reciprocal of the shortest pulse duration in a data word signal, including start, stop, and parity bits. This is often taken to mean the same as “bits per second”, a term that expresses only the number of “data” bits per second. Very often, the parity bit is included as an information or data bit. The PCL501(PC) accepts the following baud rates:

1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

## Methods of Communication

There are two methods for sending commands to the PCL501(PC). One is to directly talk to the PCL501(PC) by using Direct Talk Mode. This is usually used with a computer or PLC (Programmable Logic Controller), where the computer or PLC gives the PCL501(PC) serial commands to off-load its processor. For example: A PLC can utilize its outputs to toggle the PCL501(PC)’s inputs and gain control of variable speeds, variable programs, variable distances, etc. Simply using the PCL501(PC) as the intelligent pulse generator a PLC can remove some of the tasks that were not meant for ladder logic or any PLC processing time. The second way to give commands to the PCL501(PC) is to use the software program SMC50WIN to either manually control or to write and send programs. This method is used when the PCL501(PC) is the main controller. For example: A PCL501(PC) can replace simple motion control and replace I/O functional when minimal quantities of I/O are required to control specific machinery. Simple motion profiles that can operate with 4 or less inputs and 2 or less outputs can utilize a PCL501(PC) controller.

## RS232 to RS485 Protocol

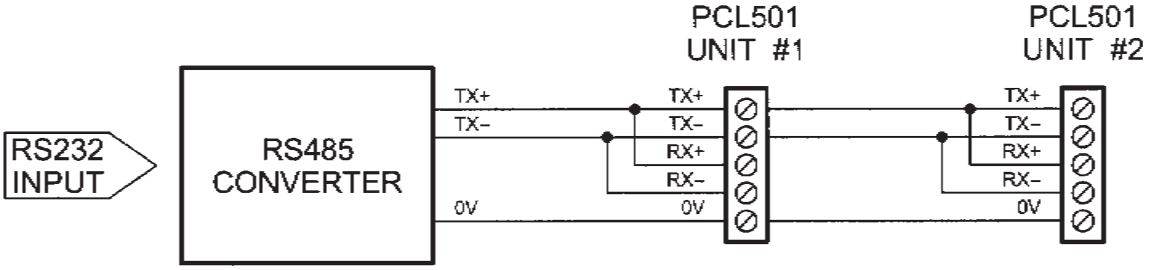
The PCL501 can be connected to your PC serial port via an RS485 or RS422 converter box. The RS232 converter will convert the RS232 communication format to the RS485 or RS422 format. Only one converter box is needed per serial port. Contact the factory for RS485 converter information and sales.

## RS485 Protocol

The RS485 protocol is as follows, on board receivers will remain in active mode indefinitely. Transmitters must be turned off when the unit is not sending data to prevent the line from sending and receiving data at the same time. Therefore when the PC is transmitting data its driver will be turned on and each of the units connected will have their drivers off. If they are requested to send data back to the PC, the selected unit will turn it's driver on to send the data then turn it off after it has completed transmission. Note: The above protocol is done internally between the converter and the PCL501. The RS485 method of communication allows increased noise immunity and increased communication distance of up to 4000 feet without repeaters. RS485 repeaters allow an additional 4000 feet per repeater.

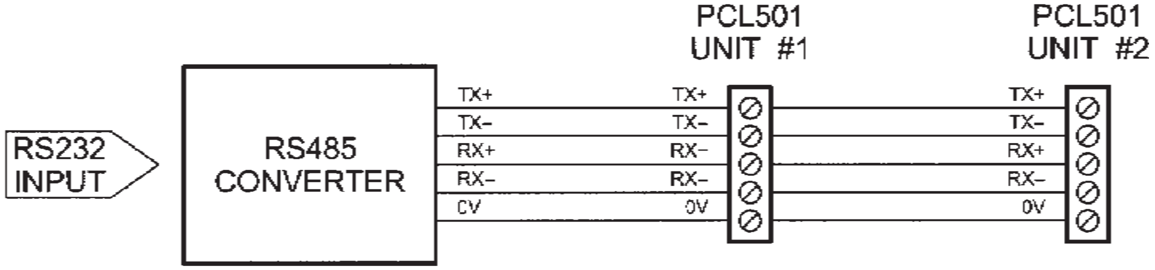
## Two Wire Configuration

The two wire configuration reduces cabling costs by requiring only three wires. A, B and ground. The PCL501 is designed to allow either the two or four wire configuration. To use the 2 wire configuration simply wire TX+ to RX+, and TX- to RX- on your converter box. Then run a wire from ground, a wire from TX+/RX+ and a wire from TX-/RX-, to the first PCL501 in the network. Finally do the same on the terminal block of the PCL501 as the converter box. The diagram below illustrates how this configuration is connected. RS422 systems require a dedicated pair of wires for each signal, a transmit pair, a receive pair, and an additional pair for each handshake/control signal used (if required). The tristate capabilities of RS485 allow a single pair of wires to share transmit and receive signals for half duplex communications. This "two wire" configuration (note that an additional ground conductor should be used) reduces cabling cost. RS485 devices may be internally or externally configured for two wire systems. Internally configured RS485 devices simply provide "A" and "B" connections (sometimes labeled "-" and "+").



## Four Wire Configuration

Devices configured for four wire communications use TX and RX connections for both the transmit and the receive pairs. The user can connect the transmit lines to the receive lines to create a two wire configuration flexibility. Note that the signal ground line should also be connected in the system. This connection is necessary to keep the VCM common mode voltage at the receiver within a safe range. From the diagram below, it can be seen that all wires are run directly from the converter to the PCL501. For example TX+ from the converter goes to TX+ on the PCL501 and so on.



## Terminating Resistor

To eliminate noise on the transmission lines a terminating resistor may need to be used. If needed the termination resistor need only be added to the last (furthest from the converter box) PCL501 in the network. A termination resistor with a value of 120 ohms is needed in certain conditions; when using a 4000 ft. or longer cable and a baud rate of 38400 or when using a 2000 ft. or longer cable and a baud rate of 57600. If you need to add a terminating resistor, contact the factory for the exact location on the board.

## Status LED

When powered and operated properly, the status LED will be green. When an error occurs, the LED will change to RED and an error code will be generated in the error code register. To read and clear the error with the software, click on the “Verify Parameters” button located in the “Motion Tab”. To read and clear the error while in “Direct Mode” use the ! command. Once the error has been read and cleared, the LED will return to green and the error code register will be cleared. Refer to the table on page 30 for a list of the error codes.

## Interfacing With A Driver

The PCL501(PC) was designed to control a step motor driver. For drivers with opto-isolated inputs, the PCL501(PC) needs to sink current through the driver’s input LED. To do this, connect the +5VDC to the positive terminals of the inputs and connect the clock and direction outputs of the PCL501(PC) to the negative terminals of the corresponding inputs on the driver. For the ON/OFF input, one of the PCL501(PC) outputs can control this function. For a driver with TTL/CMOS inputs the PCL501(PC) has a negative going clock and will sink the current from the driver’s inputs. Wire the clock and direction outputs and 0VDC reference of the PCL501(PC) to the clock and direction inputs and 0VDC reference of the driver respectively. Once again an output of the PCL501(PC) can be used to control the ON/OFF function of the driver. For a PCL501(PC) bought with an integrated driver and power supply, these connections are already internally made.

## Electrical Specifications

**Power Requirements:**
5VDC @ 50mA or
8VDC to 24VDC @ 25mA or
8VAC to 16VAC @ 25mA

**Operating Temperature:**
0 to 60 degrees C

**Pulse Output Range:**
77 to 15000 pps

**Inputs (TTL-CMOS):**
Logic “0”: 0 to 0.8VDC
Logic “1”: 3.5 to 5.0VDC

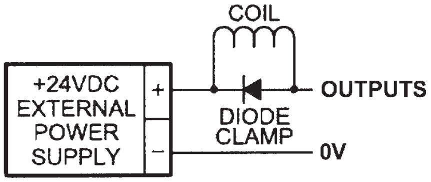
**Outputs (CLK,DIR):**
Open Drain Type
40VDC, 75mA

**Baud Rate:**
1200 to 115200 BAUD

**Data Format:**
Half-Duplex, 1 start bit, 8 data bits,
no parity, 1 stop bit

**Outputs (2 programmable I/O):**
Open Drain Type
40V, 75mA
+5VDC Output, 50mA

**Note:** For inductive loads, customers must connect a clamping diode to protect from fly back voltage spikes.



## Terminal Descriptions

**TB1: (PCL501 only)**

| Pin # | Description |
|---|---|
| 1 | TX+ |
| 2 | TX- |
| 3 | RX+ |
| 4 | RX- |
| 5 | RS485 Ground |

**TB4:**

| Pin # | Description |
|---|---|
| 1 | 8-16VAC IN |
| 2 | 8-16VAC IN |
| 3 | 8-24VDC IN |
| 4* | 5VDC IN/ OUT |
| 5 | 0VDC |

**TB2:**

| Pin # | Description | Comments |
|---|---|---|
| 1 | JOG+ | Active Low |
| 2 | JOG- | Active Low |
| 3 | FJOG | Active Low |
| 4 | SOFT+ | Active Low |
| 5 | SOFT- | Active Low |
| 6 | HARD+ | Active Low |
| 7 | HARD- | Active Low |
| 8 | HOME | Active Low |
| 9 | CLK | Open Drain |
| 10 | DIR | Open Drain |
| 11 | 0VDC | Reference |

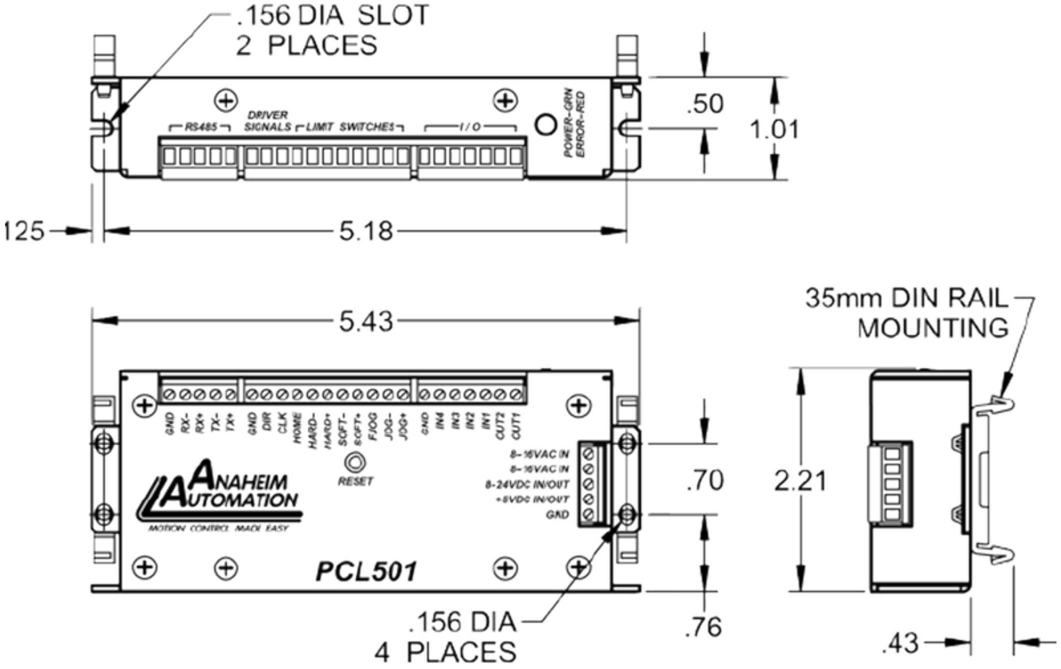
**TB3:**

| Pin # | Description | Comments |
|---|---|---|
| 1 | OUT1 | Open Drain |
| 2 | OUT2 | Open Drain |
| 3 | IN1 | Active Low = 1 |
| 4 | IN2 | Active Low = 2 |
| 5 | IN3 | Active Low = 4 |
| 6 | IN4 | Active Low = 8 |
| 7 | 0VDC | Reference |

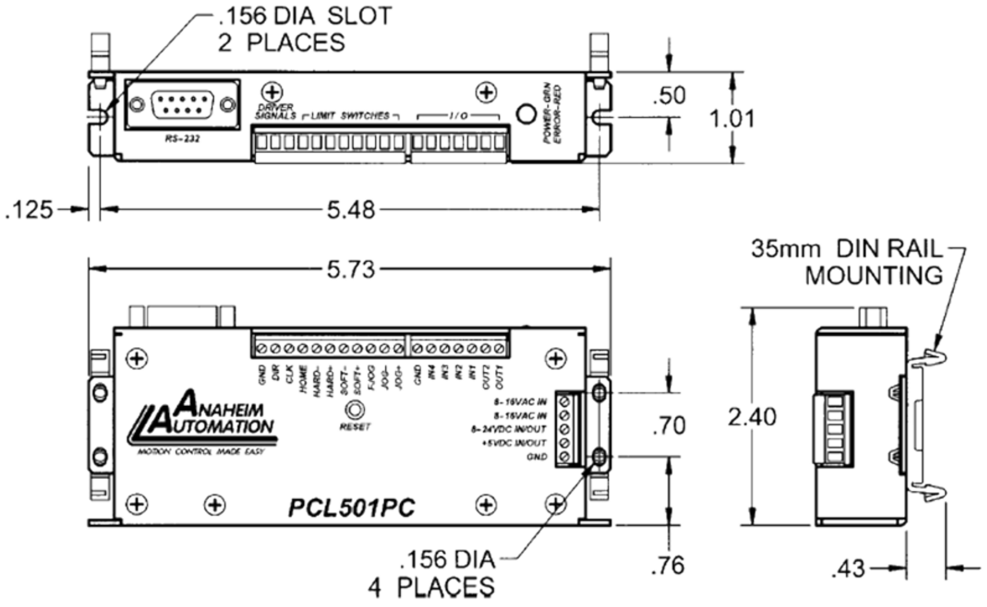
***Note:** Pin 4 on TB4 is an available 5V out at 50mA if power is supplied by the 8-24VDC or 8-16VAC.

# Dimensions

## PCL501

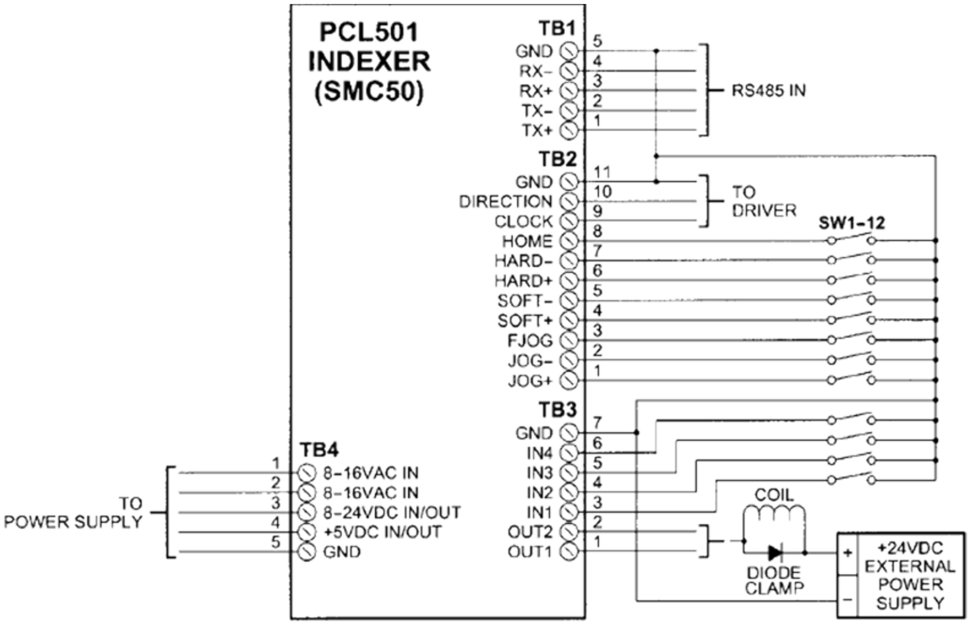


## PCL501PC

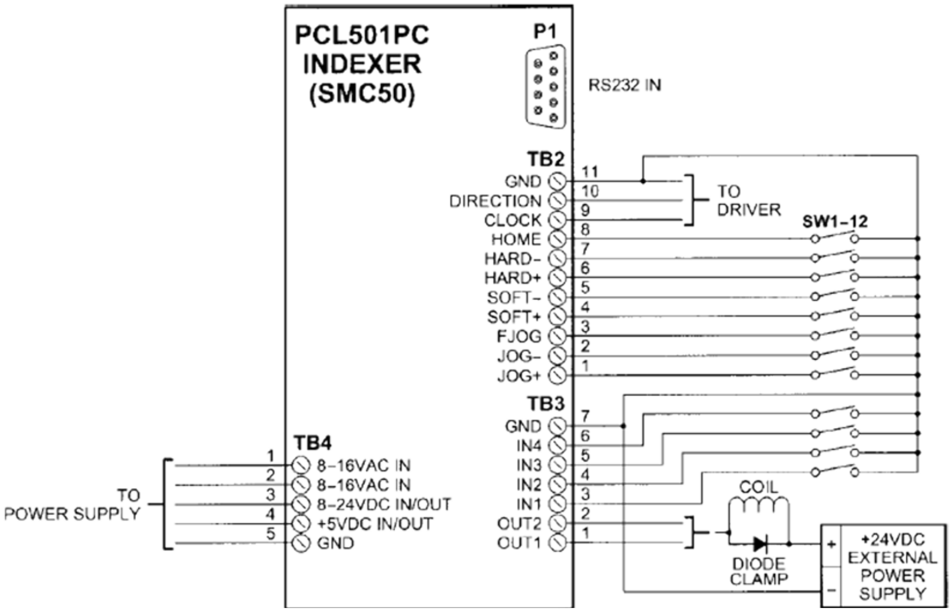


# Wiring Diagrams

## PCL501



## PCL501PC



# Section 2: Functions

**Move Number of Steps:** The move number of steps command causes the motion to start in the direction last specified. This command will move the motor the number of steps given. (Range: 0 to 8388607)

**Move to Position:** The move to position command specifies the next absolute position to go to. The PCL501(PC) automatically sets the direction and number of steps needed to go to that position.
(Range: -8388608 to +8388607)

**Home to Soft, Home Limit (2 Switch Operation):** This type requires two grounding type limit switches called home and soft. The first limit switch soft will decelerate the motor down to base speed. It will continue to run at base speed until it receives a home limit switch input causing the motor to stop. The home limit switch only activates after a soft limit is sensed. The soft limit is not bidirectional, meaning that it will work in only one direction as specified. The soft limit switch will work for any type of motion. The home limit switch will work only for home motions.

Note: Whenever a soft limit switch is activated, the motor will decelerate and run at base speed. Be sure to come back passed the soft limit switch to set any origins, otherwise the motor will decelerate as it goes passed the soft limit switch.

**Home to Home Limit**: This type of homing differs in that only one limit switch is needed. The home limit switch in this case causes the motor to ramp down to base speed, reverse direction and continue until the limit switch is released. This is a good way to compensate for any backlash in the system. It is also useful for minimizing the number of limit switches needed for homing.

**Limit Switch Inputs:** The limit switch inputs are internally pulled up by a resistor making them normally +5 volts. To activate the input, the pin must be grounded to (0VDC) on the terminal block.

**Hard Limit Switches:** When a hard limit switch is encountered, the motion will stop. The position counter will also cease counting. Hard limits are intended as an emergency stop for your system. It should not be used to do any indexing type functions.

**Soft Limit Switches:** These switches are used exclusively when homing to a datum point. If positioned properly with the appropriate parameters, it causes the motor to ramp down to the base speed before encountering the home limit switch.

**Home Limit Switch:** This switch is used to establish the reference position designated “home” in home to home limit or home to soft, home limit.

**Set Position:** The set position command sets the position register to a designated value. The number will be the new absolute position of the motor. The default value is 0. (Range: -8388608 to +8388607)

**Jog Inputs:** Jog is a manual function. The user can select the direction and speed (fast or slow) by grounding the appropriate combinations of inputs on a particular axis. These inputs are located on the terminal block. To jog a motor, it is necessary to ground the jog input on the axis for the direction desired. For fast jog, both the fast and jog command for the appropriate direction must be low at the same time. The closure of jog causes the motor to start at base speed and accelerate at a predetermined rate to jog speed. When the fast input is closed, the motor will then accelerate to a pre programmed speed of 10kHz. The actual jog rates can be programmed. Fast jog is not programmable. The position register will keep track of the number of steps that are taken during jogging. Once a +jog or a -jog function has been performed, the direction register will retain the last direction of movement; that is, a subsequent go command will be in the same direction as the last jog command.

**Programmable Inputs and Outputs:** Four inputs and two outputs are provided per axis. The inputs may be used to initiate a machine cycle, for inter-axis coordination (in stored program mode), for operator intervention, for sensing a machine condition such as out of stock, or to wait for temperature to be reached. Outputs may be used to operate coolant valve, air cylinders, relays, or, with the right interfacing, any electronically controlled device. The inputs are TTL compatible. Since the inputs have pull up resistors, all that is required for a signal is a switch closure to ground (0VDC). With zero volts on the input, the pull up resistor source current is approximately 5mA. This will make the inputs read like they are logiclly reverse. A grounded input will read a “1” and an open input will read a “0”. The outputs can drive all types of common peripheral power loads, including lamps, relays, solenoids, LED’s, printer heads, and heaters. For inductive loads, it will be necessary to connect a clamping diode (refer to specification section). The outputs are current sinking, open drain FETs. They are capable of sinking up to 75mA per output with voltages up to 40VDC. Turning an output on will pull the pin to ground and turning an output off will make the pin open.

Note: For inductive loads, customers must connect a clamping diode in order to provide adequate fly-back protection. Input wiring should be kept separate from step motor wiring.

**Slew:** The slew command will accelerate the motor up to maximum speed and continue to run at the speed until reaching a hard limit switch, soft limit switches, or receiving a “.” (stop hard) command.

**Finish Move:** When writing a program, the finish move command is used directly after a motion command. With this command, the PCL501(PC) will see a busy signal until the move is complete before executing any further commands. Unless the finish move command is used, the PCL501(PC) will keep on executing commands, even though the PCL501(PC) is not ready to use it. This data will be ignored by the PCL501(PC), so the program will not work as expected.

**Run:** The run command starts the execution of a stored program.

**Quit:** The quit command, used within a stored program, stops execution of the program. This command must be used at the end of all programs.

**Wait:** In stored program mode, the wait command pauses the program for the specified number of milliseconds. (Range: 1 to 9999)

**Verify:** The verify command causes the PCL501(PC) to send data back to the PC or PLC. The data is sent as an ASCII decimal string followed by a carriage return and a line feed. The permissible verify commands are shown below.

| Command | Description |
|---|---|
| A | Verify acceleration/deceleration |
| B | Verify base speed |
| F | Verify if controller is busy |
| J | Verify jog speed |
| M | Verify max speed |
| N | Verify number of steps |
| O | Verify outputs |
| P | Verify goto position |
| Z | Verify position |
| + | Verify direction (1 is CW, 0 is CCW) |

**Loop:** The loop instruction allows the user to loop a program a variable number of times. The program will loop to the designated address location of the program. The address must always be a lower address value than the instruction itself. No nested loops are allowed.

**Acceleration/Deceleration:** The acceleration and deceleration are by default the same value. This function controls the time that the motor will take to move from base speed to max speed. The higher the value, the slower the motor will accelerate. The same principal applies for the deceleration which is controlling the time it takes to go from maximum speed to base speed. The higher the value, the slower the motor will decelerate. (Range: 1 to 255)

**Base Speed:** The base speed is the speed at which motion starts and stops. It is entered directly as the number of steps per second. This speed must always be less than the max speed. (Range: 77 to 3500)

**Max Speed:** The max speed is the top speed the user wants the motor to run at. This speed must always be greater than the base speed. It is entered directly as the number of steps/second.
(Range: 77 to 15000)

**Jog Speed:** The jog speed sets the slow jog rate. Jog (+/-) can also be used in conjunction with the FJOG pin. The FJOG pin, when grounded, will ramp the motor to 10kHz. This speed must always be greater than the base speed.

# Section 3: SMC50WIN Software

The SMC50WIN software is a handy utility that supports Anaheim Automation's line of PCL501(PC) and PCL511(PC) step motor controllers. Connecting your PC to the PCL501(PC), via a serial cable, the SMC-50WIN software can easily perform the following tasks:

- Exercise and monitor the PCL501(PC)
- Write and edit stored programs for stand-alone operation
- Directly communicate with the PCL501(PC)

The software will automatically switch between the PCL501(PC) and PCL511(PC) screens when the controller is connected. The default screens are for the PCL501(PC) when no controller is connected.

## Installation

### Software

- The SMC50WIN is supplied on a CD, containing the setup program and the SMC-50WIN software
- SMC50WIN is compatible with all versions of Windows including Windows 2000 and Windows XP

### Windows 3.x Installation

1) Insert the CD into the drive
2) From the Program Manager select **File | Run**
3) Enter **D:\setup** and click **OK** - use the appropriate drive letter (i.e. **D** or **E**)

### Windows 95/98/NT/ME/2000/XP Installation

Option 1

1) Insert the CD into the drive
2) On the Windows Taskbar select **Start** | **Run**
3) Enter **D:\setup** and click **OK** - use the appropriate drive letter (i.e. **D** or **E**)

Option 2

1) Open Windows Explorer
2) Open CD Drive Folder (D: or E:)
3) Double click the Setup Icon

## Getting Started

1) Double click on the SMC50WIN icon to run the SMC50WIN software.

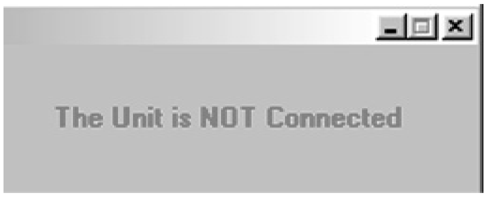
2) Apply power to the PCL501(PC) unit.

3) Set the appropriate communication setting by selecting Setup | Communication Setting from the menu bar.

4) Establish communications with the PCL501(PC) by clicking on the Connect Icon, or select Setup | Connect. If the unit is connected properly, the program will notify you when communications has been established.

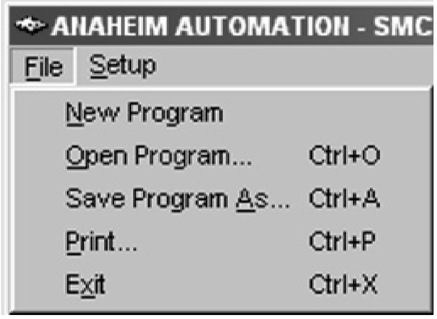
## “The Unit is Connected” / “The Unit is NOT Connected”

On the right of the Toolbar, the user will find the communication status of the PCL501(PC). If communications is not established, please refer to the troubleshooting section.



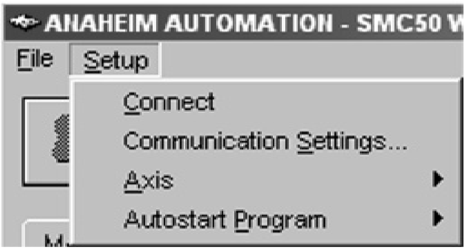
The Unit is NOT Connected

## File Menu



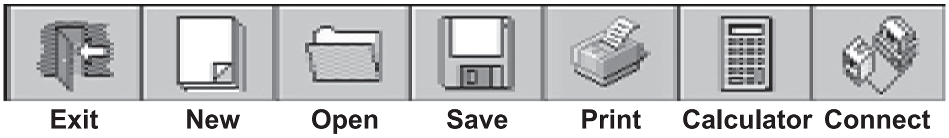
| | |
|---|---|
| New Program | Start a new program. |
| Open Program... | Open existing program. |
| Save Program As... | Save current program. |
| Print... | Print current program. |
| Exit | Exit the SMC50WIN software. |

## Setup Menu



| | |
|---|---|
| Connect | Establish communications with the controller. |
| Communication Settings... | COM port & baud rate settings. |
| Axis | Select axis (0-31) for multi drop units. |
| Autostart Program | Enable/disable program execution on power up. |

## Toolbar

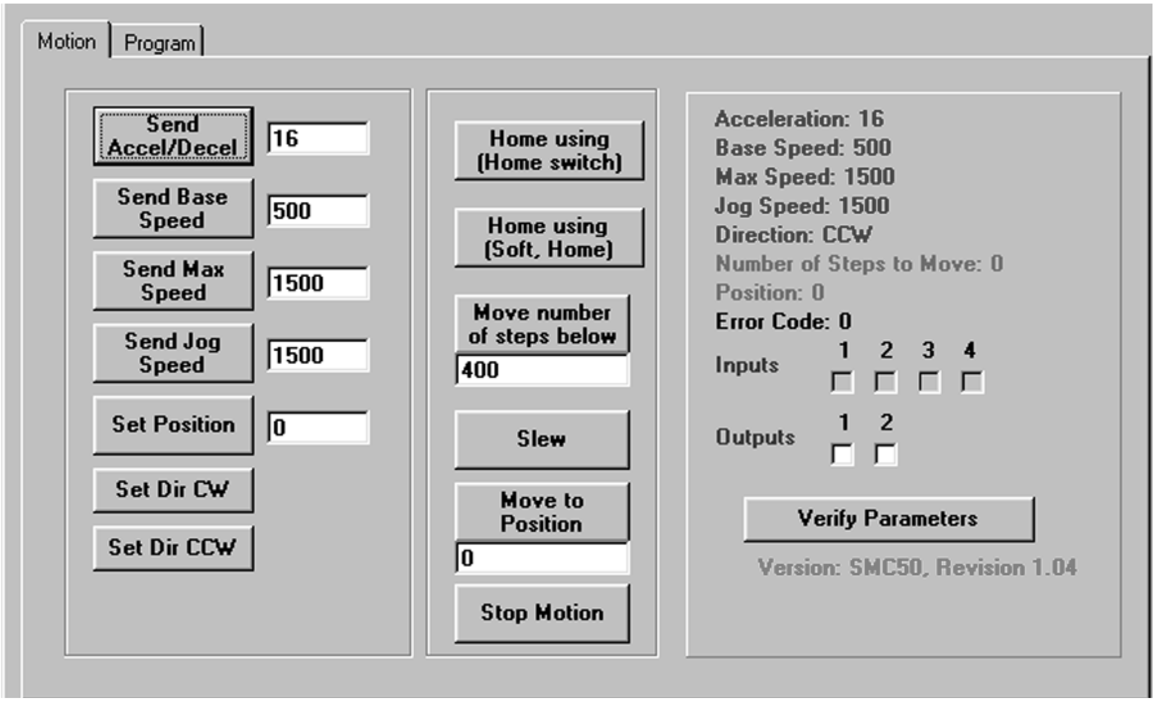


| | |
|---|---|
| Exit | Exit the SMC50WIN software. |
| New | Start a new program. |
| Open | Open an existing program. |
| Save | Save the current program. |
| Print | Print the current program. |
| Calculator | Desktop calculator. |
| Connect | Establish communication with the controller. |

## Tab Sheets

| | |
|---|---|
| Motion | Controls and executes motion on the controller. |
| Program | Write and modify PCL501(PC) stored programs. |

## Motion Tab Sheet



Send Accel/Decel 16

Send Base Speed 500

Send Max Speed 1500

Send Jog Speed 1500

Set Position 0

Set Dir CW

Set Dir CCW

Move number of steps below 400

Move to Position 0

Acceleration: 16
Base Speed: 500
Max Speed: 1500
Jog Speed: 1500
Direction: CCW
Number of Steps to Move: 0
Position: 0
Error Code: 0

Inputs 1 □ 2 □ 3 □ 4 □

Outputs 1 □ 2 □

Version: SMC50, Revision 1.04

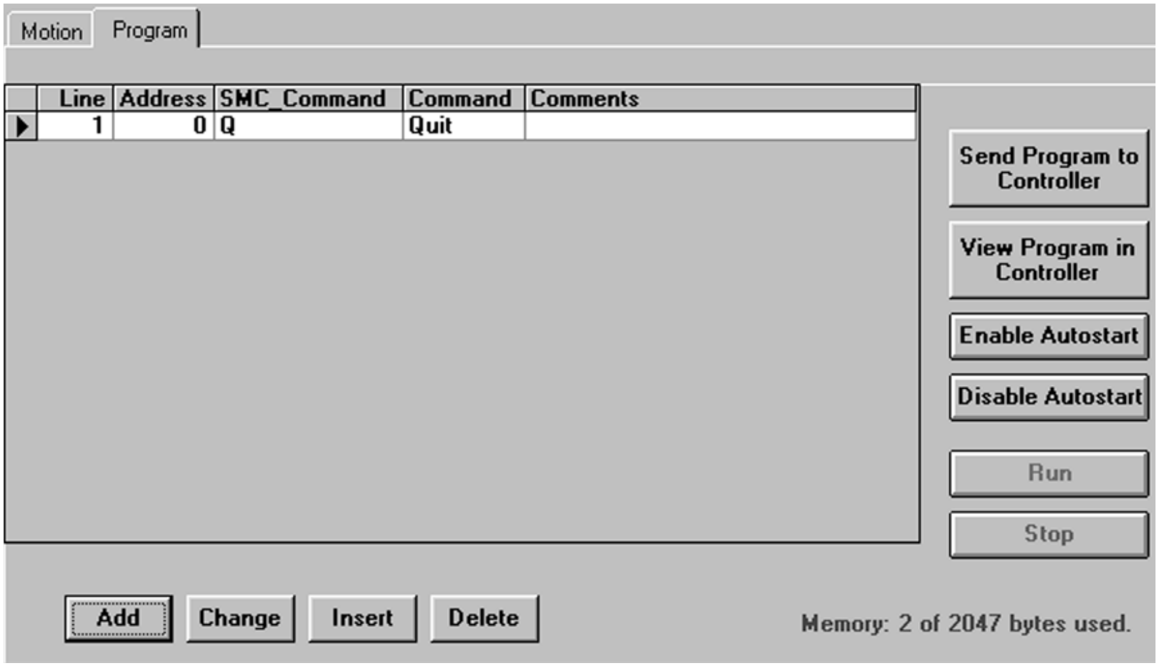
| | |
|---|---|
| Send Accel/Decel | Send acceleration & deceleration parameter to controller. (step/sec$^2$) |
| Send Base Speed | Send base speed parameter to the controller. (step/sec) |
| Send Max Speed | Send maximum speed parameter to the controller. (step/sec) |
| Send Jog Speed | Send jog speed parameter to the controller. (step/sec) |
| Set Position | Set motor position. |
| Set Dir CW | Set direction to clockwise. |
| Set Dir CCW | Set direction to counter-clockwise. |
| Home using (Home Switch) | Motor will seek the home position by moving towards home switch wich will stop the motor, reverse the motor direction and stop when the home limit switch is no longer triggered. (One switch is required to stop anti-backlash) |
| Home using (Soft, Home) | Motor will seek the home position by moving towards home switch but motor will slow down to base speed when the soft switch is triggered, followed by triggering the home switch to stop motion. (Two switches are required to stop) |
| Move number of steps below | Motor will move number of steps entered. |
| Slew | Motor will ramp up to maximum speed and keep moving until stop motion is triggered. |
| Move to Position | Motor will move to specified position. |
| Stop Motion | Stop any motor motion. |
| Inputs | View inputs. (checked = ON, blank = OFF) |
| Outputs | View and trigger outputs. (checked = ON, blank = OFF) |
| Verify Parameters | Updates and displays controllers parameters and resets the error codes. |

## Motion Tab Sheet Tutorial

This tutorial will demonstrate the motion tab sheet:

1. Start the SMC50WIN software and power up the PCL501(PC).
2. Click the connect icon and establish communications with the PCL501(PC).
3. With the motion tab sheet displayed.
4. Enter 400 for the “Move number of steps below” button.
5. Click the “Move number of steps below” button, the motor should move 400 steps - 1 revolution on a 200 steps/rev motor running in half step mode.

## Program Tab Sheet



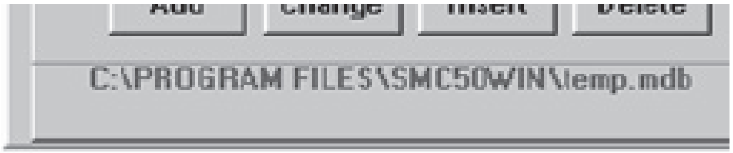
| Line | Address | SMC_Command | Command | Comments |
|---|---|---|---|---|
| 1 | 0 | Q | Quit | |

Memory: 2 of 2047 bytes used.

| | |
|---|---|
| Send Program to Controller | Send current program to the controller |
| View Program in Controller | View program in controller memory. |
| Enable Autostart | Program will start when controller is powered up. |
| Disable Autostart | Program will only execute when run is clicked. |
| Run | Execute the program in the controller memory. |
| Stop | Abort program execution. |
| Add | Adds a new line of code to the end of the program. |
| Change | Edits the currently selected line of code. |
| Insert | Insert a new line of code before the currently selected line of code. |
| Delete | Deletes the currently selected line of code. |

## Current Program Filename

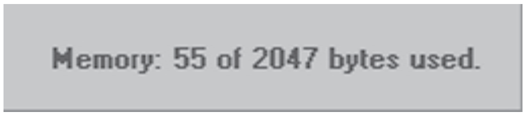
With the program tab sheet selected the user can obtain the current program filename, located in the lower left corner of the SMC50WIN window. All programs created by the SMC50WIN software will have a .mdb extension.



C:\PROGRAM FILES\SMC50WIN\temp.mdb

## SMC50 Memory Available

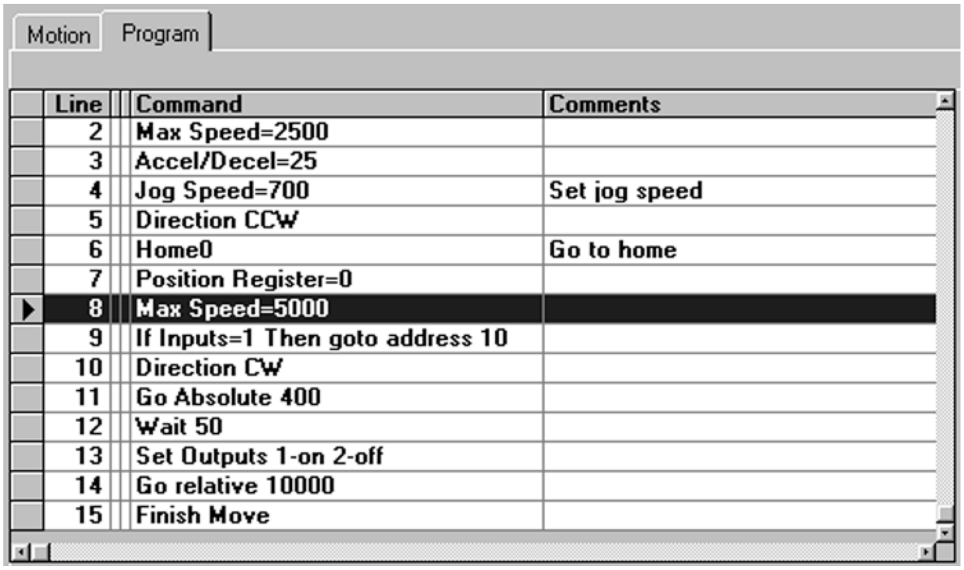
With the program tab sheet selected the user can obtain the amount of available memory, located in the lower right corner of the SMC50WIN window. The PCL501(PC) has a maximum available memory of 2047 bytes - each instruction can use from 1 to 5 bytes.



Memory: 55 of 2047 bytes used.

## Currently Selected Line

The currently selected line is indicated in the program by the right pointing arrow/triangle in the left column.



| | Line | Command | Comments |
|---|---|---|---|
| | 2 | Max Speed=2500 | |
| | 3 | Accel/Decel=25 | |
| | 4 | Jog Speed=700 | Set jog speed |
| | 5 | Direction CCW | |
| | 6 | Home0 | Go to home |
| | 7 | Position Register=0 | |
| ▶ | 8 | Max Speed=5000 | |
| | 9 | If Inputs=1 Then goto address 10 | |
| | 10 | Direction CW | |
| | 11 | Go Absolute 400 | |
| | 12 | Wait 50 | |
| | 13 | Set Outputs 1-on 2-off | |
| | 14 | Go relative 10000 | |
| | 15 | Finish Move | |

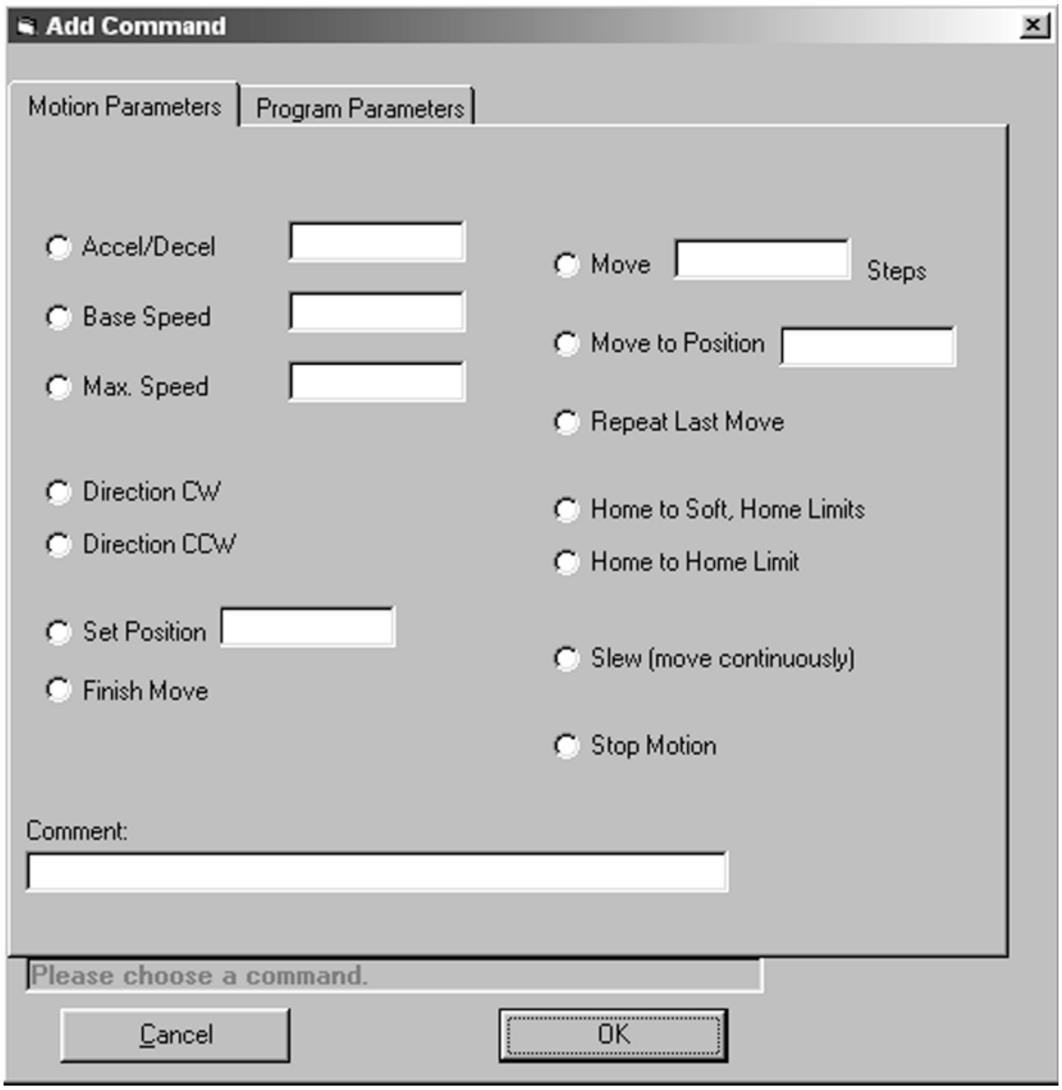
Clicking on any line will select a new currently selected line.

## Add/Change/Insert Commands

Add command contains 2 different tab sheets, which are Motion Parameters and Program Parameters.

| | |
|---|---|
| Motion Parameters | Software section that allows user to enter speeds, positions, direction, etc. |
| Program Parameters | Software section that allows user to manipulate looping routines, I/O, delays, etc. |

## Motion Command Tab Sheet



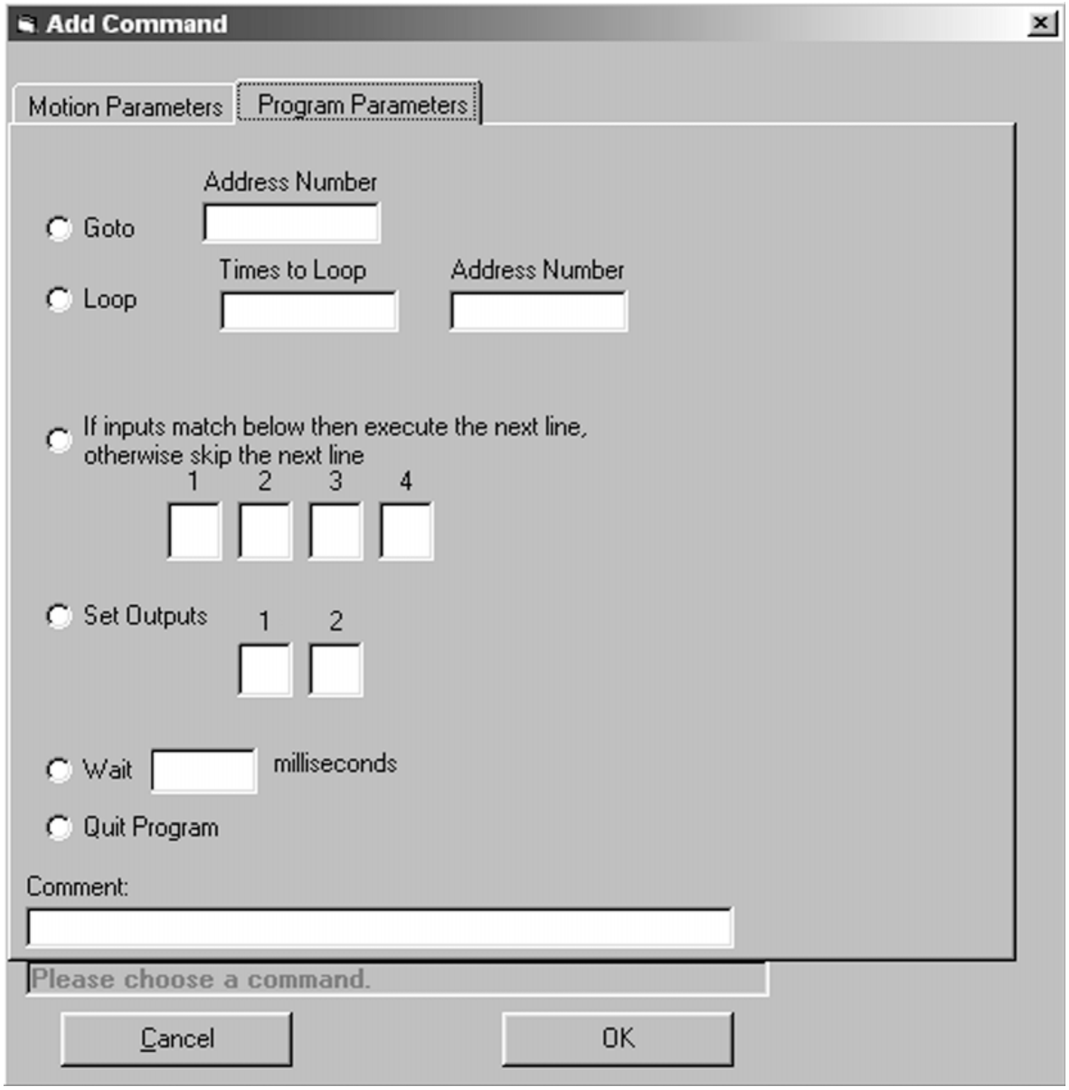
Please choose a command.

The motion command tab sheet controls the motion information of the motor such as Acceleration/ Deceleration, Maximum Speed, Base Speed, Move number of Steps, and etc.

- It works similar to the **Motion Tab Sheet** explained above in the Getting Started section.
- To add a line of motion control, select appropriate motion control from the list, then enter the required value for that particular action. Then, click **OK**.
- Comment is optional, for any lines of code.
- The text box above the **OK** and **Cancel** buttons will display useful information about each command.

| | |
|---|---|
| Accel/Decel | Set program acceleration and deceleration parameter. (step/sec$^2$) |
| Base Speed | Set program base (start) speed rate. (step/sec) |
| Max. Speed | Set program maximum (running) speed rate. (step/sec) |
| Set Position | Set motor position. |
| Move ________ Steps | Relative move command will allow motor to move the defined number of steps entered. |
| Move to Position | Absolute move command will move motor to the position specified. |
| Slew (Move Continuosly) | Command will initiate motion in the direction specified unitl a limit switch or hard stop is triggered. |
| Repeat Last Move | Command will repeat the previous motion command. |
| Home to Soft, Home Limits | Command will initiate motion in the direction last entered seeking the soft input first to slow the motor down to base speed, then to stop when the home limit is triggered. |
| Home to Home Limit | Command will initiate motion in the direction last entered seeking the home limit which will stop the motor, reverse the motor direction and stop when the home limit switch is no longer triggered. |
| Finish Move | Command will allow any motion command to be completed before continuing. This command must be entered after every motion command. |
| Stop Motion | Command will stop all motion of the motor. |
| Direction CCW | Sets motor direction for counter-clockwise movement. |
| Direction CW | Sets motor direction for clockwise movement. |
| Comment: | Comments are always useful instructions or debugging methods to help keep the user informed as to what the program is executing at the specific line. |

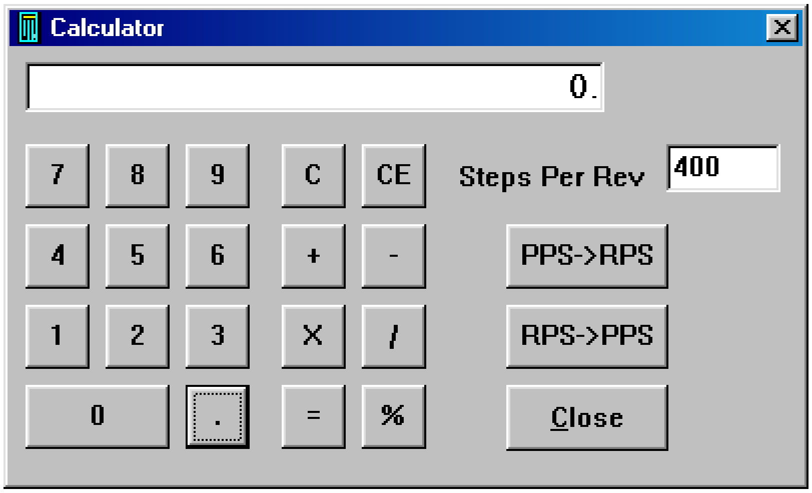
## Program Parameters



Please choose a command.

| | |
|---|---|
| Goto | The Goto command allows program to jump to specified address. |
| Loop | The loop command allows a sequence of commands to be looped a specific number of times to an address, which is lower in value. No nested loops are allowed. |
| If inputs match below then execute next line, otherwise skip next line | This conditional command allows the user to go to a specified address in the program if the inputs triggered match the selected or checked input box. If the inputs (1 through 4) do not match the next line is skipped. |
| Set Outputs | The outputs can be turned (on=1) or (off=0). These outputs can be used to trigger PLC operations, relays, solenoids, etc. |
| Wait ______ milliseconds | This command allows the user to enter a delay in milliseconds. This wait command is useful for pausing the program from reading the next command. |
| Quit Program | The quit command is a required function used **only** at the end of the program. |
| Comment: | Comments are always useful instructions or debugging methods to help keep the user informed as to what the program is executing at the specific line. |

# Calculator



0.

Steps Per Rev 400

| | |
|---|---|
| PPS->RPS | Convert from pulses per second to revolution per second. |
| RPS->PPS | Convert from revolution per second to pulses per second. |
| Steps Per Rev | Enter the number of steps per revolution of the step motor. The default is for a 200 step/rev motor in half step which is equal to 400. |
| Close | Exit Calculator |

# Section 4: Direct Talk Mode

Direct mode is used to directly control motion for real time movements through serial communication. The PCL501(PC) has 20 commands which are easy to remember for direct movement of a step motor.

## COM Port Settings

| | |
|---|---|
| **Baud Rate:** | Select one from the Baud Rate Selection chart in section 1. |
| **Data Bits:** | 8 |
| **Parity:** | None |
| **Stop Bits:** | 1 |
| **Flow Control:** | Xon/Xoff |

## Unit Selection

In order to select a unit the @ command followed by the address of the unit must be sent.
NOTE: There should be no spaces between the @ and address select.

**How to select a unit:**

| | |
|---|---|
| @0 | (Unit 0 is selected) |
| @1 | (Unit 1 is selected) |
| @29 | (Unit 29 is selected) |

**How to get a response from a unit:**

| | |
|---|---|
| @0$ | (Carriage Return) |

After the $ command, the PCL501(PC) will return a SMC50 + the current version number.
Note: In direct talk mode each command is followed by a carriage return.

The unit communicates in half duplex mode, therefore proper setup of hyper terminal is necessary to view characters, if characters are to be echoed back to the screen.

## Instructions

All instructions require that no spaces be sent between the command and the parameter followed by a carriage return.

(@0 not @ 0)

**Command Summary:**

**A** - Acceleration/Deceleration
**B** - Base Speed
**G** - Go Number of Steps
**H** - Home
**I** - Read Inputs
**J** - Jog Speed
**L0** - Get Limit Status
**LS** - Soft Limit Input Bit
**M** - Max Speed
**N** - Number of Steps
**O** - Set Outputs
**P** - Absolute Position
**S** - Go Slew
**V** - Verify
**Z** - Position
**.** - Stop Motion
**+** - Clockwise Direction
**-** - Counterclockwise Direction
**$** - Version Number Register
**!** - Error Codes Register

### $ - Version Number Register

Format: $

Description: This command requests the PCL501(PC) to return the version number.

### ! - Error Codes Register

Format: !

Description: This command requests the PCL501(PC) to get the current error code and print it to the screen.

### A - Acceleration/Deceleration

Format: A [value]

Description: This command sets the acceleration profile which can be an integer value between 1 and 255. The lower the value the faster the motor acceleration, so a 1 is the fastest profile and 255 is the slowest.

Range: 1 - 255

### B - Base Speed

Format: B [value]

Description: This command sets the base (start) speed for motion. This value must be set before motion begins and be less then the maximum speed.

Range: 77 - 3500

### M - Max Speed

Format: M [value]

Description: This command sets the maximum (running) speed for motion. This value must be set before motion begins and be greater then the base speed.

Range: 77 - 15000

### J - Jog Speed

Format: J [value]

Description: This command sets the jog speed. This value must be greater than the base speed.

Range: 77 - 15000

### N - Number of Steps

Format: N [value]

Description: This command sets the number of clocks for the PCL501(PC) to send out following a G command.

Range: 0 - 8388607

### G - Go Number of Steps

Format: G

Description: This command is used to send a set number of clocks out of the PCL501(PC). An N or P command must be entered before the G command.

### P - Absolute Position

Format: P [value]

Description: This command calculates and sets the number of steps necessary to move to the specified position. (N=P-Z)

Range: -8388608 - 8388607

### S - Go Slew

Format: S

Description: This command will send clocks out to the PCL501(PC). The only commands that can stop the clocks are; . (stop motion) or LS (soft limit). Motion can also be stopped by using the limit switch inputs. The ramp profile is specified by the B (base speed), M (max speed), and A (acceleration/deceleration) commands.

### Z - Position

Format: Z [number]

Description: This command sets the current position as a reference. This register can contain a positive or negative value but cannot be changed while motion is in progress.

Range: -8388608 to +8388607

### H - Home

Format: H [binary]

Description: This command sends clocks out of the PCL501(PC) until the home limit or the soft limit is active. There are two types of homing available.

Home Types: H0: In type 0 homing, the PCL501(PC) will send clocks until a soft limit is reached, then ramp down to base speed. Clocks will continue until a home or hard limit is reached.

H1: In type 1 homing, the unit will move until a home limit is reached, then change directions, ramp down to base speed and stop upon release of the home limit input.

### . - Stop Motion

Format: .

Description: This command will stop all motion. It can also be used to stop the current program that is running.

### + - Clockwise

Format: +

Description: This command sets the direction output to clockwise.

### - - Counter-Clockwise

Format: -

Description: This command sets the direction output to counterclockwise.

**V - Verify**

Format: V [command]

Description: This command can be used with most commands to verify the register contents. This is a read only command. Valid Commands are: A, B, F, J, M, N, 0, P, Z, and +.

**O - Set Outputs**

Format: O [value]

Description: This command sets the outputs according to the binary value. Output 1 is the LSB and output 2 is the MSB.

Range: 0 - 3

**I - Read Inputs**

Format: I

Description: This command returns the binary value of the inputs to the PC. Since the inputs are pulled up internally, they will return a high when they are pulled low. For example, if all inputs are active (grounded), the command will return a 15. If all inputs are inactive (open), the command will return a 0. Input 1 is the LSB, input 2 is the second bit, input 3 is the third bit, and input 4 is the MSB.

**L0 - Get Limit Status**

Format: L0

Description: This command returns the binary value of the hard and soft in a binary format. The soft limit is the LSB and the hard limit is the MSB.

**LS - Soft Limit Input Bit**

Format: LS

Description: This command will ramp the clocks down to base speed. The move type then determines what will happen. In a relative or absolute type motion the PCL501(PC) will continue to the set position and stop. In a slew type motion the PCL501(PC) will ramp down and stop.

# Section 5: Troubleshooting

**Problem:**

*Can not establish communications with the PCL501(PC).*

**Possible Solutions:**

1) Make sure the PCL501(PC) has power. Is the Green LED on?
2) Check the RS232/RS485 connections.
3) Check for loose cable connection either on the PCL501(PC) or COM Port.
4) Was the software installed successfully?
5) Go to **Setup | Communication Settings** and verify COM port and baud rate settings.
6) Physically Verify that the Axis Address matches with the PCL501(PC) unit selected address.
7) Go to **Setup | Axis** and verify address setting.
8) Click on **Connect** icon to communicate with the PCL501(PC).
9) If problems still exist, contact Anaheim Automation Tech Support.

Anaheim Automation, Inc.
Tech Support:

4985 E Landon Drive
Anaheim, CA 92807
Phone: (714) 992-6990
Fax: (714) 992-0471
www.anaheimautomation.com

**Problem:**

*There is no power to the PCL501(PC).*

**Possible Solutions:**

1) Is the PCL501(PC) connected to the appropriate power supply?
2) Check for any blown fuses in line with the PCL501(PC).
3) If problems still exist, contact Anaheim Automation Tech Support.

**Problem:**

*My program won’t “Autostart”.*

**Possible Solutions:**

1) Verify that the Autostart Function has been enabled.
2) Go to **Setup | Autostart Program** and Click on Enable.
3) If problems still exist, contact Anaheim Automation Tech Support.

**Problem:**

*The PCL501(PC) has a fault condition.*

**Possible Solutions:**

1) Verify your program for improper syntax that may cause an error code.
2) Physically press the reset button on the PCL501(PC) to clear an error.
3) Another way to clear an error is by using either the SMC50WIN software or the direct mode command instructions set.
4) The SMC50WIN can clear an error in the motion tab section by clicking on the verify parameters button.
5) The direct mode commands can clear an error by simply prompting the error codes register.

Example: @0! (carriage return)

Description: Select the unit address by typing @ followed by the address number and ! (Error Codes Register) and a carriage return.

Note: Read the Error returned to the screen to better understand what can be causing the fault condition. The error is returned in binary coded decimal format. If two errors were received their binary values would be added together.

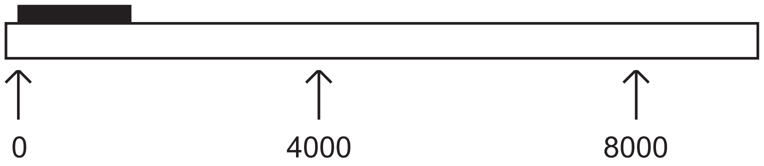
## Error Codes

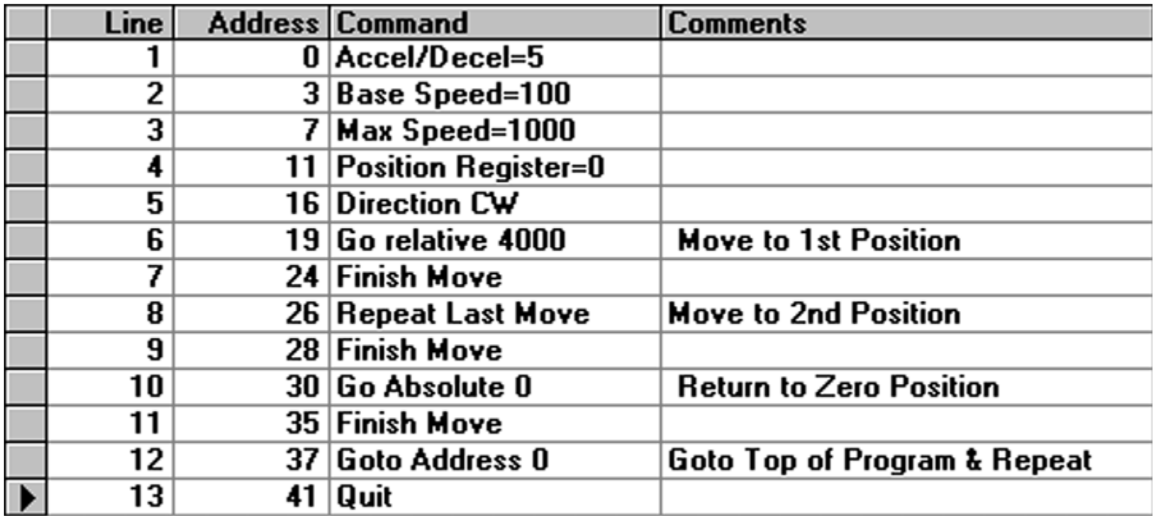
| Error Code | Type | Description |
|---|---|---|
| 1 | Recieve Overflow Error | The serial communications had a recieving error. This is an internal error caused by the computer. |
| 4 | Command Error | A bad command was sent to the controller. Please check to see that the command being sent is valid. |
| 16 | Motor Error | Motor speed profiles are set incorrectly. Please make sure that the base speed is less than the max speed and that the speeds are within their valid ranges. |
| 32 | Range Overflow Error | The go to position has an overflow error. This is caused by the P command trying to find a position that is out of its range. |
| 64 | Range Error | There was an invalid number of commands and characters sent to the controller. Check to see if the parameters are invalid for the command that was sent. |
| 128 | Transmitt Error | To many parameters sent back to the PC. This is an internal error caused by the eeprom. |
| 256 | Mode Error | Controller is in wrong mode. Some commands are good only in programming mode, while others are good only in direct mode. Check the direct mode section to see which commands are good in direct mode. |
| 512 | Zero Parameters Error | There were no parameters sent to the controller. A command was sent to the controller that expected to see parameters after the command. |
| 2048 | Memory Range Error | The specified address is out of range. This is caused by overflowing the program memory by having a program that is to large. |
| 4096 | Memory Command Error | The command pulled from memory is invalid. The command that was stored into the eeprom was non executable by the program. This is an internal error. |
| 8192 | Busy Error | The controller is busy indexing. The controller is sending out clocks to the driver and can not execute the next instruction. |

# Section 6: Tutorial

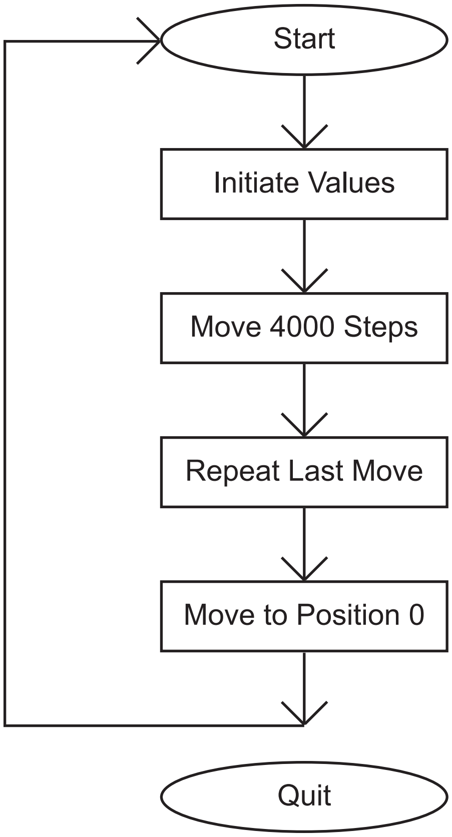
## Sample Program 1:

Sample porgram 1 illustrates a typical application where a system moves to a specific position required. The sample program shows how to use the motion and goto instruction commands.



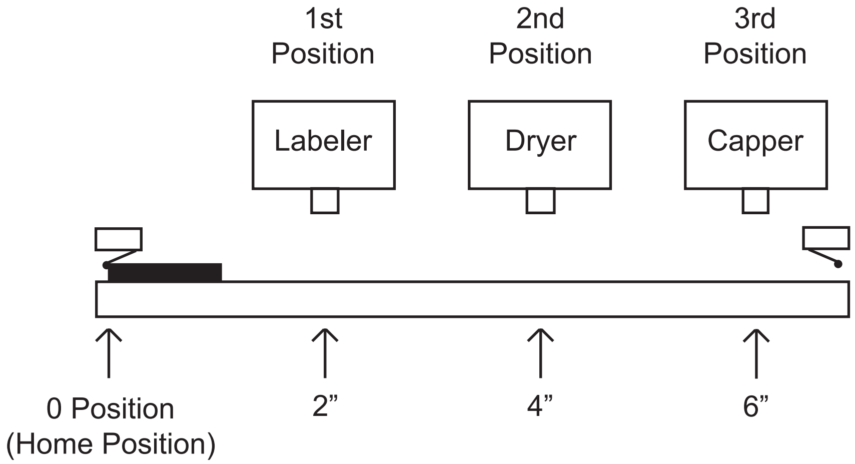


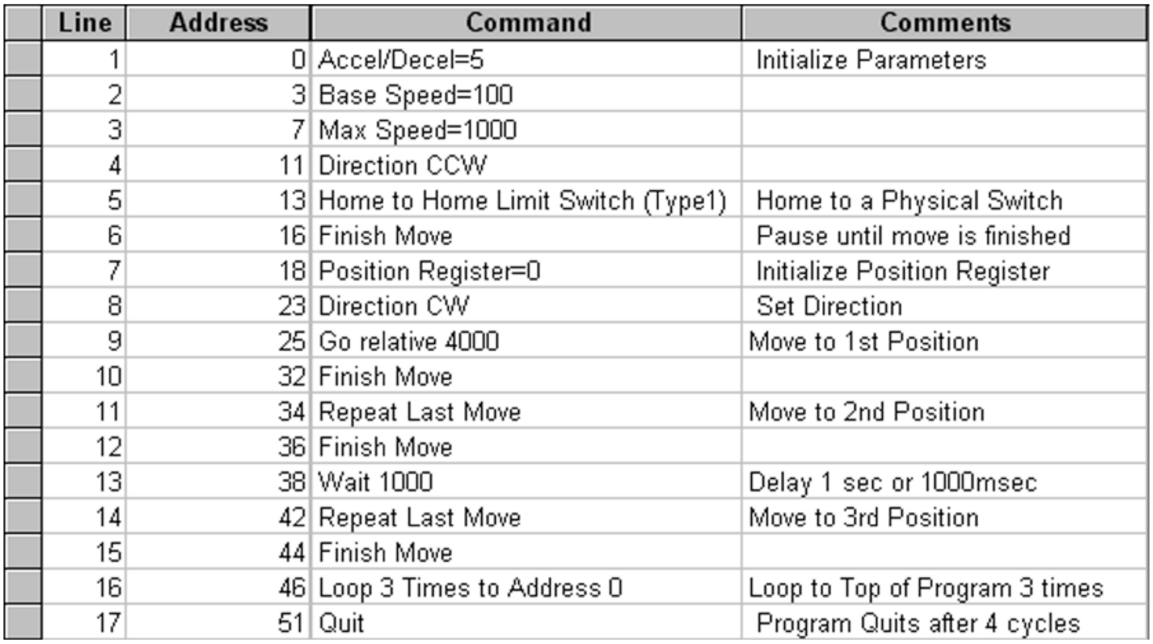
| | Line | Address | Command | Comments |
|---|---|---|---|---|
| | 1 | 0 | Accel/Decel=5 | |
| | 2 | 3 | Base Speed=100 | |
| | 3 | 7 | Max Speed=1000 | |
| | 4 | 11 | Position Register=0 | |
| | 5 | 16 | Direction CW | |
| | 6 | 19 | Go relative 4000 | Move to 1st Position |
| | 7 | 24 | Finish Move | |
| | 8 | 26 | Repeat Last Move | Move to 2nd Position |
| | 9 | 28 | Finish Move | |
| | 10 | 30 | Go Absolute 0 | Return to Zero Position |
| | 11 | 35 | Finish Move | |
| | 12 | 37 | Goto Address 0 | Goto Top of Program & Repeat |
| ▶ | 13 | 41 | Quit | |



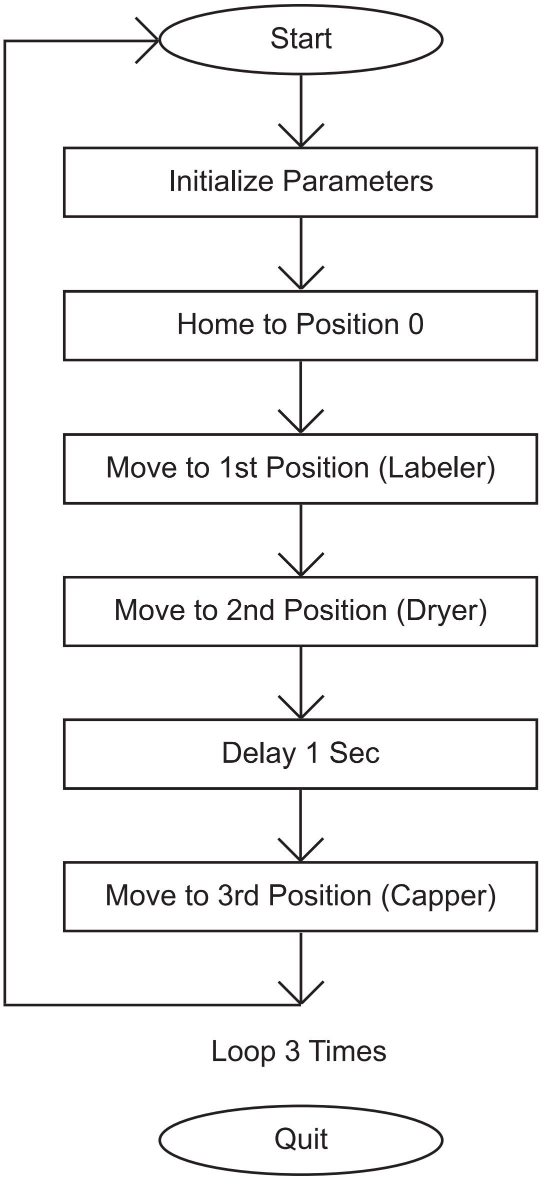
## Sample Program 2:

Sample program 2 illustrates a typical application where a system is first sent home to a datum or 0 position. This sample program shows how a motor will move to 3 different positions utilizing some of the motion commands and loop routine.





| Line | Address | Command | Comments |
|---|---|---|---|
| 1 | 0 | Accel/Decel=5 | Initialize Parameters |
| 2 | 3 | Base Speed=100 | |
| 3 | 7 | Max Speed=1000 | |
| 4 | 11 | Direction CCW | |
| 5 | 13 | Home to Home Limit Switch (Type1) | Home to a Physical Switch |
| 6 | 16 | Finish Move | Pause until move is finished |
| 7 | 18 | Position Register=0 | Initialize Position Register |
| 8 | 23 | Direction CW | Set Direction |
| 9 | 25 | Go relative 4000 | Move to 1st Position |
| 10 | 32 | Finish Move | |
| 11 | 34 | Repeat Last Move | Move to 2nd Position |
| 12 | 36 | Finish Move | |
| 13 | 38 | Wait 1000 | Delay 1 sec or 1000msec |
| 14 | 42 | Repeat Last Move | Move to 3rd Position |
| 15 | 44 | Finish Move | |
| 16 | 46 | Loop 3 Times to Address 0 | Loop to Top of Program 3 times |
| 17 | 51 | Quit | Program Quits after 4 cycles |


Start
Initialize Parameters
Home to Position 0
Move to 1st Position (Labeler)
Move to 2nd Position (Dryer)
Delay 1 Sec
Move to 3rd Position (Capper)
Loop 3 Times
Quit

# Appendix 1: ASCII Table for Direct Mode

| ASCII Symbol | Hex Value | ASCII Symbol | Hex Value |
|---|---|---|---|
| Carriage Return | 0D | I | 49 |
| 0 | 30 | J | 4A |
| 1 | 31 | L0 | 4C 30 |
| 2 | 32 | LS | 4C 53 |
| 3 | 33 | M | 4D |
| 4 | 34 | N | 4E |
| 5 | 35 | O | 4F |
| 6 | 36 | P | 50 |
| 7 | 37 | S | 53 |
| 8 | 38 | V | 56 |
| 9 | 39 | Z | 5A |
| A | 41 | ! | 21 |
| B | 42 | $ | 24 |
| F | 46 | + | 2B |
| G | 47 | - | 2D |
| H | 48 | . | 2E |

### Copyright

### Disclaimer

Though every effort has been made to supply complete and accurate information in this manual, the contents are subject to change without notice or obligation to inform the buyer. In no event will Anaheim Automation be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation.

### Limited Warranty

All Anaheim Automation products are warranted against defects in workmanship, materials and construction, when used under Normal Operating Conditions and when used in accordance with specifications. This warranty shall be in effect for a period of twelve months from the date of purchase or eighteen months from the date of manufacture, whichever comes first. Warranty provisions may be voided if the products are subjected to physical damage or abuse.

Anaheim Automation will repair or replace at its option, any of its products which have been found to be defective and are within the warranty period, provided that the item is shipped freight prepaid, with RMA (return material authorization), to Anaheim Automation's plant in Anaheim, California.

### Trademarks

Control Link and Driver Pack are registered trademarks of Anaheim Automation.

IBM PC is a registered trademark of International Business Machines, Inc.

ANAHEIM AUTOMATION