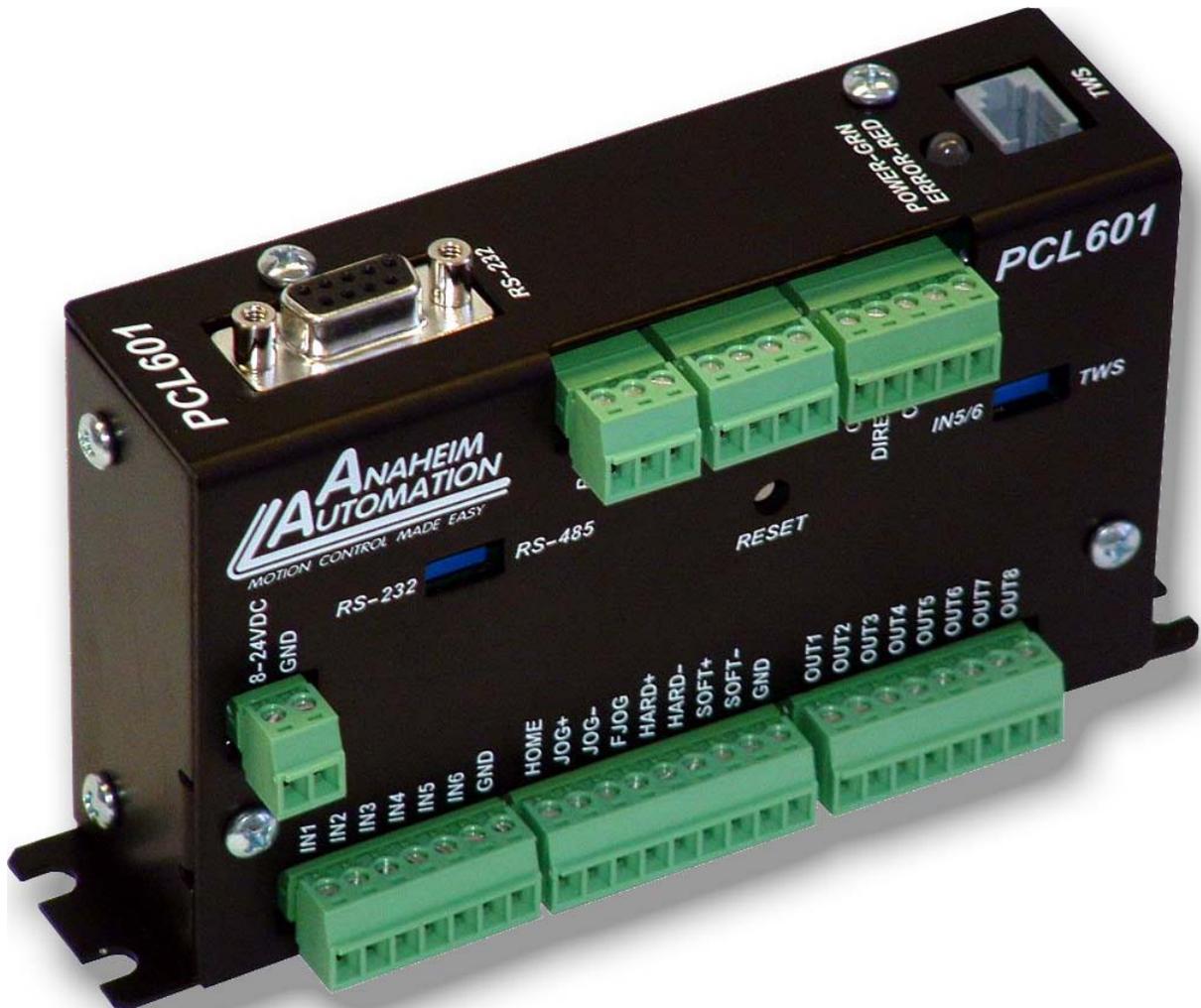


PCL601 Programmable Step Motor Controller

User's Guide



ANAHEIM AUTOMATION, INC.

910 East Orangefair Lane, Anaheim, CA 92801
e-mail: info@anaheimautomation.com

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

Table of Contents

| | |
|---|-----------|
| Section 1: Introduction | 3 |
| Description | 3 |
| Methods of Communication | 4 |
| Baud Rate | 4 |
| RS232 Protocol - SW1 in RS232 position | 4 |
| RS485 Protocol - SW1 in RS485 position | 4 |
| Terminating Resistor | 5 |
| Axis Selection | 5 |
| Status LED | 5 |
| Interfacing With A Driver | 5 |
| Technical Support | 5 |
| Electrical Specifications | 6 |
| Ordering Information | 6 |
| Dimensions/Switch Locations | 7 |
| Wiring Diagrams..... | 7 |
| Terminal Descriptions | 8 |
| Connector Descriptions | 8 |
| Slide Switch Descriptions | 8 |
| Section 2: Functions | 9 |
| Section 3: SMC60WIN Software | 13 |
| File Menu | 14 |
| Setup Menu | 14 |
| Setup - Axis Menu | 14 |
| Program Menu | 15 |
| Program - Autostart Program Menu | 15 |
| Edit Menu | 15 |
| Help Menu | 16 |
| “The Unit is Connected” / “The Unit is NOT Connected” | 16 |
| Toolbar | 17 |
| Tab Sheets | 17 |
| Add/Change/Insert Commands | 22 |
| Calculator | 27 |
| Section 4: Direct Talk Mode | 28 |
| Section 5: Troubleshooting | 38 |
| Error Codes | 39 |
| Section 6: Tutorial | 40 |
| Appendix 1: ASCII Table for Direct Mode | 45 |
| Appendix 2: Firmware Revisions | 45 |

Section 1: Introduction

The PCL601 is a single-axis step motor controller containing 2 Kbytes of nonvolatile stored programming space and quadrature encoder feedback. The user can write the program and then have the PCL601 autostart the program on power up. It provides flexible, independent control of stepper motors from computers, or any machine controller with a serial port. It is also capable of standalone operation, making it an embedded machine controller. The easy to use Windows software, SMC60WIN, can be used to directly control motion and to program the PCL601. The PCL601 also has the ability for real time functions. "Direct Mode" is used to directly control motion for Real Time Movements through serial communication.

The PCL601 has 40 commands, which are easy-to-remember for direct movement of a stepper motor. The PCL601 communicates via either an RS232 or RS485 bidirectional serial data bus. Up to 99 PCL601's can be networked from one communications port on your PC or PLC, utilizing the RS485 communications protocol. Special functions of the PCL601 include 8 programmable open collector outputs and 6 TTL, CMOS and 24V compatible inputs, a quadrature encoder input with the ability to autocorrect, an analog input to control either maximum speed or absolute position, registration mark indexing during a slew command, an output that will trigger during an index command at an absolute position, and a thumbwheel input for indexing a motor. The PCL601 can be powered with DC (5VDC or 8-24VDC) or AC (8-16VAC) voltages and uses only 2 watts at 24VDC and only a 1/2 watt at 5VDC (with no connections to terminal blocks).

Description

The PCL601 step motor controller provides independent programming of acceleration/deceleration, base speed (start up speed), max speed (running speed), jog speed, and the number of steps to be taken in both relative and absolute positioning modes. On absolute positioning moves, the PCL601 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 PCL601 also has specific functions such as *index-on-the-fly*, which during a slew move will move a predefined number of steps after an input has been triggered. *Output-on-the-fly*, which will trigger an output on for 50uS during an indexing move at an absolute position and repeat triggering the output on after a given number of steps. An analog input can be used to set either the maximum speed or goto an absolute position based between the upper and lower programmable limits. A seven decade thumbwheel switch can be read for relative indexing. The PCL601 also has a high level programming command set that includes: branching, looping, conditional statements, time delays, text strings, and I/O which the user can use in the programming mode to fully control all machine functionality.

A home input, a set of bidirectional hard and soft limit switch inputs and bidirectional jog inputs are provided for each axis. These features are generally required in most machine control designs. 6 testable TTL, CMOS and 24V compatible inputs and 8 programmable open-collector outputs are provided per axis. The I/O may be used for monitoring and controlling machine operation and/or interaxis coordination. The I/O are accessible independent of the busy state of the axis controls.

The PCL601 has a built-in programmable reset circuit. Reset is automatic on power-up, or by pressing the external reset button. A CD, provided when you purchase the unit, contains this user's manual, along with the SMC60WIN software and PCL601 program examples. The software allows you to write and change programs that are to be stored in the PCL601 for autostart use, and also upload the program that is stored in the PCL601 itself for editing and viewing. The software also allows you to save the programs onto your computer hard drive, and easily retrieve them when needed.

Methods of Communication

There are two methods for sending commands to the PCL601. One is to directly talk to the PCL601 by using Direct Talk Mode. This is usually used with a computer or PLC (Programmable Logic Controller), where the computer or PLC gives the PCL601 serial commands to off-load its processor. For example: A PLC can utilize its outputs to toggle the PCL601's inputs and gain control of variable speeds, variable programs, variable distances, etc. Simply using the PCL601 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 PCL601 is to use the software program SMC60WIN to either manually control, or to write and send programs. This method is used when the PCL601 is the main controller. For example: A PCL601 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 6 or less inputs and 8 or less outputs can utilize a PCL601 controller.

Baud Rate

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 PCL601 accepts a baud rate of 38400 only .**

RS232 Protocol - SW1 in RS232 position

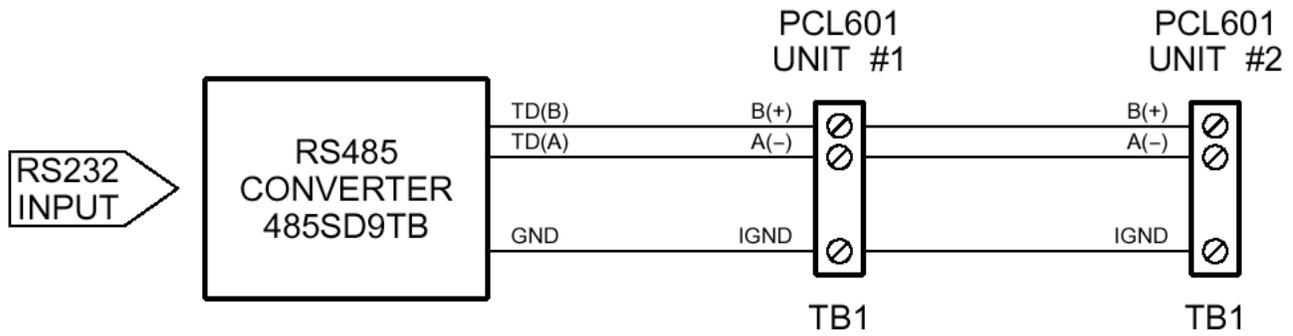
The PCL601 is a DCE device, therefore it will transmit on pin 2 and receive on pin3 of the DB9 RS-232 connector . The RS232 serial communication mode is single ended. This means that for each signal there is one wire, and a common ground reference used by all the signals. The PCL601 does not use handshaking, thus the CTS and RTS lines are internally connected, and the CD, DTR and DSR lines are internally connected inside the PCL601. The signal line maintains levels of +5VDC to +15VDC and -5VDC to -15VDC. For a valid logic level in the controller, the voltage must be at least +/-3 volts. RS232 works at distances of up to 50 feet maximum. **RS232 is susceptible to electrical noise, and should not be used in noisy areas. Always use the shortest cable connection possible. NOTE: Keep controller wiring separated from motor cable/wiring.**

RS485 Protocol - SW1 in RS485 position

The RS485 protocol mode 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 and then turn it off after it has completed transmission. Note: The above protocol is done internally between the converter and the PCL601. 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. The PCL601 is designed for two wire configuration. The 2 wire configuration makes use of the tristate capabilities of RS485 to 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 must be used) reduces cabling cost. **NOTE: Keep control wiring separated from motor cable/wiring.**

RS232 to RS485 for multiple units or cables longer than 50ft

The PCL601 can be connected to your PC serial port via a RS485 converter (*model number: 485SD9TB sold separately*). This converter will convert the RS232 voltage signals to the compatible RS485 differential signals. Only one converter box is needed per serial port. Contact the factory or use the website www.anaheimautomation.com for RS485 converter information and sales.



Terminating Resistor

To eliminate noise on the transmission lines or when using a 4000 ft. or longer cable, a terminating resistor is suggested. If used, the termination resistor need only be added to the last (furthest from the converter box) PCL601 in the network between pins A(-) and B(+) on the RS485 Terminal Block. The value of this resistor should be 120 ohms.

Axis Selection

Each PCL601 is addressed using a programmable register allowing the PC to address up to 99 PCL601's from one port. The Default axis is "0". To change the axis, use the SMC60WIN software or the "~" command. To verify or check the axis, use the SMC60WIN software or the "%" command. The axis designation is nonvolatile and will remain the same until changed by the user.

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 error code "!" command. Once the error has been read and cleared, the LED will return to green and the error code register will be cleared to 0. Refer to the table on page 39 for a complete list of the error codes.

Interfacing With A Driver

The PCL601 controller was designed to control a step motor driver. For drivers with "Opto-isolated" inputs, the PCL601 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, direction and on/off outputs of the PCL601 to the negative terminals of the corresponding inputs on the driver. For a driver with "TTL/CMOS" inputs, the PCL601 has a negative going clock that will sink the current from the driver's inputs. Wire the clock, direction and on/off outputs and 0VDC reference of the PCL601 to the corresponding inputs of the driver respectively. For a PCL601 controller purchased with an integrated driver and power supply (Driver Pack), these connections are already internally made.

Technical Support

Everyone needs assistance 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 your questions. 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 may call the factory direct for application assistance. If possible, have this manual in hand. It is often helpful to have the controller connected to a computer with the software installed.

Electrical Specifications

Power Requirements With No Connections:

5VDC @ 100mA or
8VDC to 24VDC @ 90mA

Operating Temperature:

0 to 60 degrees C

Pulse Output Range:

1 to 50,000 Hz
10uS negative going pulse width

Inputs (TTL-CMOS):

Logic "0": 0 to 0.8VDC
Logic "1": 3.5 to 24VDC
Analog input 1: 0 to 5VDC

Outputs (CLK,DIR,ON/OFF):

Open Drain Type
40VDC, 75mA

Baud Rate:

38400 Baud, Fixed

Data Format:

Half-Duplex, 1 start bit, 8 data bits,
no parity, 1 stop bit

Outputs (8 programmable):

Open Drain Type
40V, 100mA

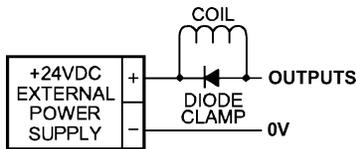
Output1 active low time for output on the fly:

50uS

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

Encoder and Driver Output:

+5VDC Output, 50mA

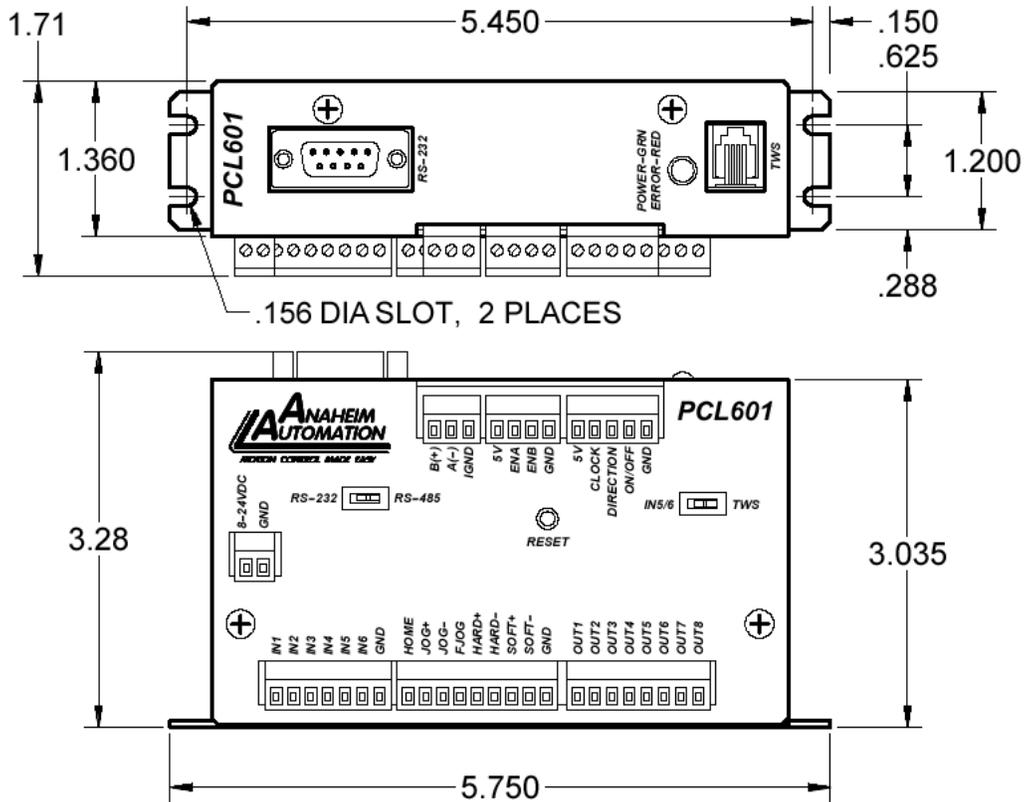


Ordering Information

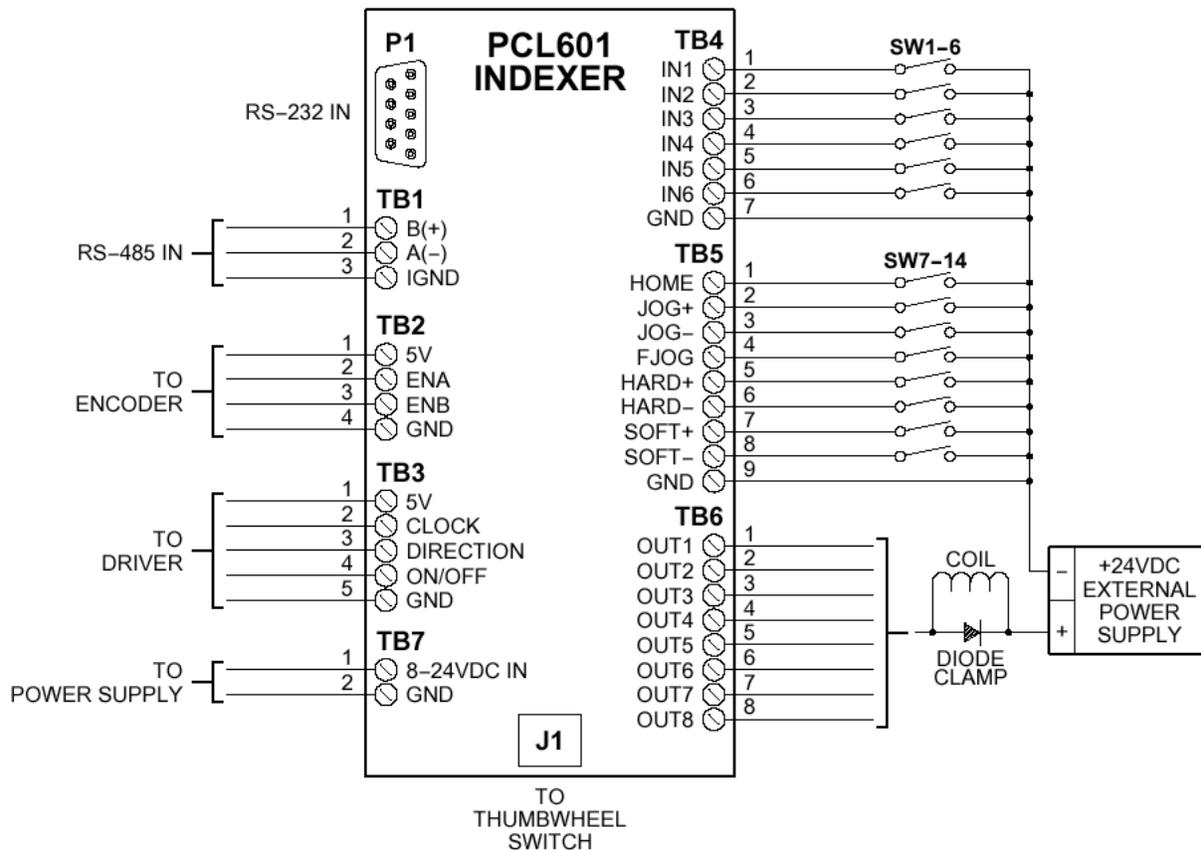
The table below lists a variety of products available from Anaheim Automation, Inc. These products include those covered by this manual, along with supporting cables and devices. We are continually adding new products to our line, so please consult Anaheim Automation, Inc. or its representatives for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

| Part Number | Description |
|--------------------|--|
| PCL601 | Featured step motor controller with encoder feedback. |
| DPD75601 | Controller/Drive Pack - Features a 7 amp unipolar drive and power supply. |
| DPN10601 | Controller/Drive Pack - Features a 10 amp bipolar drive and power supply. |
| 485SD9TB | RS232 to RS485 converter . |
| TWS7 | Seven position thumbwheel switch compatible with any SMC60 series controller. |
| AA9MFC-6 | 6 foot straight through serial cable with one DB9 male and one DB9 female connector. |
| PSAM24V1.2A-5V3.5A | Power supply for PCL601. (24V@1.2A, 5V@3.5A) |

Dimensions/Switch Locations



Wiring Diagrams



Terminal Descriptions

| Position | Description - Power |
|----------|---------------------|
| 1 | 8-24VDC Power Input |
| 2 | Ground Power Return |

| Position | Description - RS485 |
|----------|--|
| 1 | A(-) |
| 2 | B(+) |
| 3 | IGND - This is an isolated ground for RS485 only |

| Position | Description - Encoder |
|----------|---------------------------|
| 1 | +5VDC supply for encoder |
| 2 | A channel for encoder |
| 3 | B channel for encoder |
| 4 | Ground return for encoder |

| Position | Description - Limit Switch Inputs |
|----------|-----------------------------------|
| 1 | Home Limit |
| 2 | Jog + |
| 3 | Jog - |
| 4 | Fast Jog |
| 5 | Hard Limit + |
| 6 | Hard Limit - |
| 7 | Soft Limit + |
| 8 | Soft Limit - |
| 9 | Ground |

| Position | Description - Inputs |
|----------|----------------------------------|
| 1 | Input 1 - Analog input |
| 2 | Input 2 - Index on the fly input |
| 3 | Input 3 |
| 4 | Input 4 |
| 5 | Input 5 - SW2 in position IN5/6 |
| 6 | Input 6 - SW2 in position IN5/6 |
| 7 | Ground |

| Position | Description - Outputs |
|----------|---|
| 1 | Output 1 - Output on the fly output |
| 2 | Output 2 |
| 3 | Output 3 |
| 4 | Output 4 |
| 5 | Output 5 |
| 6 | Output 6 |
| 7 | Output 7 |
| 8 | Output 8 - Encoder retries error output |

| Position | Description - Driver Outputs |
|----------|--|
| 1 | +5VDC Supply for opto-isolated driver inputs |
| 2 | Clock output |
| 3 | Direction output |
| 4 | Motor current on/off output |
| 5 | Ground reference for non opto-isolated driver inputs |

Connector Descriptions

| Switch Number | Description |
|---------------|---|
| P1 | This connector is for the RS-232 communication and is labeled RS-232. |
| J1 | This connector is for the thumbwheel module and is labeled TWS. |

Slide Switch Descriptions

| Switch Number | Description |
|---------------|--|
| SW1 | This switch is used to select either RS232 or RS485. |
| SW2 | This switch is used to select either the thumbwheel or inputs 5 and 6. |

Section 2: Functions

Move Number of Steps: This command causes the motion to start in the direction last specified. This command will move the motor the number of steps given. (Range: 1 to 8388607)

Move to Position: The move to position command specifies the next absolute position to go to. The PCL601 controller automatically sets the direction and number of steps needed to go to that position. (Range: -8388607 to +8388607)

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

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: -8388607 to +8388607)

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). All limit switch inputs are internally clamped to +5V, thus allowing voltages of upto +24VDC to be used.

Hard Limit Inputs: When a hard limit switch is encountered, the motion will stop immediately. 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 positioning type functions.* These limits are directional.

Soft Limit Inputs: These switches should be used exclusively for homing. Once positioned properly with the appropriate parameters, it causes the motor to ramp down to the base speed before encountering the home limit switch. However, the soft limit switch will work for any type of motion command. These limits are directional.

NOTE: Whenever a soft limit switch is activated, the motor will decelerate and run at base speed during an indexing move, or stop during a slewing move. Be sure to come back past the soft limit switch to set any origins, otherwise the motor will decelerate as it goes past the soft limit switch during normal operation.

Home Limit Input: This switch is used to establish a position designated “home” or datum position using the following: home to *soft and home limit*, or home to *home limit*. This limit is not directional.

Home to Soft, Home Limit (2 Switch Operation): This type of homing routine requires two grounding type limit switches called home and soft. The first limit switch seen is the soft limit. This will decelerate the motor down to base speed. The motor will then continue to run at base speed until it contacts the home limit switch input causing the motor to stop. The home limit switch activates as a hard limit if a soft limit is not sensed. The soft limit is directional, meaning that it will work in only one direction as specified. The soft limit switch will work for any type of motion command. The home limit switch will work only for the two home motion commands.

NOTE: There should be sufficient distance between the two limit switches, as to let the motor reach base speed.

Home to Home Limit (1 Switch Operation): This type of homing differs in that only one limit switch is needed. In this homing routine the motor moves toward the home limit switch. When the home limit switch is contacted the motor will ramp down to base speed, reverse direction and continue at base speed until the limit switch is released. This is a good way to compensate for any backlash in a system. It is also useful for minimizing the number of limit switches needed for homing.

NOTE: The home switch needs to be low during the entire deceleration and reversing time.

Jog Inputs: The jog 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). All jog switch inputs are internally clamped to +5V, thus allowing voltages of upto +24VDC to be used. Jog is a manual function. The user can select the direction and speed (fast or slow) by grounding the appropriate combinations of inputs. To jog a motor, it is necessary to ground the jog input for the direction desired. For fast jog, both the fast input and jog input for the appropriate direction must be low at the same time. By grounding one of the jog inputs, the user causes the motor to run at base speed. When the fast input is grounded, the motor will then accelerate to the programmed jog speed. 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.

Inputs: All inputs (except input 1) are internally pulled up by a resistor making them normally +5 volts. To activate the input, the pin must be grounded to (0VDC). All inputs are internally clamped to +5V, thus allowing voltages of upto +24VDC to be used. Six inputs are provided per axis. The inputs are TTL, CMOS and 24V compatible. 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 wait for temperature to be reached, etc. A grounded input will read a "0" and an open or high input will read as a "1". Input 1 is a special input that is capable of reading an analog voltage between 0 and +5VDC. Since this input does not have a pull-up resistor, biasing of this input is needed if it is not used as an analog input. Inputs 5 and 6 are used together with the thumbwheel switch. To use inputs 5 and 6, SW2 must be in the IN5/6 position. If SW2 is in the TWS position, then these two inputs are not connected to the processor.

Analog Input: Input 1 can be configured to read an analog voltage to either set the absolute position of the motor or to set the maximum speed of the motor.

To set the position, when told via the *goto analog position* command, the input will read a voltage between 0 and +5VDC and based on the "upper and lower" limits of the function, a move will occur to a calculated position between the two limits. The motor must finish the move before it can be told to read the input again for the next position. For example, if the lower limit is set to 0 and the upper limit is set to 5000 and the analog position is set at +2.0VDC, then the motor will move to position 2000. Changing the lower limit to 1000 and the voltage to +3.2VDC, the motor will move to position 3560. See examples below for calculations of the analog inputs. (Range of limits: 0 to 65535 and the lower limit < upper limit)

To set the max speed, when told via the *set analog speed* command the input will read a voltage between 0 and +5VDC, and based on the "upper and lower" limits of the function, a max speed can be obtained based on a calculated frequency between the two points. The speed however can not be changed when the DPY50601 is busy (moving). See examples below for calculations of the analog inputs. (Range of limits: 0 to 50000 and the lower limit < upper limit)

Analog calculations.

(Upper-Lower) * (Voltage/5) = X
Lower + X = Position or Frequency

Example1:

(5000 - 0) * (2 / 5) = 2000
0 + 2000 = 2000

Example2:

(5000 - 1000) * (3.2 / 5) = 2560
1000 + 2560 = 3560

Outputs: Eight outputs are provided per axis. Outputs may be used to operate relays, coolant valves, air cylinders, or, with the correct interfacing, any electronically controlled device. 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) from the output to the power source in order to provide adequate fly-back protection. The outputs are current sinking, open collector darlington. They are capable of sinking up to 100mA per output with voltages up to 40VDC. Turning an output on will pull the output pin to ground and turning an output off will make the output pin open. Output 1 has a special function (*output on the fly*) that will enable it to be triggered at a certain absolute position during a move. Output 8 has a special function that will trigger when the encoder retries function fails.

Output on the fly: This special function enables output 1 to turn on during a relative index or absolute move. There are three critical portions of information needed to make this function work correctly. First, output 1 will turn on (0VDC) for a preset delay of 50uS at a specific absolute position set by the *1st output position* command. Second, the output can then repeat this after a preset amount of steps set by the *number of steps between outputs* command, and third a predetermined amount of times to set the output is required by the *number of outputs* command which determines the preset amount of times to trigger the output. So if you start at position 0 and want to move to an absolute position of 10,000, you can set output 1 to turn on at position 2000, and every 1000 steps after that 5 times. So at position 2000, 3000, 4000, 5000, and 6000 output 1 will turn on for 50uS. To only have the output turn on at one position set both the “*number of steps between outputs*” and the “*number of outputs*” commands to 0. This function must be enabled, and will only work during a relative index or absolute position move. The output will trigger while going in either direction. If you do not want the output to trigger in the negative direction, the function must be turned off before the index move is started.

Index on the Fly: This special function uses Input 2 when a motor is slewing to move a predetermined amount of steps, set with the registration index command, before stopping. This function must be enabled, and will only work during a slew move. The registration index must be set before movement begins. (Range: 1 to 8388607)

End of Program: The end of program 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 65535)

If/Then Statements: The if/then statements are conditional based on the values preset in the program. The user can either test each individual input or all inputs at once. If the input or input register matches the given value or values, then the program will execute the next line. If the input or input register does not match the given value, the program will skip the next line and execute the following line. An open input is read as a 1, and a grounded input is read as a 0.

Branching or Goto Statements: The goto instruction will have the program jump to the given label. If no label is in the program, it will error when trying to send.

Return from Subroutine: This function can be placed anywhere in the program as long as a goto statement has been already executed. The program will jump back to the last goto statement encountered and execute the next line in the program.

Inner and Outer Loop: The loop instructions allow the user to loop a program a variable number of times. The program will loop to the designated label location of the program. However, the label must always be at a lower line number than the loop instruction itself. You can only nest inner loops inside an outer loop. You may not nest an inner loop inside an inner loop, or an outer loop inside an outer loop. Multiple nested inner loops are allowed in one outer loop.

Finish Move: When writing a program, the finish move command is used directly after every motion command. When using this command, the PCL601 internally generates a busy signal and will wait until the move is complete before executing any further commands. Unless the finish move command is used, the PCL601 will continue to execute the program. If it encounters a command that cannot be used when the motor is moving, the PCL601 will error and stop the program prematurely.

Repeat Last Move: This command will move the motor the number of steps given in the last indexing move. This command will not work correctly if the encoder autocorrect function is enabled.

Encoder Commands: The PCL601 controller is capable of using a quadrature incremental encoder with A and B channels.

Encoder Auto Correct: This command will enable or disable the encoder feature of the PCL601. When enabled, the encoder function will compare the desired position with the actual encoder position. If it is not in the correct position a correction move will be made.

Encoder Delay: This sets the wait time, which is a specified number of milliseconds after a relative index or absolute move is finished, prior to reading the encoder. This is used to remove the ringing that might be associated with the mechanics of the system. (Range: 0 to 65535)

Encoder Motor Ratio: This represents the ratio for the number of encoder pulses to one motor step. This ratio must be a whole number. For example, given a 1000 line quadrature encoder and a 400 step/revolution motor, the motor ratio is $(1000 * 4) / 400 = 10$ (Range: 1 to 255 and must be a whole number)

Encoder Retries: This is the number of times the PCL601 will try to autocorrect the motor shaft position before producing an error. When the error is produced, Output 8 is triggered. (Range: 0 to 255)

Encoder Window: This is the allowable error in encoder pulses (either plus or minus) from the desired position that is allowed before the motor autocorrects. (Range: 0 to 255)

Thumbwheel Index: This special function allows a thumbwheel with up to 7 decades to be used with the PCL601 to set a relative index. To use the thumbwheel, SW2 must be in the TWS position or the thumbwheel will be disabled.

Acceleration/Deceleration: The acceleration and deceleration are the same value. The acceleration is entered directly as steps/sec² and controls the time that the motor will take to move from base speed to max speed, and from max speed to base speed. The higher the value, the faster the motor will accelerate. The same principal applies for the deceleration which is controlling the time it takes to go from the max speed to base speed. (Range: 100 to 9,999,999)

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 and jog speed. (Range: 1 to 5000)

Max Speed: The max speed is the top speed the user wants the motor to run. This speed must always be greater than the base speed. It is entered directly as the number of steps per second. (Range: 1 to 50000)

Jog Speed: The jog speed sets the fast jog rate. Jog (+/-) is used to run at base speed. The FJOG pin, when grounded, will ramp the motor to the set jog speed. This speed must always be greater than the base speed. It is entered directly as the number of steps per second. (Range: 1 to 50000)

Motor Current: This command will control the on/off output which is designed to connect to the on/off input of Anaheim Automation's step motor drivers. To energize and allow current to flow through the coil of the motor, set the value to on. To de-energize and turn the current off to the motor, set the value to off. This is a dedicated output and not controlled with the output register.

Verify: The verify command causes the PCL601 controller 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 verify commands are shown in the table on page 34.

Section 3: SMC60WIN Software

The SMC60WIN software is a handy utility that supports Anaheim Automation's line of PCL601's step motor controllers. Connecting your PC to the PCL601, via a serial cable, the SMC60WIN software can easily perform the following tasks:

- Exercise and monitor the PCL601 controller
- Write and edit stored programs for standalone operation
- Directly communicate with the PCL601 controller

Installation

Software

- The SMC60WIN is supplied on a CD, containing the setup program and the SMC60WIN software, PCL601 manual and sample programs.
- SMC60WIN is compatible with all versions of Windows including Windows 2000 and Windows XP.

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**)

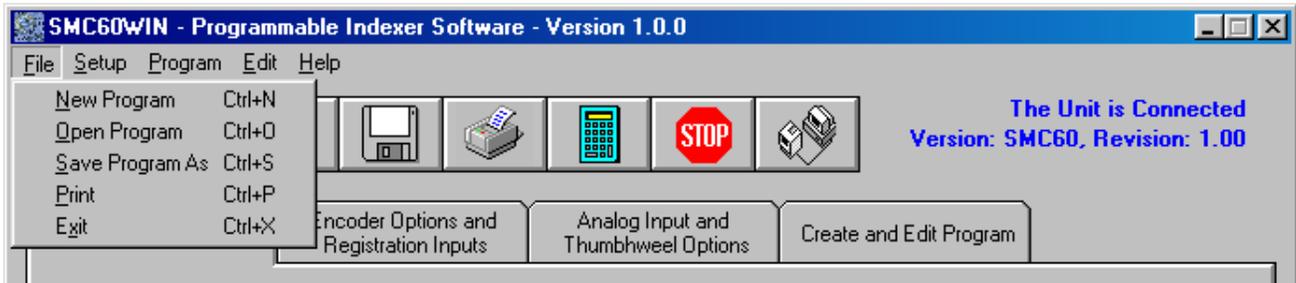
Option 2

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

Getting Started

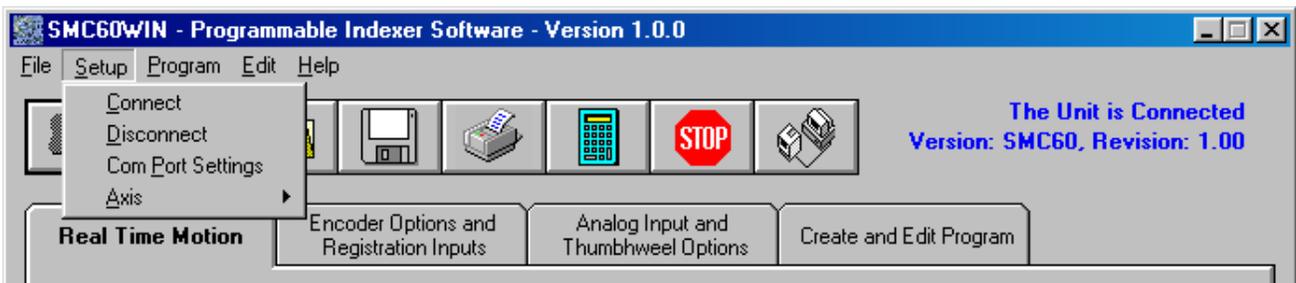
- 1) Double click on the SMC60WIN icon to run the SMC60WIN software.
- 2) Apply power to the PCL601 controller.
- 3) Set the appropriate com port setting by selecting Setup | Com Port Settings from the menu bar. (Ctrl+M is a shortcut)
- 4) Set the appropriate axis setting by selecting Setup | Axis from the menu bar. (Ctrl+A is a shortcut)
- 5) Establish communications with the PCL601 by clicking on the Connect Icon, or select Setup | Connect from the menu bar. If the unit is connected properly, the program will notify you when communications has been established. (Ctrl+C is a shortcut)

File Menu



| | |
|-----------------|-------------------------------------|
| New Program | Start editing a new program. |
| Open Program | Open an existing program from disk. |
| Save Program As | Save the current program to disk. |
| Print... | Print the current program. |
| Exit | Exit the SMC60WIN software. |

Setup Menu



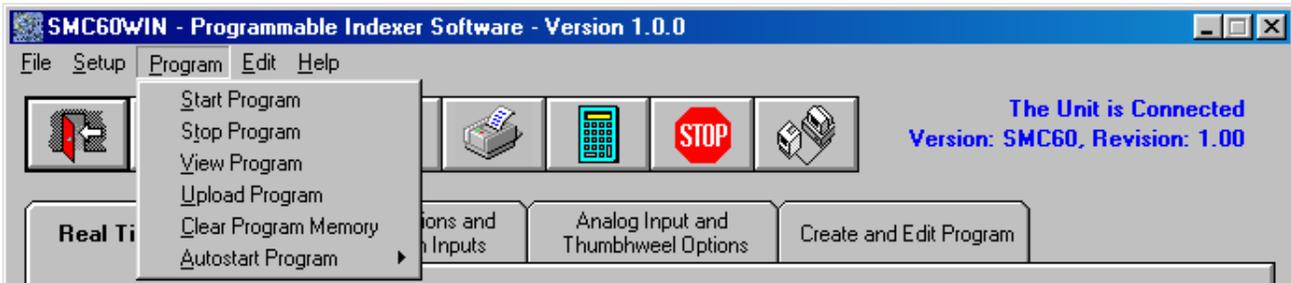
| | |
|-------------------|---|
| Connect | Establish communications with the controller. |
| Disconnect | Release the COM port for other devices to use. |
| Com Port Settings | Select COM port. |
| Axis | Set axis selection and stored axis of the controller. |

Setup - Axis Menu



| | |
|-------------|---|
| Select Axis | Sets the axis select parameter in the SMC60WIN software. (1-99) |
| Define Axis | Sets the programmable address in the controller. (1-99) |

Program Menu



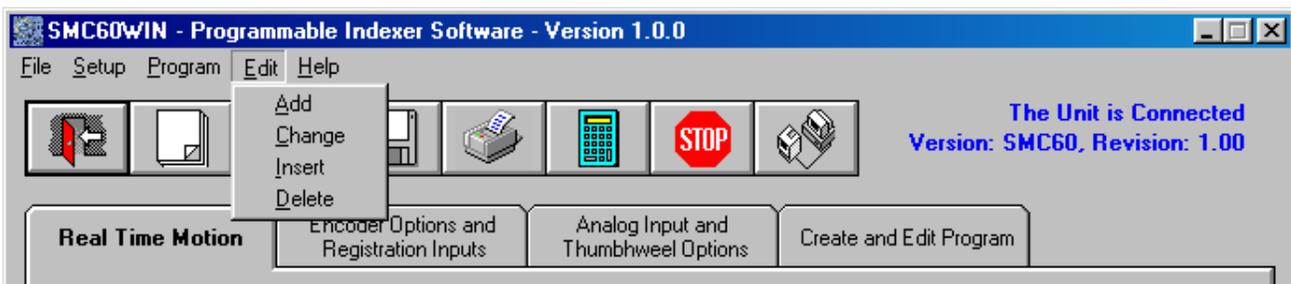
| | |
|----------------------|--|
| Start Program | Start the execution of the program in the controller memory. |
| Stop Program | Stop the execution of the program in the controller memory. |
| View Program | View the program stored in the controller memory. |
| Clear Program Memory | Clear the program memory in the controller. |
| Autostart Program | Turn the autostart function on or off. |

Program - Autostart Program Menu



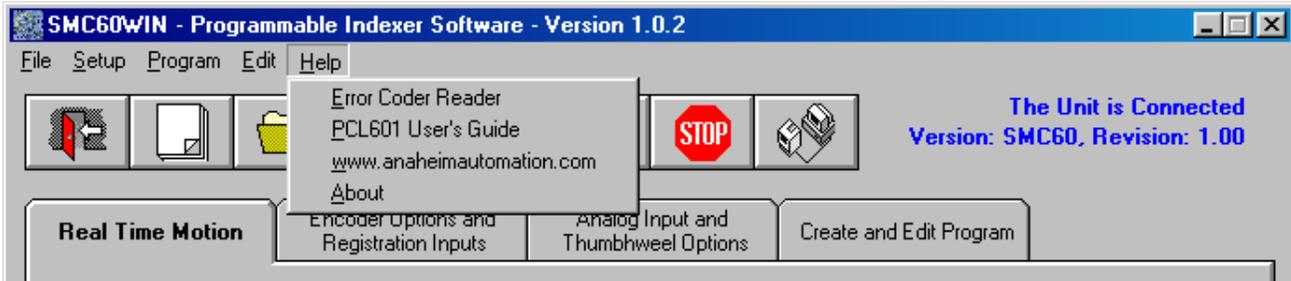
| | |
|---------|---|
| Disable | Program will disable the execution of a stored program at power up. |
| Enable | Program will start execution when controller is powered up. |

Edit Menu



| | |
|--------|---|
| 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. |

Help Menu



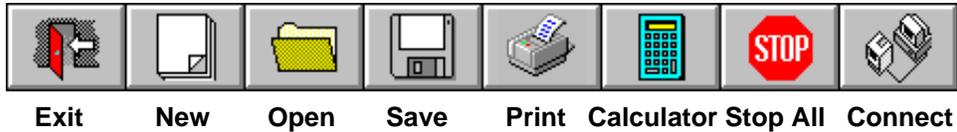
| | |
|---------------------------|--|
| Error Coder Reader | Utility to read the error code generated by the PCL601 controller. |
| PCL601 User's Guide | Opens up the User's Guide in .pdf format. |
| www.anaheimautomation.com | Opens up the Anaheim Automation Website. |
| About | Displays the version of the SMC60WIN and contact information. |

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

On the right of the Toolbar, the user will find the communication status of the PCL601 controller. If communications are not established, please refer to the Troubleshooting Section.

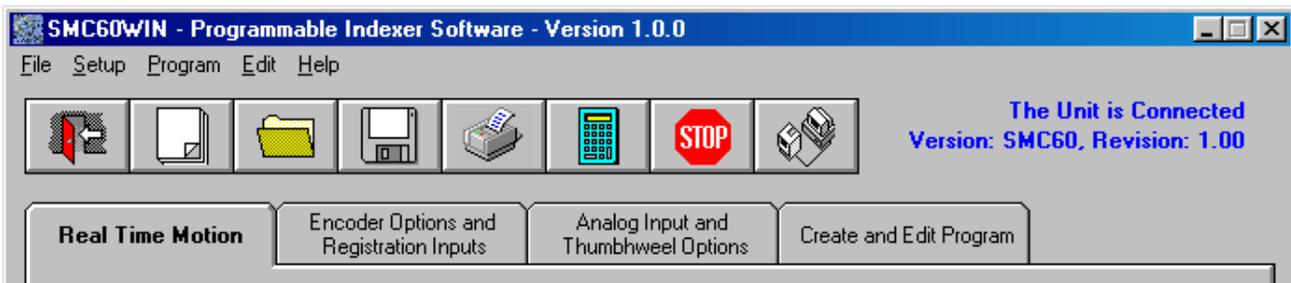


Toolbar



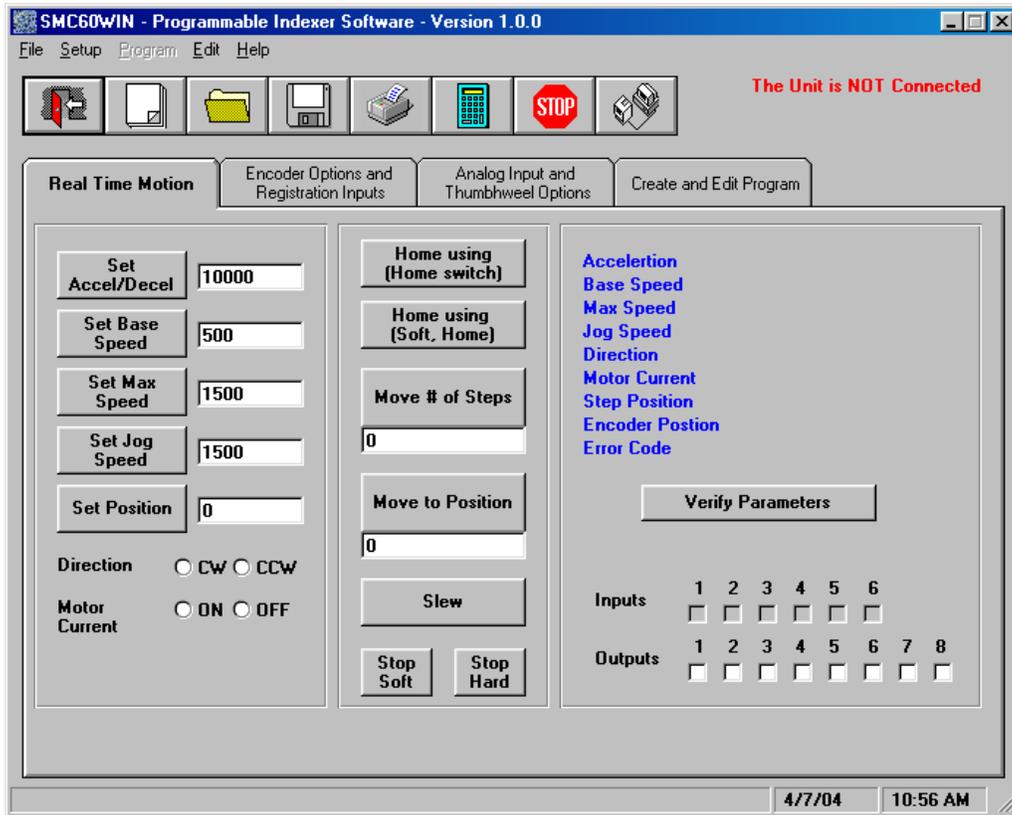
| | |
|------------|--|
| Exit | Exit the SMC60WIN software. |
| New | Start editing a new program. |
| Open | Open an existing program from disk or directory. |
| Save | Save the current program to disk or directory. |
| Print | Print the current program. |
| Calculator | Open the desktop calculator. |
| Stop All | Stop the program and all motion from running. |
| Connect | Establish communication with the controller. |

Tab Sheets



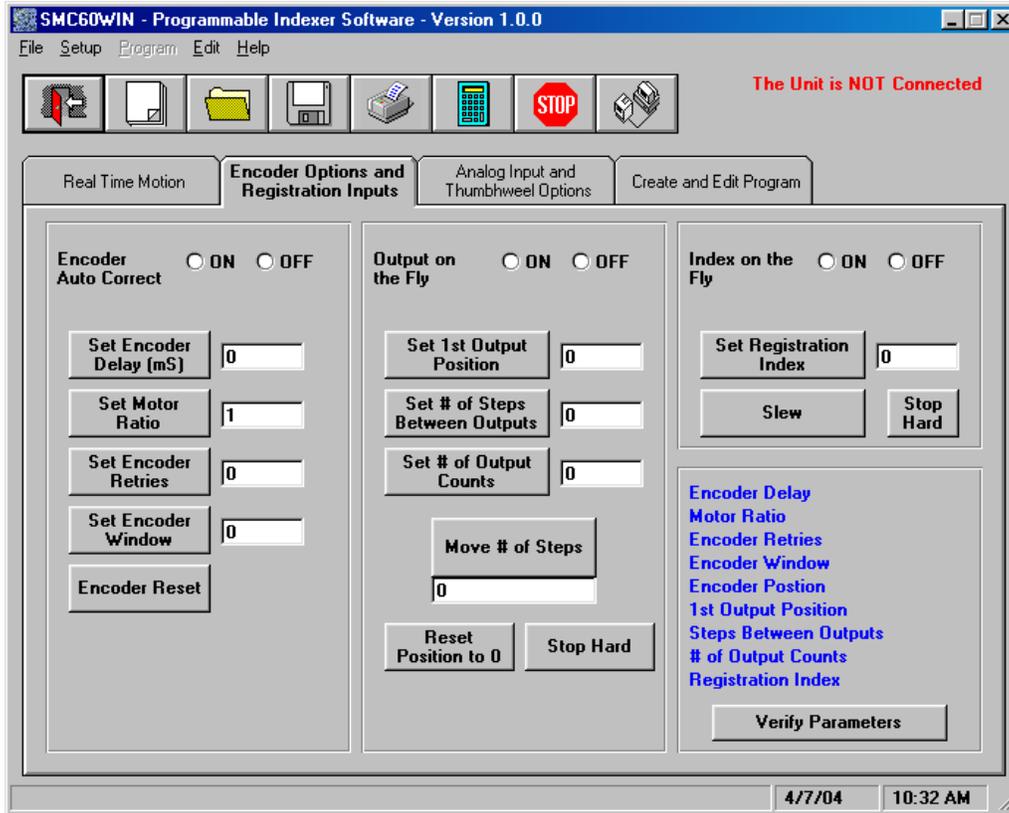
| | |
|---|--|
| Real Time Motion | Monitor and control motion of the controller. |
| Encoder Options and Registration Inputs | Monitor and change settings for encoder options, input on the fly and output on the fly. |
| Analog Input and Thumbwheel Options | Monitor and change settings to Analog Inputs and the thumbwheel switches. |
| Create and Edit Programs | Write and edit PCL601 stored programs. |

Tab Sheets - Real Time Motion



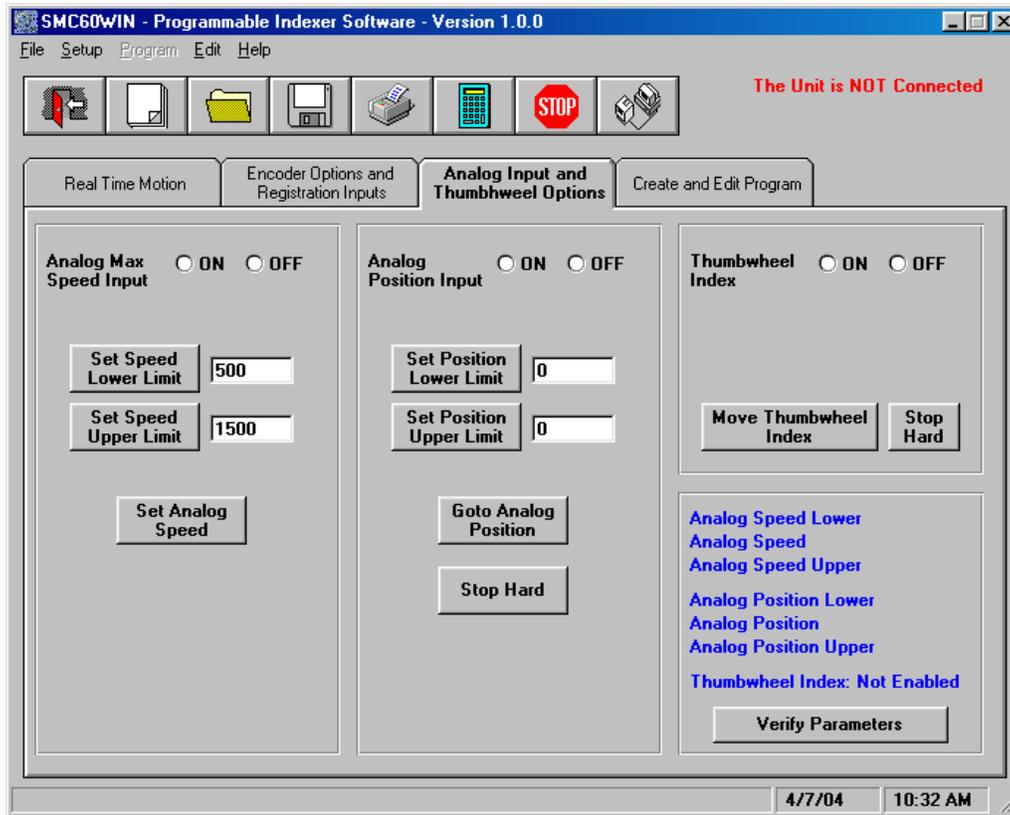
| | |
|-------------------------------------|--|
| Set Accel/Decel | Send acceleration & deceleration parameter to controller. (step/sec ²) |
| Set Base Speed | Send base speed parameter to the controller. (step/sec) |
| Set Max Speed | Send maximum speed parameter to the controller. (step/sec) |
| Set Jog Speed | Send fast jog speed parameter to the controller. (step/sec) |
| Set Position | Set motor position. |
| Set Direction | Set direction to clockwise or counter-clockwise. |
| Set Motor Current | Set the current in the motor on or off. |
| Home using (Home Switch) | Motor will seek the home position by moving towards home switch which 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 and Home Switches) | 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 | Motor will move number of steps entered. |
| Move to Position | Motor will move to specified position. |
| Slew | Motor will ramp up to maximum speed and keep moving until stop motion is triggered. |
| Stop Soft | Ramp motor down to base speed and stop. |
| Stop Hard | Stop any motor motion immediately. |
| Inputs | View inputs. (checked = On, blank = Off) |
| Outputs | View and trigger outputs. (checked = ON, blank = OFF) |
| Verify Parameters | Updates and displays controllers parameters sheet and resets the error codes. |

Tab Sheets - Encoder Options and Registration Inputs



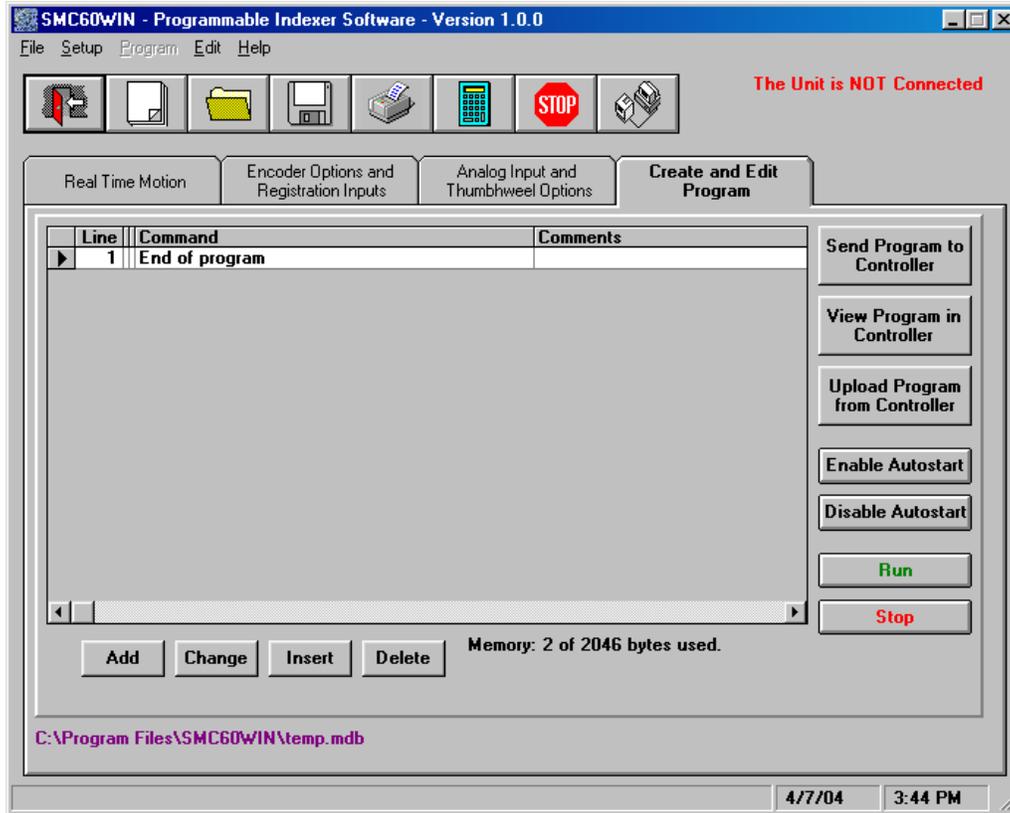
| | |
|--------------------------------|--|
| Encoder Auto Correct | Set the encoder autocorrect feature on or off. |
| Set Encoder Delay | Send the encoder delay parameter to the controller. (ms) |
| Set Motor Ratio | Send the encoder pulse to motor step ratio to the controller. |
| Set Encoder Retries | Send the number of encoder autocorrect retries to the controller. When the autocorrect errors, Output 8 will be triggered. |
| Set Encoder Window | Send the encoder window to the controller. |
| Encoder Reset | Reset the encoder count to 0 in the controller. |
| Output on the Fly | Set the output on the fly feature on or off. |
| Set 1st Output Position | Send the 1st position to set the output to the controller. |
| Set # of Steps Between Outputs | Send # of steps to take between activated outputs to the controller. |
| Set # of Output Counts | Send # of output counts to the controller. |
| Move # of Steps | Motor will move number of steps entered. |
| Reset Position to 0 | Reset the controller position to 0 (zero). |
| Stop Hard | Stop any motor motion immediately. |
| Index on the Fly | Set the index on the fly feature on or off. |
| Set Registration Index | Send registration index to controller. |
| Slew | Motor will ramp up to maximum speed and keep moving until the index on the fly input is activated or a stop motion is triggered. |
| Stop Hard | Stop any motor motion immediately. |
| Verify Parameters | Updates and displays controllers parameters for this tab sheet. |

Tab Sheets - Analog Input and Thumbwheel Options



| | |
|-----------------------------|--|
| Analog Max Speed Input | Sets the analog speed input feature on or off. |
| Set Speed Lower Limit | Send the analog speed lower limit to the controller. |
| Set Speed Upper Limit | Send the analog speed upper limit to the controller. |
| Set Analog Speed | Sets the max speed based on analog voltage measured at input. |
| Analog Position Speed Input | Sets the analog position input feature on or off. |
| Set Position Lower Limit | Send the analog position lower limit to the controller. |
| Set Position Upper Limit | Send the analog position upper limit to the controller. |
| Set Analog Position | Motor will move to position based on analog voltage measured at input. |
| Stop Hard | Stop any motor motion immediately. |
| Thumbwheel Index | Sets the thumbwheel index feature on or off. |
| Move Thumbwheel index | Motor will move number of steps set by the thumbwheel switches. |
| Stop Hard | Stop any motor motion immediately. |
| Verify Parameters | Updates and displays controllers parameters for this tab sheet. |

Tab Sheets - Create and Edit Program



| | |
|------------------------------|---|
| Send Program to Controller | Send current program to the controller. |
| View Program in Controller | View program in the controller memory. |
| Upload Program in Controller | Upload the program in the controller for editing and saving. |
| 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. |

PCL601 Memory Available

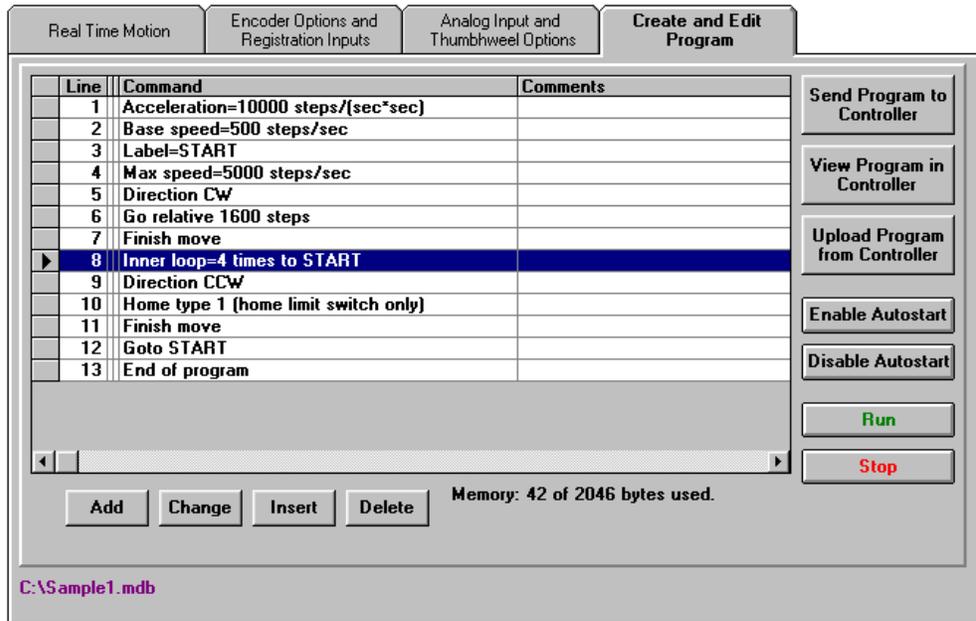
With the create and edit program tab sheet selected, the user can obtain the amount of available memory, located to the right of the Delete command button. The PCL601 has a maximum available memory of 2046 bytes - each instruction can use from 2 to 7 bytes.

Current Program Filename

With the create and edit program tab sheet selected, the user can obtain the current program filename, located in the lower left corner of the SMC60WIN window. All programs created by the SMC60WIN software will have a .mdb and a .bak extension.

Currently Selected Line

The currently selected line is indicated in the program by the right pointing arrow/triangle in the left column. Clicking on any line will select a new currently selected line.



Add/Change/Insert Commands

The Add/Change/Insert commands contain four different tab sheets, which are “Motion Commands”, “If/Then and Output Commands”, “Goto, For Loops, Encoder and Thumbwheel Commands” and “Analog, Registration and Text Commands”.

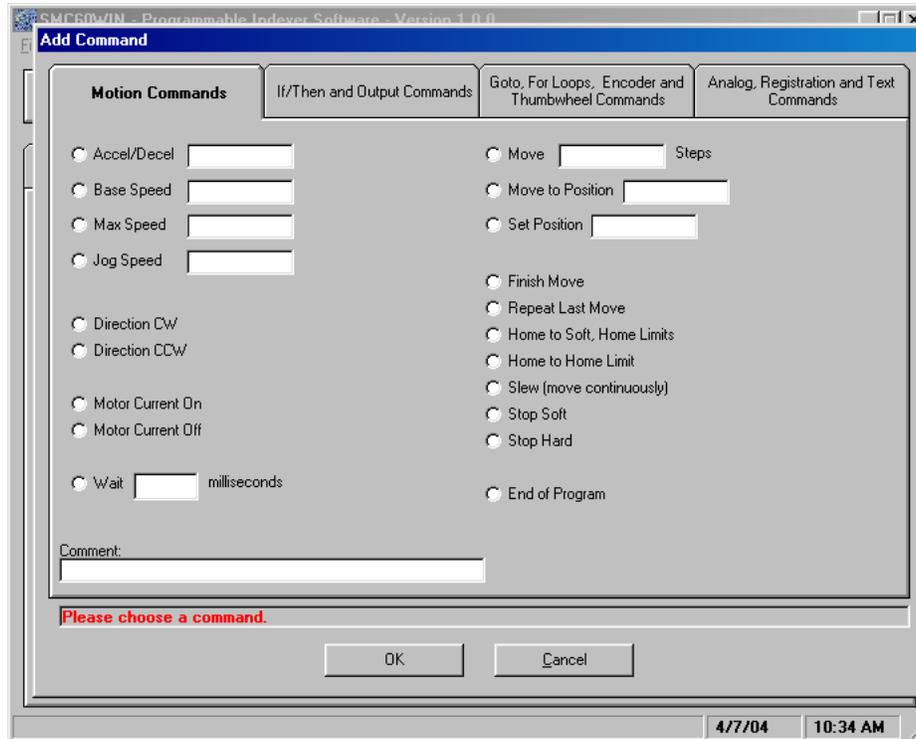


| | |
|---|---|
| Motion Commands | Software section that allows user to enter speeds, positions, direction, etc. |
| If/Then and Output Commands | Software section that allows user to manipulate conditional statements and I/O routines. |
| Goto, For Loop, Encoder and Thumbwheel Commands | Software section that allows user to manipulate branching and loop routines ,enter encoder parameters and control the thumbwheel switches. |
| Analog, Registration and Text Commands | Software section that allow user to enter parameters for analog speed and position limits, index on the fly and output on the fly parameters and text strings to be sent. |

These tab sheets is where the program functions are selected to be added to or to change existing lines of programming code in the Create and Edit Program tab.

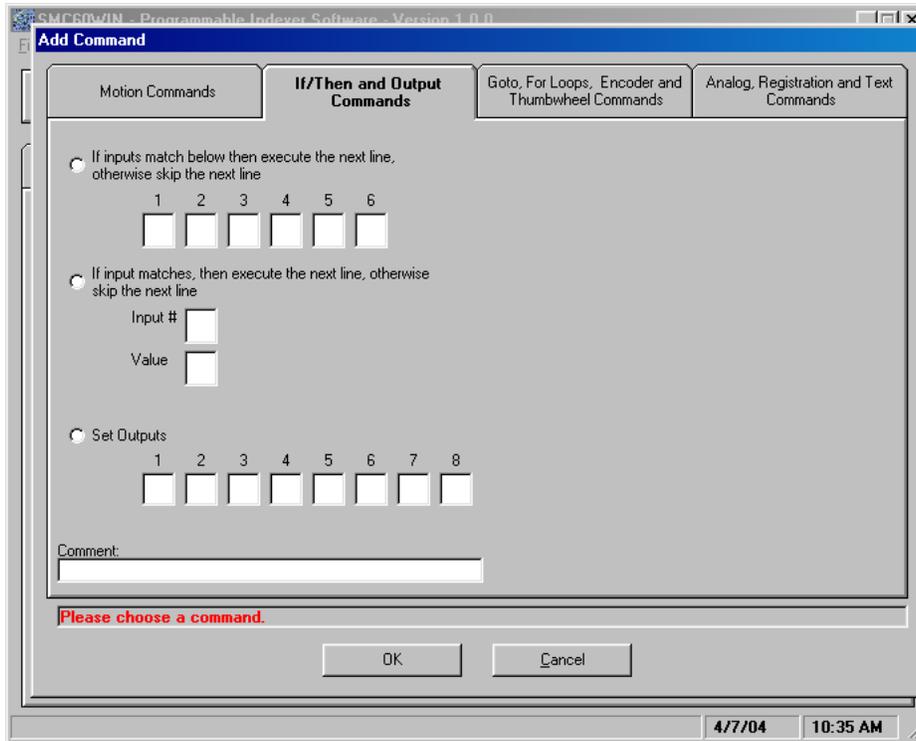
- To add a line of motion control, select appropriate command, and if required, enter the required value for that particular command. Then, click **OK**.
- Comment is optional, for any lines of code.
- The text box above the **OK** and **Cancel** buttons will display useful information about each command.

Add Tab Sheets - Motion Commands



| | |
|---------------------------|--|
| Accel/Decel | Set program acceleration & 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 Jog Speed | Set program jogging speed rate. (step/sec) |
| Set Position | Set motor position. |
| Direction CW (CCW) | Set direction to clockwise or counter-clockwise. |
| Motor Current ON (OFF) | Set the current in the motor on or off. |
| Wait ___ Milliseconds | This command allows the user to enter a delay in milliseconds. |
| 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. |
| Set Position | Allows user to change the position register in the controller. |
| Finish Move | Command will allow any motion command to be completed before continuing to the next line of code. This command should be used after every motion command. |
| Repeat Last Move | Command will repeat the previous index move. Do not use with encoder autocorrect enabled. |
| Home to Soft, Home Limits | Command will begin 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 begin motion seeking the home limit which will slow the motor down to base speed, reverse the direction and stop when the home limit is no longer triggered. |
| Slew (move continuously) | Command will ramp motor up to max speed and keep moving until triggered to stop. |
| Stop Soft | Ramp motor down to base speed and stop. |
| Stop Hard | Stop any motor motion immediately. |
| End of Program | This command is required as the last command in the program code. |

Add Tab Sheets - If/Then and Output Commands



| | |
|---|--|
| <p>If inputs match below then execute the next line, otherwise skip the next line</p> | <p>This conditional command allows the user to execute the next line of code if the inputs triggered match the given value. If the inputs do not match, the next line is skipped.</p> |
| <p>If input matches, then execute the next line, otherwise skip the next line</p> | <p>This conditional command allows the user to execute the next line of code if the specific input triggered matches the given value. If the input does not match, the next line is skipped.</p> |
| <p>Set Outputs</p> | <p>The outputs can be turned (on=1) or (off=0). These outputs can be used to trigger PLC operations, relays, solenoids, etc.</p> |

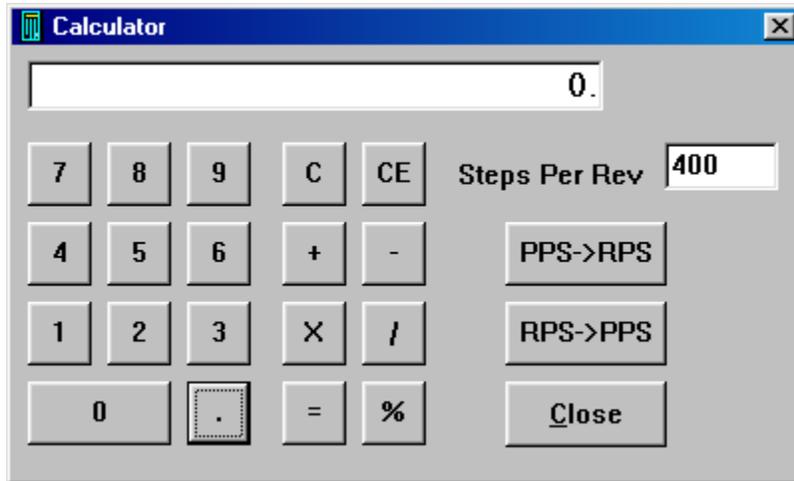
Add Tab Sheets - Goto, For Loops, Encoder and Thumbwheel Commands

| | |
|--------------------------|---|
| Goto | Command allows the program to jump to the specified label. |
| Label | Command inserts a label for goto and loop commands. |
| Return from Subroutine | Command will return to the last goto and execute the next line of code. |
| Outer Loop | Command allows a sequence of commands to be looped a specific number of times to a label. This label must be before the outer loop command. This command cannot be used within an inner loop. |
| Inner Loop | Command allows a sequence of commands to be looped a specific number of times to a label. This label must be before the inner loop command. This command can be used within an outer loop. |
| Thumbwheel Index Off | Command will disable the ability to use the thumbwheel switches to index. |
| Thumbwheel Index On | Command allows the user to use the thumbwheel index. Switch 2 must be in the thumbwheel position for the thumbwheel to be enabled. |
| Move Thumbwheel Index | Relative move command will allow motor to move the defined number of steps set by the thumbwheel switches. |
| Encoder Auto Correct Off | Command will disable the autocorrect of the motor. The encoder value can still be read. |
| Encoder Auto Correct On | Command will enable the use of the encoder and autocorrect the motor if needed, based on the encoder registers. |
| Delay | Command sets a time delay used for settling time needed after an index is finished: The delay occurs before the encoder count is read. |
| Motor Ratio | Command sets the number of encoder counts to one motor step. |
| Retries | Command sets the number of times the motor can autocorrect itself before erroring. When the autocorrect errors, Output 8 will be triggered. |
| Window | Command sets the number of encoder counts the motor is allowed to be off. |
| Reset Encoder Count | Command will reset the encoder count to 0 (zero). |

Add Tab Sheets - Analog, Registration and Text Commands

| | |
|-----------------------------|--|
| Analog Speed On (Off) | Sets the analog speed input feature on or off. |
| Set Analog Speed | Sets the maximum (running) speed based on the analog input. |
| Analog Speed Lower Limit | Sets the analog speed lower limit to the value specified. |
| Analog Speed Upper Limit | Sets the analog speed upper limit to the value specified. |
| Analog Position On (Off) | Sets the analog position input feature on or off. |
| Goto Analog Position | Motor will move to the position based on the analog input. |
| Analog Position Lower Limit | Sets the analog position lower limit to the value specified. |
| Analog Position Upper Limit | Sets the analog position upper limit to the value specified. |
| Output on the Fly On (Off) | Sets the output on the fly feature on or off. |
| 1st Output Position | Sets the position that the 1st output will trigger during and index. |
| Steps Between Outputs | Sets the number of steps between the triggered output. |
| # of Output Counts | Set the number of times the output will trigger. |
| Index on the Fly On (Off) | Sets the index on the fly feature on or off. |
| Registration Index | Sets the value of the index that will occur after the registration input is activated. |
| Send Text | Will send the entered text string back to the user. (20 characters maximum) |

Calculator



| | |
|---------------|--|
| 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 the Calculator. |

Section 4: Direct Talk Mode

Direct mode is used to directly control motion for real time movements through serial communication. The PCL601 controller has 40 commands, which are easy to remember for direct movement of a step motor.

COM Port Settings

Baud Rate: 38400
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 PCL601 will return a SMC60 + the current revision 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 correct: @0(carriage return)
incorrect: @ 0 (carriage return)

Command Summary:

| | | |
|------------------------------------|-------------------------------|---------------------------------|
| A - Acceleration/Deceleration | M - Max speed | + - Clockwise direction |
| B - Base speed | N - Number of steps | , - Stop soft |
| C - Steps between outputs | O - Set outputs | - - Counterclockwise direction |
| D - 1st output on the fly position | P - Absolute position | . - Stop hard |
| EA - Encoder autocorrect enabled | S - Go slew | / - Thumbwheel enabled |
| ED - Encoder delay | T - Motor current enabled | : - Analog position enabled |
| EM - Encoder motor ratio | V - Verify | ; - Analog speed enabled |
| ER - Encoder retries | Z - Position | [- Analog speed lower limit |
| ET - Encoder reset | ! - Error codes register |] - Analog speed upper limit |
| EW - Encoder window | \$ - Version number | ^ - Number of outputs |
| G - Go number of steps | % - Verify axis number | { - Analog position lower limit |
| H - Home | ' - Index on the fly enabled | } - Analog position upper limit |
| I - Read inputs | (- Output on the fly enabled | ~ - Set address of PCL601 |
| J- Fast jog speed | | |

A - Acceleration/Deceleration

Format: A[value]

Description: This command sets the acceleration profile which is an integer value between 100 and 9,999,999. The higher the value, the faster the motor acceleration.

Range: 100 - 9,999,999

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 than the maximum speed and fast jog speed.

Range: 1 - 5000

C - Number of steps between outputs during output on the fly

Format: C[value]

Description: This command sets the number of steps between when output 1 turns on during an output on the fly move. If only one output turn on is needed, set this value to 0. This command is used in conjunction with the output on the fly enabled “(“ command.

Range: 0 - 65535

D - 1st Output on the fly position

Format: D[value]

Description: This command sets the position at which output 1 will turn on during an output on the fly move. This command is used in conjunction with the output on the fly enabled “(“ command.

Range: 0 - 65535

EA - Encoder autocorrect enabled

Format: EA[0 or 1]

Description: This command will either enable or disable the encoder autocorrect function. To enable the function use a 1, to disable the function use a 0. When this function is enabled, the relative register is used to calculate the encoder position, therefore before the next move is made, the relative register needs to be set. This command is used in conjunction with the encoder delay “ED”, encoder ratio “EM”, encoder retries “ER” and encoder window “EW” commands.

ED - Encoder delay

Format: ED[value]

Description: This command sets the wait time a specified number of milliseconds after a relative index or absolute move is finished, before reading the encoder. This is used to remove the ringing that might be associated with the mechanics of the system. This command is used in conjunction with the encoder autocorrect "EA" command.

Range: 0 - 65535

EM - Encoder motor ratio

Format: EM[value]

Description: This represents the ratio for the number of encoder pulses to one motor step. An example is for a 1000 line quadrature encoder and a 400 step/revolution motor, the motor ratio is $(1000 * 4) / 400 = 10$. This command is used in conjunction with the encoder autocorrect EA command.

Range: 1 - 255

ER -Encoder retries

Format: ER[value]

Description: This is the number of times the PCL601 controller will try to autocorrect the motor before erroring. This command is used in conjunction with the encoder autocorrect EA command. When the autocorrect errors, Output 8 will be triggered.

Range: 0 - 255

ET - Encoder reset

Format: ET

Description: This command will reset the internal encoder count register to 0.

EW -Encoder window

Format: EW[value]

Description: This is the allowable error in encoder pulses (either plus or minus) from the desired position that is allowed before the motor autocorrects. This command is used in conjunction with the encoder autocorrect EA command.

Range: 0 - 255

G - Go number of steps

Format: G

Description: This command is used to send a set number of clocks out of the PCL601 controller. An N or P command must be entered before the G command. The ramp profile is specified by the B (base speed), M (max speed), and A (acceleration/deceleration) commands.

H - Home

Format: H[0 or 1]

Description:

Home Types: H0: In type 0 homing, the PCL601 will send clocks until a soft limit is reached, then ramp down to base speed. Clocks will continue at base speed until a home limit is reached. The ramp profile is specified by the B (base speed), M (max speed), and A (acceleration/deceleration) commands.

H1: In type 1 homing, the PCL601 will send clocks until a home limit is reached, ramp down to base speed, change directions and run at base speed until the release of the home limit input. The ramp profile is specified by the B (base speed), M (max speed), and A (acceleration/deceleration) commands.

I - Read inputs

Format 1: IR

Description: This command returns the binary value of the inputs to the PC. Since the inputs are pulled up internally (except input 1), they will return a high when they are open. For example; if all inputs are active (grounded), the command will return a 0. If all inputs are inactive (open), the command will return a 63. Input 1 is the LSB and input 6 is the MSB.

Format 2: I[input]

Description: This command returns the value of the selected input to the PC. If the input is open or high it will return a 1. If the input is grounded or low it will return a 0.

Ranged: 0 - 6

J - Fast jog speed

Format: J[value]

Description: This command sets the fast jog speed. This value must be set before motion begins and be greater than the base speed.

Range: 1 - 50000

M - Max speed and analog speed

Format 1: M[value]

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

Range: 1 - 50000

Format 2: M

Description: This command uses the voltage on input 1 to calculate and set the max speed. The analog speed must be enabled for this command to work. This command is used in conjunction with the analog speed “;”, the analog speed lower limit “[”, and the analog speed lower limit “]” commands. This value must be set before motion begins.

N - Number of steps

Format 1: N[value]

Description: This command sets the number of clocks for the PCL601 to send out following a G command. It is also used to set the registration index during and index on the fly move. Motion is not activated by this command; it only sets the index register.

Range: 0 - 8388607

Format 2: N

Description: This command reads the thumbwheel switches to set the number of clocks for the PCL601 to send out following a G command. For this command to work SW2 must be in the TWS position, and the thumbwheel enable bit must be enabled. Motion is not activated by this command, it only sets the index register. This command is used in conjunction with the thumbwheel enabled “/” command.

O - Set outputs

Format 1: OR[value]

Description: This command sets the output register according to the binary value entered. Output 1 is the LSB and output 8 is the MSB.

Range: 0 - 255

Format 2: O[output]=[0 or 1]

Description: This command sets the selected output either on or off. A 1 will turn the output on (0VDC) and a 0 will turn the output off (open).

Range: 0 - 8

P - Absolute position and analog position

Format 1: P[value]

Description: This command calculates and sets the number of clocks for the PCL601 to send out following a G command. Motion is not activated by this command; it only sets the register. ($N = P - Z$)

Range: -8388607 to +8388607

Format 2: P

Description: This command uses the voltage on input 1 to calculate and set the number of clocks for the PCL601 to send out following a G command. The analog position must be enabled for this command to work. Motion is not activated by this command, it only sets the register ($N = P - Z$). This command is used in conjunction with the analog position “.”, the analog position lower limit “{”, and the analog position lower limit “}” commands.

S - Go slew

Format: S

Description: This command will send clocks out to the PCL601. The only commands that can stop the clocks are; “.” (stop motion) or “,” (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.

T - Motor current enabled

Format: T[0 or 1]

Description: This command will control the On/Off output, which is designed to connect to the on/off input of Anaheim Automation’s step motor drivers. To energize and allow current to flow through the coil of the motor, set the value to 1. To de-energize and turn the current off to the motor, set the value to 0. This is a dedicated output, and not controlled with the output register.

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 shown below.

| Command | Description | Command | Description |
|---------|---|---------|--|
| A | Verify acceleration/deceleration | O | Verify outputs |
| B | Verify base speed | T | Verify motor current (1 is on, 0 is off) |
| C | Verify steps between outputs on the fly | Z | Verify position |
| D | Verify 1st output on the fly position | ' | Verify index on the fly enabled** |
| EA | Verify encoder autocorrect enabled** | (| Verify output on the fly enabled** |
| ED | Verify encoder delay | + | Verify Direction |
| EM | Verify encoder motor ratio | / | Verify thumbwheel index enabled** |
| EP | Verify encoder position | : | Verify analog position enabled** |
| ER | Verify encoder retries | ; | Verify analog speed enabled** |
| EW | Verify encoder window | [| Verify lower analog speed value |
| F | Verify if controller is busy |] | Verify upper analog speed value |
| J | Verify jog speed | ^ | Verify number of outputs |
| L | Verify Limits (Hard-Bit 0, Soft-Bit 1) | { | Verify lower analog position value |
| M | Verify max speed | } | Verify upper analog position value |
| N | Verify number of steps | | ** 1 is enabled, 0 is disabled |

Z - Position

Format: Z[value]

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: -8388607 to +8388607

! - Error codes register

Format: !

Description: This command requests the PCL601 controller to get the current error code and print it to the screen. For a description of the error codes see page 39.

\$ - Version number register

Format: \$

Description: This command requests the PCL601 controller to return its internal firmware version number.

% - Verify address register

Format: % (No address is needed before this function. @% will return the address)

Description: This command requests the PCL601 controller to return its internal address number to the PC or PLC.

' - Index on the fly enabled

Format: '[0 or 1]

Description: This command will either enable or disable the index on the fly function. To enable the function, use a 1. To disable the function use a 0. This command is used in conjunction with the number of steps "N" and go slew "S" commands.

(- Output on the fly enabled

Format: '([0 or 1]

Description: This command will either enable or disable the output on the fly function. To enable the function, use a 1. To disable the function use a 0. This command is used in conjunction with the number of steps between outputs "C", 1st output position "D", and number of outputs "^" commands.

+ - Clockwise

Format: +

Description: This command sets the direction output to clockwise.

, - Soft Limit Input Bit

Format: ,

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 PCL601 controller will continue to the set position and stop. In a slew type motion the PCL601 controller will ramp down and stop. In a home type motion the PCL601 controller will ramp down and run at base speed, until the home limit is activated.

-- Counter-Clockwise

Format: -

Description: This command sets the direction output to counterclockwise.

. - Stop Motion

Format: .

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

/ - Thumbwheel index enabled

Format: /[0 or 1]

Description: This command will either enable or disable the ability to use the thumbwheel switches for indexing. If enable, SW2 must be in the TWS position for the thumbwheel to be connected to the processor.

: - Analog position enabled

Format: :[0 or 1]

Description: This command will either enable or disable input 1 to be used to set the analog position. To enable the function, use a 1. To disable the function use a 0. This command is used in conjunction with the analog position lower limit "{" and analog position upper limit "}" commands.

; - Analog speed enabled

Format: ;[0 or 1]

Description: This command will either enable or disable input 1 to be used to set the analog speed. To enable the function, use a 1. To disable the function use a 0. This command is used in conjunction with the analog speed lower limit "[" and analog speed upper limit "]" commands.

[- Analog speed lower limit

Format: [[value]

Description: This command sets the lower limit that is used during the calculation following an M command for the analog speed input. This command is used in conjunction with the analog speed enabled ";" and max speed "M" commands.

Range: 1 - 50000

] - Analog speed upper limit

Format:][value]

Description: This command sets the upper limit that is used during the calculation following an M command for the analog speed input. This command is used in conjunction with the analog speed enabled ";" and max speed "M" commands.

Range: 1 - 50000

^ - Number of outputs during output on the fly

Format: ^[value]

Description: This command sets the number of times output 1 will turn on during an output on the fly move. This command is used in conjunction with the output on the fly enabled "(" command.

Range: 0 - 255

{ - Analog position lower limit

Format: {[value]

Description: This command sets the lower limit that is used during the calculation following a P command for the analog position input. This command is used in conjunction with the analog position enable ":" and absolute position "P" commands.

Range: 0 - 65535

} - Analog position upper limit

Format: }[value]

Description: This command sets the upper limit that is used during the calculation following a P command for the analog position input. This command is used in conjunction with the analog position enable ":" and absolute position "P" commands.

Range: 0 - 65535

~ - Set address register

Format: ~[value] (No address is needed before this function. @~[value] will set the address)

Description: This command sets the address for communication inside the PCL601 controller.

Range: 0 - 99

Section 5: Troubleshooting

Problem:

Cannot establish communications with the PCL601.

Possible Solutions:

- 1) Make sure the PCL601 controller has power. Is the Green LED on?
- 2) Check the RS232/RS485 connections.
- 3) Check for loose cable connections either on the PCL601 controller or COM Port.
- 4) Was the software installed successfully?
- 5) Go to **Setup | Com Port Settings** and verify COM port and baud rate settings.
- 6) Go to **Setup | Axis** and verify address settings are the same.
- 7) Click on **Setup |Connect** icon to communicate with the PCL601 controller.
- 8) If problems still exist, contact Anaheim Automation Tech Support.

Anaheim Automation, Inc.
Tech Support:

910 East Orangefair Lane
Anaheim, CA, 92801-1195
phone: (714) 992-6990
fax: (714) 992-0471
www.anaheimautomation.com

Problem:

There is no power to the PCL601 controller.

Possible Solutions:

- 1) Is the PCL601 controller connected to the appropriate power supply?
- 2) Check for any blown fuses in line with the PCL601 controller.
- 3) If problems still exist, contact Anaheim Automation, Inc. 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 PCL601 controller 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 PCL601 to clear an error.
- 3) Another way to clear an error is by using either the SMC60WIN software or the direct mode command instructions set.
- 4) The SMC60WIN can clear an error in the real time motion tab section by clicking on the verify parameters button.
- 5) The direct mode command “!” can clear an error by simply prompting the error code register to return the value back to the PC or PLC.

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.

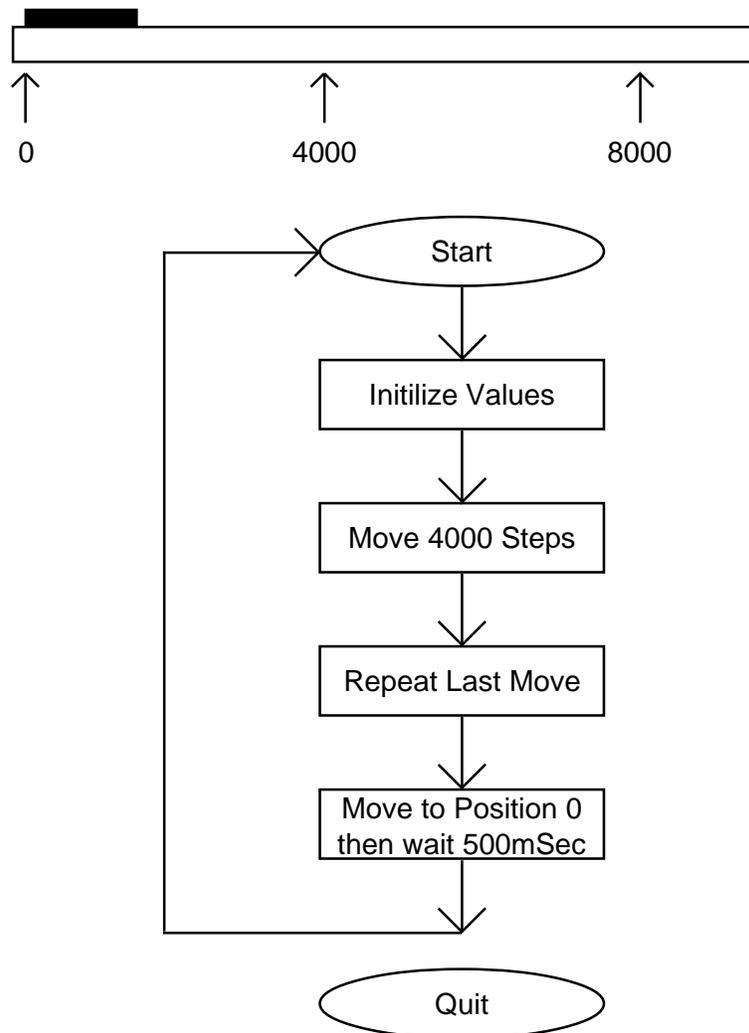
Error Codes

| Error Code | Type | Description |
|------------|------------------------|---|
| 1 | Receive Overflow Error | The serial communications had a receiving error. This is an internal error caused by the computer. |
| 2 | Encoder Error 1 | The encoder needed to correct the index. |
| 4 | Encoder Error 2 | The encoder could not finish the index. Output 8 is set with this error. |
| 8 | 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 a 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. |
| 1024 | Busy Error | The controller is busy indexing. The controller is sending out clocks to the driver and can not execute the next instruction. |
| 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 | Thumbwheel Read Error | There was an error reading the thumbwheel or the thumbwheel is not present. |

Section 6: Tutorial

Sample Program 1:

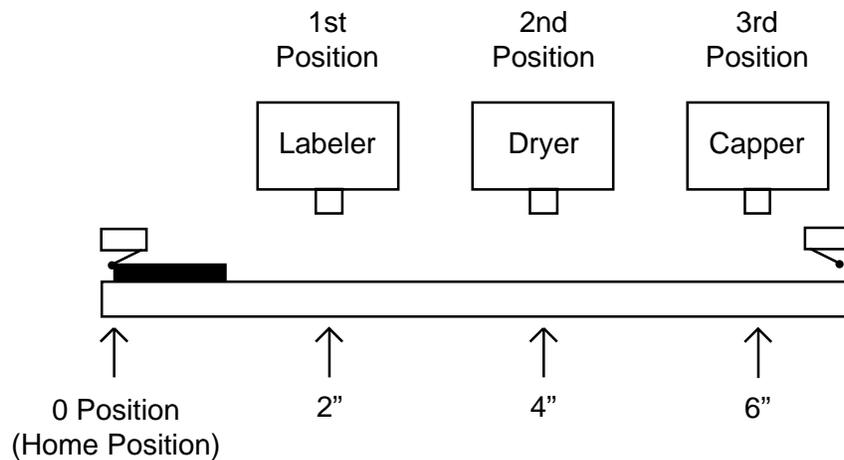
Sample Program 1 illustrates a typical application where a system moves to a specific position required. The sample program shows how to use the motion and goto instruction commands.



| Line | Command | Comments |
|------|-----------------------------------|-------------------------|
| 1 | Acceleration=1000 steps/(sec*sec) | |
| 2 | Base speed=500 steps/sec | |
| 3 | Max speed=5000 steps/sec | |
| 4 | Position register=0 | |
| 5 | Label=TOP | |
| 6 | Direction CW | |
| 7 | Go relative 4000 steps | Move to 1st position |
| 8 | Finish move | |
| 9 | Repeat last index | Move to 2nd Position |
| 10 | Finish move | |
| 11 | Go to position 0 | Return to Zero Position |
| 12 | Wait 500 msec | Wait 0.5 Seconds |
| 13 | Goto TOP | Repeat moves |
| ▶ 14 | End of program | |

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, loop routines and encoder routine.



| Line | Command | Comments |
|------|--------------------------------------|-------------------------------|
| 1 | Acceleration=1000 steps/(sec*sec) | |
| 2 | Base speed=500 steps/sec | |
| 3 | Max speed=5000 steps/sec | |
| 4 | Encoder delay=10 msec | Wait 10mS for encoder ringing |
| 5 | Encoder motor ratio=110 counts/step | 1000 Lines: 400 step/rev |
| 6 | Encoder retries=3 | Allow three retries |
| 7 | Encoder window=10 counts | Allow 1 motor step error |
| 8 | Turn on encoder autocorrect | |
| 9 | Label=TOP | |
| 10 | Direction CCW | |
| 11 | Home type 1 (home limit switch only) | Home to a physical switch |
| 12 | Finish move | |
| 13 | Position register=0 | Reset position register to 0 |
| 14 | Reset the encoder position to 0 | Reset the encoder count |
| 15 | Direction CW | |
| 16 | Go relative 4000 steps | Move to 1st Position |
| 17 | Finish move | |
| 18 | Go relative 4000 steps | Move to 2nd Position |
| 19 | Finish move | |
| 20 | Wait 1000 msec | |
| 21 | Go relative 4000 steps | Move to 3rd Position |
| 22 | Finish move | |
| 23 | Outer loop=3 times to TOP | |
| 24 | End of program | |

Sample Program 3:

Sample Program 3 illustrates the setup and operation of the output on the fly function, and the use of the if/then statement. The system is first homed using home type 0, waits for input 1 to be a value of 0 (grounded) and then is indexed 10,000 steps. During this index, output 1 needs to be turned on every 1000 steps 5 times starting at position 2000. At the end of the index, output 1 is then turned on again for 1 mS and the unit is then sent back the 10,000 steps to position 0, but the output should not be turned on. The unit then repeats waiting for input 1 to be 0 and indexes again.

| Line | Command | Comments |
|------|--|--------------------------------|
| 1 | Acceleration=1000 steps/(sec*sec) | |
| 2 | Base speed=500 steps/sec | |
| 3 | Max speed=5000 steps/sec | |
| 4 | Direction CCW | |
| 5 | Home type 0 (soft and home switch) | |
| 6 | Finish move | |
| 7 | Position register=0 | |
| 8 | Output on at position=2000 | |
| 9 | Steps between outputs=1000 steps | |
| 10 | # of output counts=5 | |
| 11 | Label=TOP | |
| 12 | If input 1=1, then execute the next line | Loop to TOP until input 1 is 0 |
| 13 | Goto TOP | |
| 14 | Enable output on the fly | |
| 15 | Direction CW | |
| 16 | Go relative 10000 steps | |
| 17 | Finish move | |
| 18 | Output register=1 | Turn on Output 1 |
| 19 | Wait 1 msec | |
| 20 | Output register=0 | Turn off Output 1 |
| 21 | Disable output on the fly | |
| 22 | Go to position 0 | |
| 23 | Finish move | |
| 24 | Goto TOP | |
| ▶ 25 | End of program | |

Sample Program 4:

Sample Program 4 illustrates the setup of the analog speed function and the use of “indexing-on-the-fly.” The system is first homed using home type 1. The next step is to wait for the input register to read 110111 (input 2 must be high while input 3 is low, all other inputs are not used and input 1 is masked high due to the analog function being used). The third step is to slew using the analog input as the maximum speed between 5000 and 10000 steps/revolution. When input 2 is switched low, the unit will index 2500 more steps and ramp down to base speed and stop. At the end of the index, output 1 will turn on for 100 mS. After the output is turned off the unit will be sent back to position 0. The program then is sent back to repeat itself, waiting for the input register to be 110111 again.

| Line | Command | Comments |
|------|--|------------------------------|
| 1 | Acceleration=1000 steps/(sec*sec) | |
| 2 | Base speed=500 steps/sec | |
| 3 | Analog speed lower limit=5000 steps/sec | |
| 4 | Analog speed upper limit=10000 steps/sec | |
| 5 | Enable analog speed | |
| 6 | Enable index on the fly | |
| 7 | Direction CCW | |
| 8 | Home type 1 (home limit switch only) | Home to a physical switch |
| 9 | Finish move | |
| 10 | Position register=0 | Reset position register to 0 |
| 11 | Label=TOP | |
| 12 | If input register=59, then execute the next line | Input 2 high, Input 3 low |
| 13 | Goto INPUTGOOD | |
| 14 | Goto TOP | |
| 15 | Label=INPUTGOOD | |
| 16 | Get max speed from analog input | |
| 17 | Registration index=2500 steps | |
| 18 | Direction CW | |
| 19 | Slew | |
| 20 | Finish move | |
| 21 | Output register=1 | Turn output 1 on |
| 22 | Wait 100 msec | |
| 23 | Output register=0 | Turn output 1 off |
| 24 | Go to position 0 | |
| 25 | Finish move | |
| 26 | Goto TOP | |
| ▶ 27 | End of program | |

Sample Program 5:

Sample Program 5 illustrates a typical 3 axis application where one PCL601 (Axis 0) is controlling the other two axes (Axis 1 and Axis 2) by using the send text string commands. The program first sets the accelerations, base speeds and maximum speeds for each axis. It then is enabling Axis 0 and Axis 1 to use the thumbwheel switch that is connected to each unit for indexing. The direction is then set for Axis 0 and Axis 1 and these two axes are then told to index the distance set in each thumbwheel at the same time. The program then waits for the motion to finish before it tells Axis 2 to move. Axis 2 first moves in the negative direction waits for the index to finish, and then moves back in the positive direction. After Axis 2 is finished moving, Axis 0 and Axis 1 are both sent back to their zero positions at the same time. The program is then repeated.

| Line | Command | Comments |
|------|-----------------------------------|------------------------------|
| 1 | Acceleration=1000 steps/(sec*sec) | Axis0 Accel = 1000 |
| 2 | Send Text String @1A1000 | Axis1 Accel = 1000 |
| 3 | Send Text String @2A100 | Axis2 Accel = 100 |
| 4 | Base speed=500 steps/sec | Axis0 Base = 500 |
| 5 | Send Text String @1B500 | Axis1 Base = 500 |
| 6 | Send Text String @2B100 | Axis2 Base = 100 |
| 7 | Max speed=5000 steps/sec | Axis0 Max = 5000 |
| 8 | Send Text String @1M5000 | Axis1 Max = 5000 |
| 9 | Send Text String @2M500 | Axis2 Max = 500 |
| 10 | Enable thumbwheel | Enable Axis0 Thumbwheel |
| 11 | Send Text String @1/1 | Enable Axis1 Thumbwheel |
| 12 | Label=TOP | |
| 13 | Direction CW | Axis0 CW |
| 14 | Send Text String @1+ | Axis1 CW |
| 15 | Send Text String @1N | Set Distance with thumbwheel |
| 16 | Go relative thumbwheel index | Axis0 |
| 17 | Send Text String @1G | Axis1 Go relative thumbwheel |
| 18 | Finish move | Wait for moves to end |
| 19 | Wait 500 msec | |
| 20 | Send Text String @2- | Axis2 CCW |
| 21 | Send Text String @2N400 | Set axis2 index |
| 22 | Send Text String @2G | Axis2 go relative |
| 23 | Wait 5000 msec | |
| 24 | Send Text String @2+ | Axis2 CW |
| 25 | Send Text String @2G | Axis2 go relative |
| 26 | Wait 1000 msec | |
| 27 | Send Text String @1P0 | Axis1 set goto position 0 |
| 28 | Go to position 0 | Axis0 Go to position 0 |
| 29 | Send Text String @1G | Axis1 Go to position 0 |
| 30 | Finish move | Wait for moves to end |
| 31 | Wait 1000 msec | |
| 32 | Goto TOP | Return and repeat |
| ▶ 33 | End of program | |

Appendix 1: ASCII Table for Direct Mode

| ASCII Symbol | Hex Value | ASCII Symbol | Hex Value | ASCII Symbol | Hex Value |
|--------------|-----------|-----------------|-----------|--------------|-----------|
| 0 | 30 | J | 4A | # | 23 |
| 1 | 31 | K | 4B | \$ | 24 |
| 2 | 32 | L | 4C | % | 25 |
| 3 | 33 | M | 4D | " | 27 |
| 4 | 34 | N | 4E | (| 28 |
| 5 | 35 | O | 4F | + | 2B |
| 6 | 36 | P | 50 | , | 2C |
| 7 | 37 | Q | 51 | - | 2D |
| 8 | 38 | R | 52 | . | 2E |
| 9 | 39 | S | 53 | : | 3A |
| A | 41 | T | 54 | ; | 3B |
| B | 42 | U | 55 | @ | 40 |
| C | 43 | V | 56 | [| 5B |
| D | 44 | W | 57 |] | 5D |
| E | 45 | X | 58 | ^ | 5E |
| F | 46 | Y | 59 | { | 7B |
| G | 47 | Z | 5A | } | 7D |
| H | 48 | Carriage Return | 0D | ~ | 7E |
| I | 49 | ! | 21 | | |

Appendix 2: Firmware Revisions

Version 1.00 - Initial Release.

Version 1.10 - Fixes Lockup on RS485 Communication.

Added capability to stop the motor on an Encoder Error after the Encoder Retries number has been reached.

Version 2.00 - Added capability for the Encoder Ration to have 2 decimal points. Also when the encoder retries fails, Output 8 is set. This version of firmware also requires an upgrade to the SMC60WIN version 2.00.

Version 2.10 - Fixes the update of the Analog Input when the pot was not changed after a manual update of the Max Speed, or Goto Position registers.

Added the capability to verify the Hard and Soft Limit inputs.

Copyright

Copyright 2004 by Anaheim Automation, Inc.. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of Anaheim Automation, Inc., 910 East Orangefair Lane, CA, 92801. The only exception to this would be use of the program examples in this manual.

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, Inc. 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, Inc. 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, Inc. 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, Inc.

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

ANAHEIM AUTOMATION, INC.