PCL602USB
Dual-Axis Programmable Step Motor Controller

User's Guide

ANAHEIM AUTOMATION, INC.

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

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

#L011364

July 2018

Table of Contents

Section 1: Introduction

The PCL602 is a dual-axis step motor controller containing 2 Kbytes of nonvolatile stored programming space and quadrature encoder feedback for each of its axes. It provides flexible and independent control of two independent stepper motors from computers, or any machine controller with a USB port. Additionally, the user can write a program and then have the PCL602 autostart the program upon power up. This makes the device capable of standalone operation as an embedded machine controller. The easy-to-use Windows software, SMC62WIN, can be used to directly control motion and to program the PCL602. The PCL602 also has the ability for real time functions. “Direct Mode” is used to directly control the device and execute real time moves through serial communication.

The PCL602 has 40 commands, which are easy to remember for direct movement of a stepper motor. The PCL602 communicates via a Universal Serial Bus (USB). A windows driver is provided to communicate with the PCL602 through a virtual comport from the PC. This driver will turn any USB port into a virtual comport, thus enabling simple serial programs to send information to the PCL602. To use the SMC62WIN software, the virtual comport driver must be installed. Special functions of the PCL602 include 16 programmable open collector outputs (8 per axis) and 6 TTL, CMOS and 24V compatible inputs, a quadrature encoder input for each axis 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 PCL602 can be powered with a DC voltage of wither 5VDC or 8-24VDC and uses only 600 milliwatts at 24VDC and only 400 milliwatts at 5VDC (with no connections to terminal blocks).

Description

The PCL602 dual-axis 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 for each of its two axes. On absolute positioning moves, the PCL602 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 PCL602 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 PCL602 also has a high level programming command set that includes: branching, looping, conditional statements, time delays, text strings, user defined variable registers, 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 16 programmable open-collector outputs are provided. 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 PCL602 has a built-in programmable reset circuit. Reset is automatic on power-up, or by pressing the external reset button. A downloadable package available at **www.anaheimautomation.com/downloads/software/software.php** contains this user’s manual, along with the SMC62WIN software, PCL602 program examples, and USB drivers. The software allows you to write and change programs that are to be stored in the PCL602 for autostart use, and also upload the program that is stored in the PCL602 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 PCL602. One is to directly talk to the PCL602 by using Direct Talk Mode. This is usually used with a computer, where the computer gives the PCL602 serial commands. A windows driver is provided to communicate with the PCL602 through a virtual comport from the PC. The driver can be found in the drivers folder in the device's downloadable software package.

The second way to give commands to the PCL602 is to use the software program SMC62WIN to either manually control, or to write and send programs. This method is used when the PCL602 is the main controller. For example: A PCL602 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, 16 or less outputs and one or two motion axes can utilize a PCL602 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 virtual comport that communicates with the PCL602 thorugh the computer's USB port accepts a baud rate of 38400 only** .

Dual-Axis / Single-Axis Operation

The PCL602 is capable of dual-axis operation allowing the user to independently control two separate stepper motors. The user can nonetheless choose to use the device either as a dual-axis or as a single-axis stepper motor controller. When taking advantage of the dual-axis capabilities of the PCL602, the user can send commands to the controller that are directed to a specific axis. The axis being addressed in the command is specified by either the 'X' or the 'Y' ASCII character (See page 28 for more details). In the case of using the PCL602 as a single-axis controller, the user can use either set of axis control inputs and outputs and address that axis specifically in the commands sent to the controller. Note that the user can still use the outputs of the axis not being used by addressing them properly.

Device Address

Each PCL602USB is addressed using a programmable register allowing the PC to address up to 99 controllers. The Default device address is "0". To change the device address, use the SMC62WIN software or the "~" command. To verify the current device address, use the SMC62WIN software or the "%" command. The device address designation is nonvolatile and will remain the same until changed by the user.

Interfacing With A Driver

The PCL602 controller was designed to control either one or two step motor drivers. For drivers with "*Opto-isolated*" inputs, the PCL602 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 PCL602 to the negative terminals of the corresponding inputs on the driver. For a driver with "*TTL/CMOS"* inputs, the PCL602 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 PCL602 to the corresponding inputs of the driver respectively.

Status LEDs

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. The RX LED will blink when data is being received from the PC. The TX LED will blink when data is being transmitted back to the PC.

Electrical Specifications

Power Requirements With No Connections:
5VDC @ 80mA or
8VDC to 24VDC @ 60mA

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

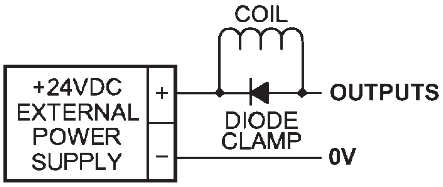
Baud Rate:
38400 Baud, Fixed

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

Outputs (16 programmable):
Open Drain Type
40V, 100mA

Output1(X & Y) active low time for output on the fly:
50uS

Encoder and Driver Outputs:
+5VDC Output, 1000mA (total)



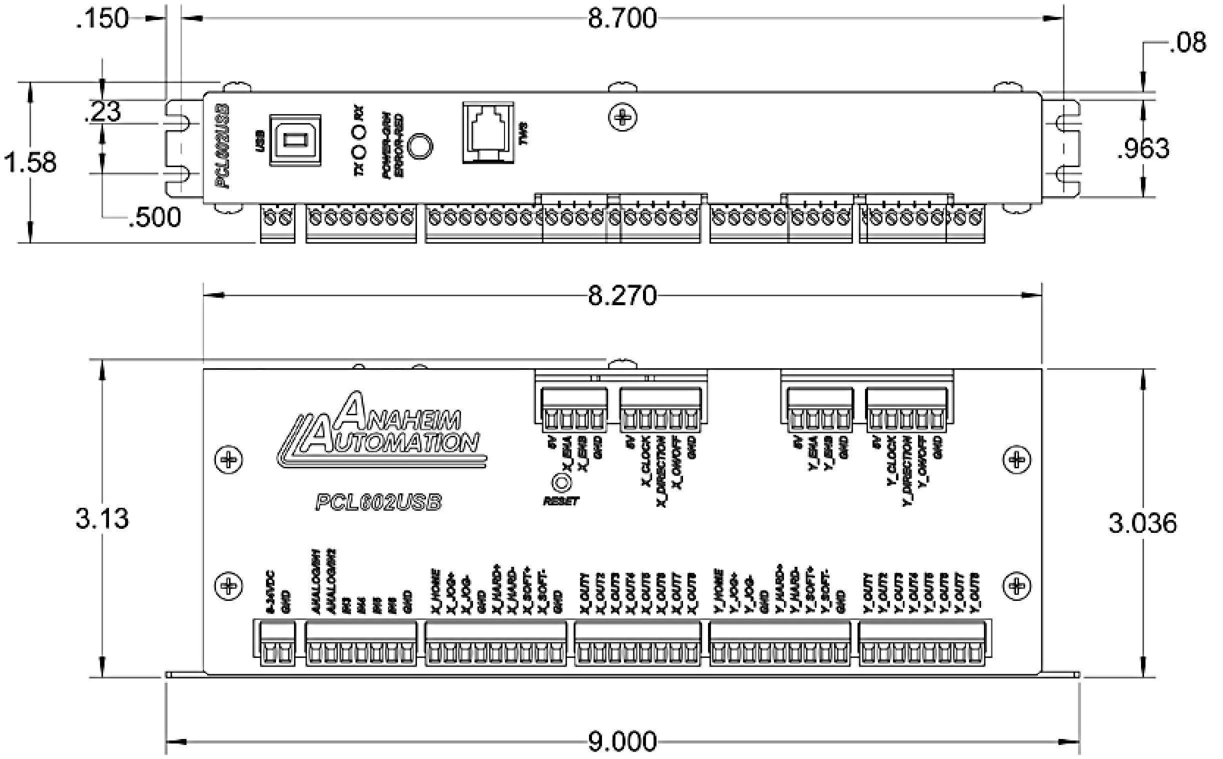
Note: For inductive loads, customers must connect a clamping diode to protect from flyback voltage spikes.

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.

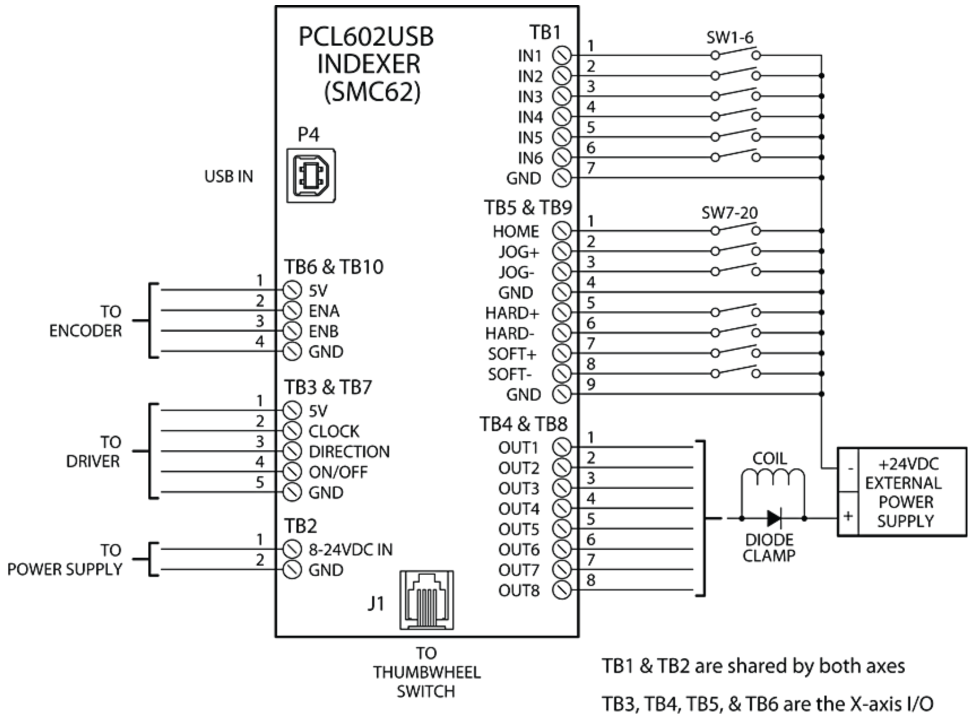
Part Number	Description
PCL602USB	Featured dual-axis step motor controller with encoder feedback.
PCL601USB	Single-axis step motor controller with encoder feedback.
MBC25081TB	Stepper Motor Driver with DC input and 2.5A output current
MBC12101	Stepper Motor Driver with DC input and 10A output current
TWS7	Seven position thumbwheel switch
AAUSB-AB-6	6-foot USB cable with one USB-A male connector and one USB-B male connector.
PSAM24V2.7A	Power supply for PCL602. (24V@2.7A)

Dimensions



All units are in inches

Wiring Diagrams



TB1 & TB2 are shared by both axes

TB3, TB4, TB5, & TB6 are the X-axis I/O

TB7, TB8, TB9, & TB10 are the Y-axis I/O

Terminal Descriptions

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

Position	Description - Encoder (X and Y)
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 (X and Y)
1	Home Limit
2	Jog +
3	Jog -
4	Ground
5	Hard Limit +
6	Hard Limit -
7	Soft Limit +
8	Soft Limit -
9	Ground

Position	Description - Inputs
1	Input 1 - Analog input X
2	Input 2 - Analog input Y
3	Input 3 - Index on the fly input X
4	Input 4 - Index on the fly input Y
5	Input 5
6	Input 6
7	Ground

Position	Description - Outputs (X and Y)
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 (X and Y)
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 USB communication and is labeled USB.
J1	This connector is for the thumbwheel module and is labeled TWS.

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 8,388,607)

Move to Position: The move to position command specifies the next absolute position to go to. The PCL602 controller automatically sets the direction and number of steps needed to go to that position. (Range: -8,388,607 to +8,388,607)

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: -8,388,607 to +8,388,607)

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 by grounding the appropriate input. To jog a motor, it is necessary to ground the jog input for the direction desired. By grounding one of the jog inputs, the user causes the motor to start running at base speed and immediately ramp to the set job speed at the set acerleartion value. 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 65,535 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 50,000 and the lower limit < upper limit)

Analog calculations.	**Example1:**	**Example2:**
(Upper-Lower) * (Voltage/5) = X	(5000 - 0) * (2 / 5) = 2000	(5000 - 1000) * (3.2 / 5) = 2560
Lower + X = Position or Frequency	0 + 2000 = 2000	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 darlingtons. 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 8,388,607)

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 65,535)

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 PCL602 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 PCL602 will continue to execute the program. If it encounters a command that cannot be used when the motor is moving, the PCL602 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 PCL602 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 PCL602. 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 65,535)

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 PCL602 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 PCL602 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/sec^2 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 50,000)

Jog Speed: The jog speed sets the jog rate. Jog (+/-) is used to run at base speed and accelerate to the set job 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 50,000)

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 PCL602 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: SMC62WIN Software

The SMC62WIN software is a handy utility specially designed for the PCL602's dual-axis step motor controller. Connecting your PC to the PCL602, via a USB calbe, the SMC62WIN software can easily perform the following tasks:

- Exercise and monitor the PCL602 controller
- Write and edit stored programs for standalone operation
- Directly communicate with the PCL602 controller

Installation

Software

- The SMC62WIN is supplied on a CD, containing the setup program and the SMC62WIN software, PCL602 manual and sample programs.
- SMC62WIN is compatible with all versions of Windows including Windows XP, Windows Vista, and Windows 7.

Windows XP/Vista/7 Installation

1) Download software from the Anaheim Automation website.
2) Unzip the file.
3) Double click on the **setup** application file.

Getting Started

1) Double click on the SMC62WIN icon to run the SMC62WIN software.

2) Apply power to the PCL602 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)

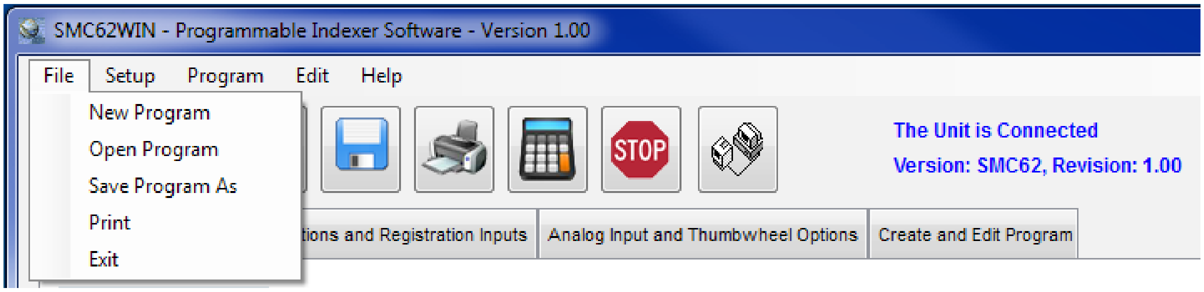
5) Establish communications with the PCL602 by clicking on the Connect Icon, or select Setup | Connect from the menu bar. If the unit is connected properly, the program will notify you when communications has been established. (Ctrl+C is a shortcut)

Changing the COM Port Number of the USB Port

1) From **Device Manager**, select **View Device by Type**, then **Ports(COM & LPT)**. Select the USB serial port and click **Properties**. Select the **Port Settings** tab, then click **Advanced**.

2) Choose the required COM port number from the list and click **OK**.

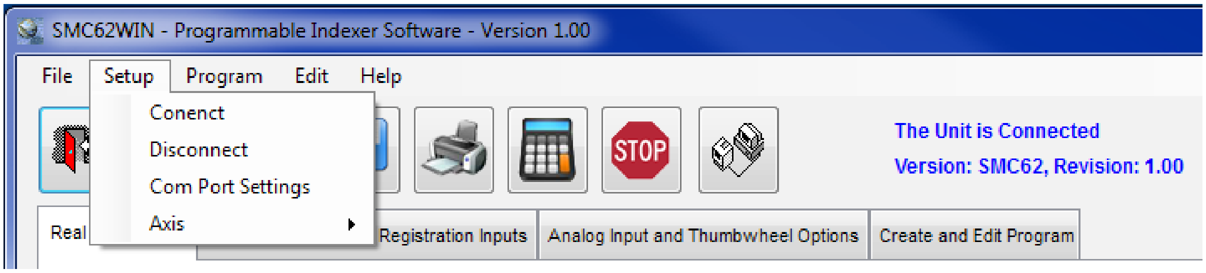
File Menu


SMC62WIN - Programmable Indexer Software - Version 1.00
File
Setup
Program
Edit
Help
New Program
Open Program
Save Program As
Print
Exit
STOP
tions and Registration Inputs
Analog Input and Thumbwheel Options
Create and Edit Program

The Unit is Connected
Version: SMC62, Revision: 1.00

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 SMC62WIN software.

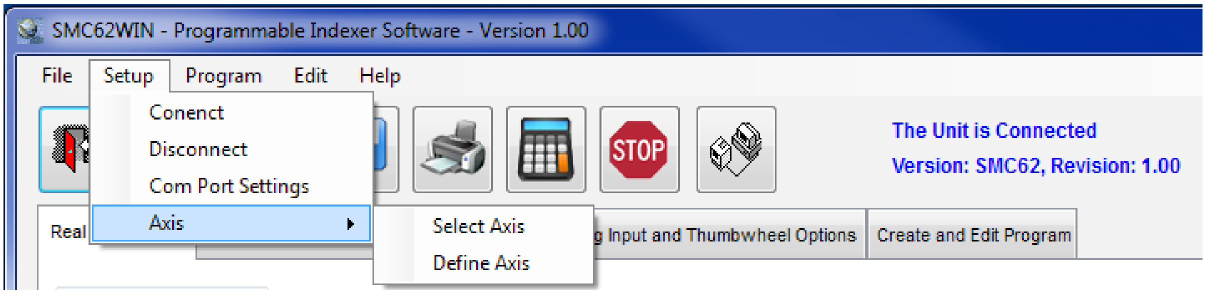
Setup Menu



The Unit is Connected
Version: SMC62, Revision: 1.00

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.

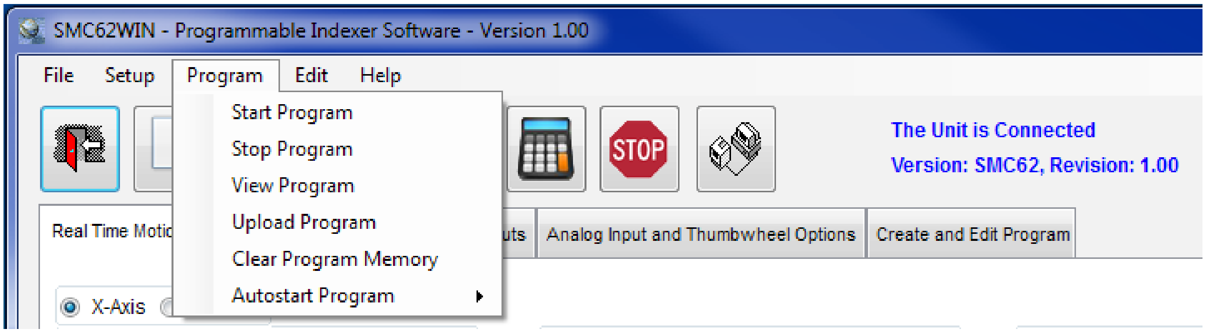
Setup - Axis Menu



The Unit is Connected
Version: SMC62, Revision: 1.00

Select Axis	Sets the axis select parameter in the SMC62WIN software. (1-99)
Define Axis	Sets the programmable address in the controller. (1-99)

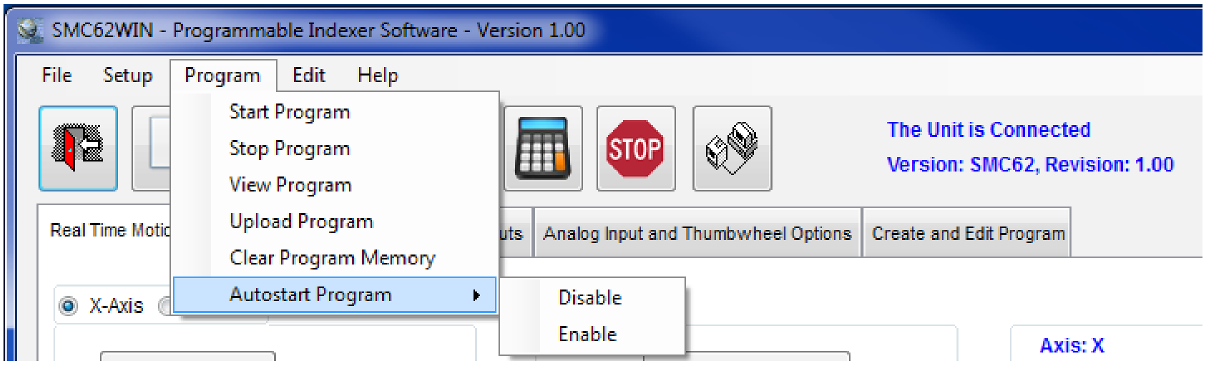
Program Menu



The Unit is Connected
Version: SMC62, Revision: 1.00

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.

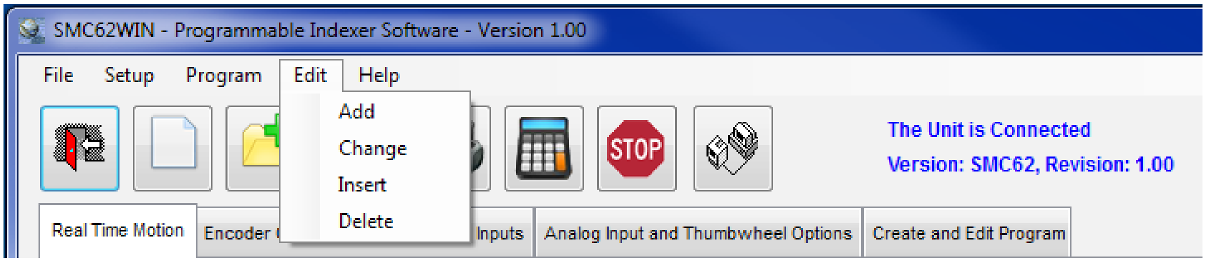
Program - Autostart Program Menu



The Unit is Connected
Version: SMC62, Revision: 1.00
Axis: X

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

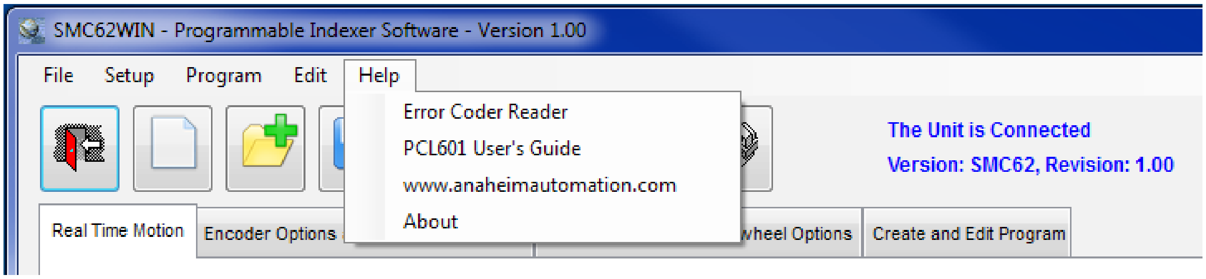
Edit Menu



The Unit is Connected
Version: SMC62, Revision: 1.00

Add	Adds a new line of code to the end of the progam.
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.

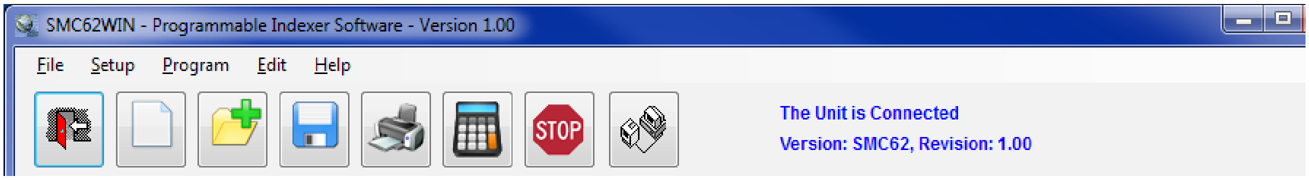
Help Menu

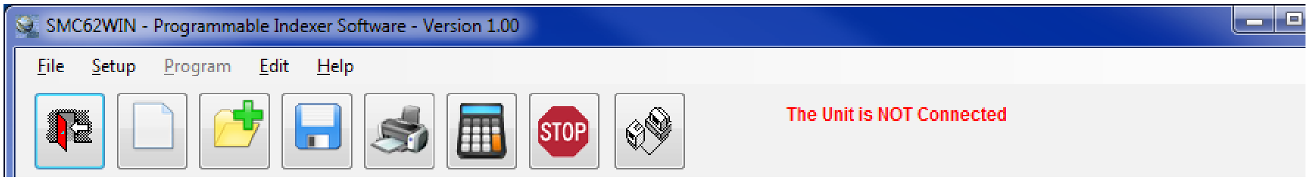


Error Coder Reader	Utility to read the error code generated by the PCL602 controller.
PCL602 User's Guide	Opens up the User's Guide in .pdf format.
www.anaheimautomation.com	Opens up the Anaheim Automation Website.
About	Displays the version of the SMC62WIN and contact information.

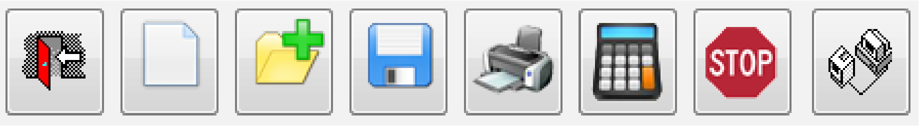
"The Unit is Connected" / "The Unit is NOT Connected"

On the right of the Toolbar, the user will find the communication status of the PCL602 controller. If communications are not established, please refer to the Troubleshooting Section.


SMC62WIN - Programmable Indexer Software - Version 1.00
File
Setup
Program
Edit
Help
STOP
The Unit is Connected
Version: SMC62, Revision: 1.00


SMC62WIN - Programmable Indexer Software - Version 1.00
File
Setup
Program
Edit
Help
STOP
The Unit is NOT Connected

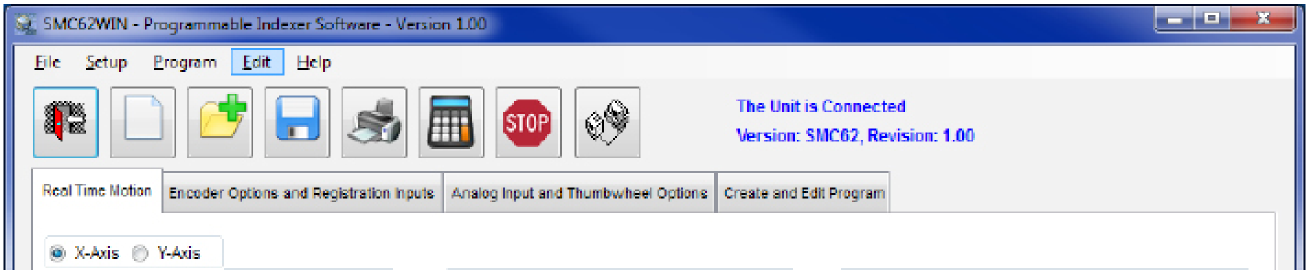
Toolbar



Exit New Open Save Print Calculator Stop Connect

Exit	Exit the SMC62WIN 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



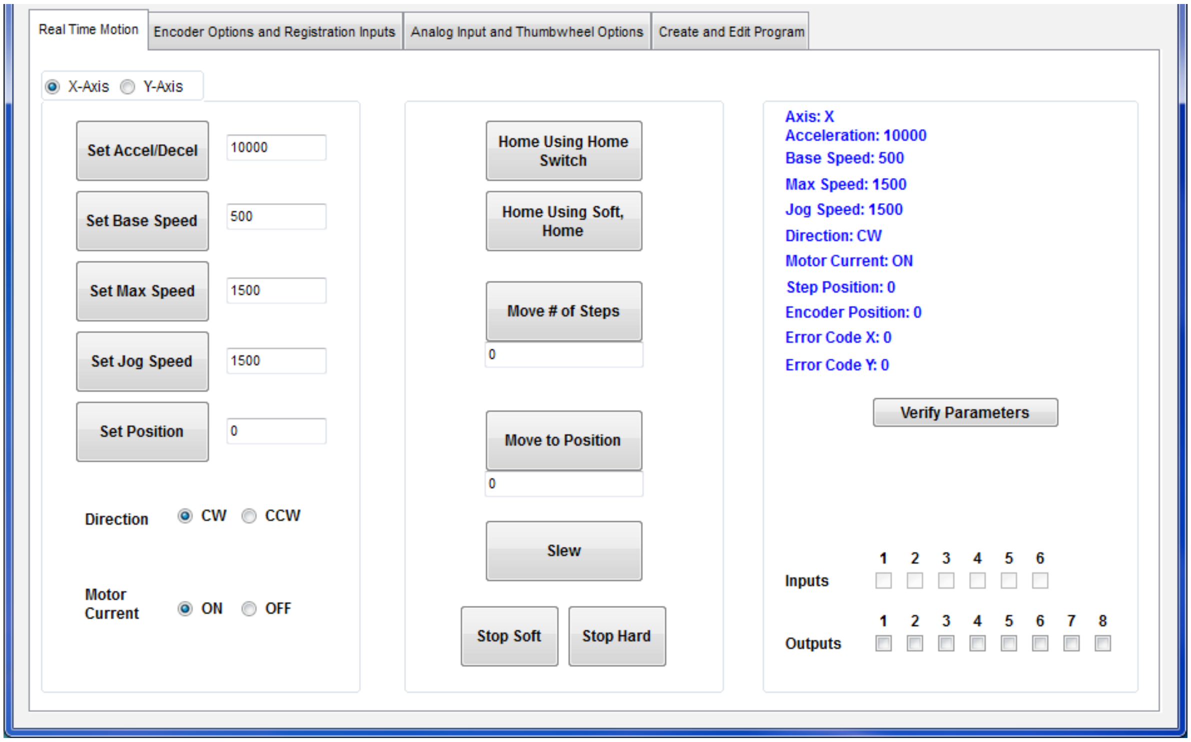
SMC62WIN - Programmable Indexer Software - Version 1.00

The Unit is Connected

Version: SMC62, Revision: 1.00

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 PCL602 stored programs.
X-Axis / Y-Axis	Selects the axis that is to execute the commands being sent.

Tab Sheets - Real Time Motion



● X-Axis ○ Y-Axis

Set Accel/Decel: 10000
Set Base Speed: 500
Set Max Speed: 1500
Set Jog Speed: 1500
Set Position: 0

Direction ● CW ○ CCW

Motor Current ● ON ○ OFF

Move # of Steps: 0

Move to Position: 0

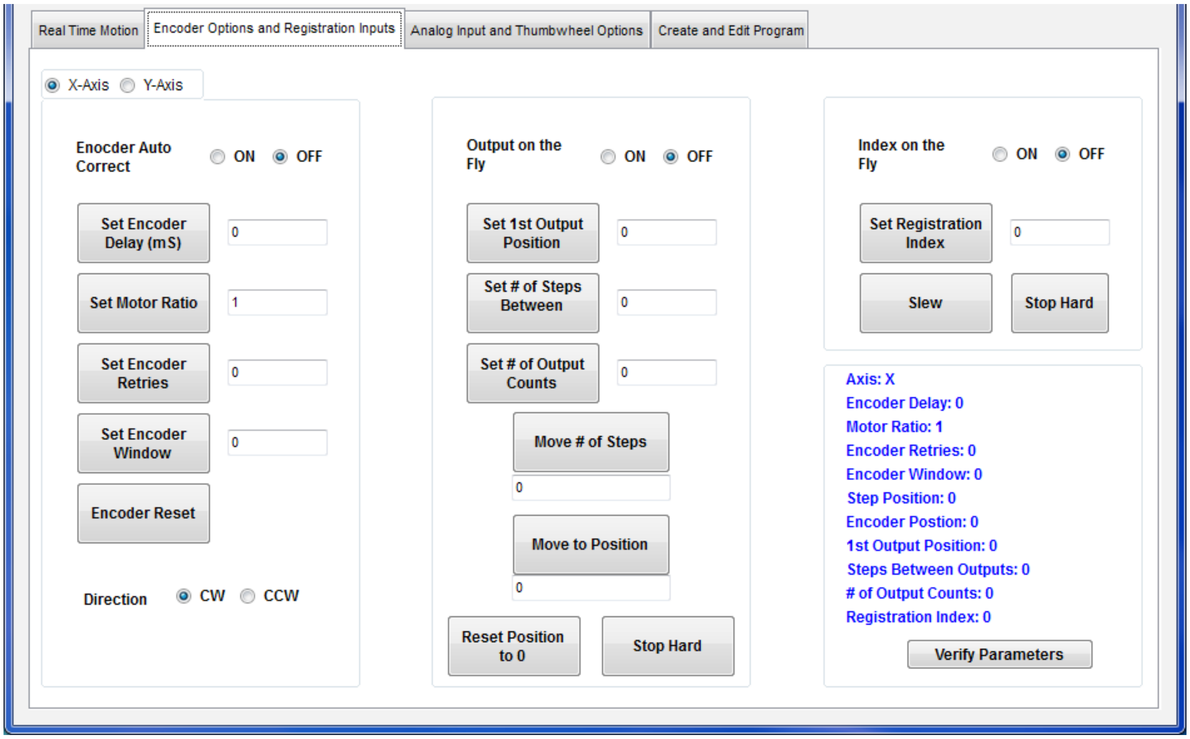
Axis: X
Acceleration: 10000
Base Speed: 500
Max Speed: 1500
Jog Speed: 1500
Direction: CW
Motor Current: ON
Step Position: 0
Encoder Position: 0
Error Code X: 0
Error Code Y: 0

Inputs 1 □ 2 □ 3 □ 4 □ 5 □ 6 □

Outputs 1 □ 2 □ 3 □ 4 □ 5 □ 6 □ 7 □ 8 □

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 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



● X-Axis ○ Y-Axis

Enocder Auto Correct ○ ON ● OFF

Set Encoder Delay (mS) 0

Set Motor Ratio 1

Set Encoder Retries 0

Set Encoder Window 0

Direction ● CW ○ CCW

Output on the Fly ○ ON ● OFF

Set 1st Output Position 0

Set # of Steps Between 0

Set # of Output Counts 0

Move # of Steps 0

Move to Position 0

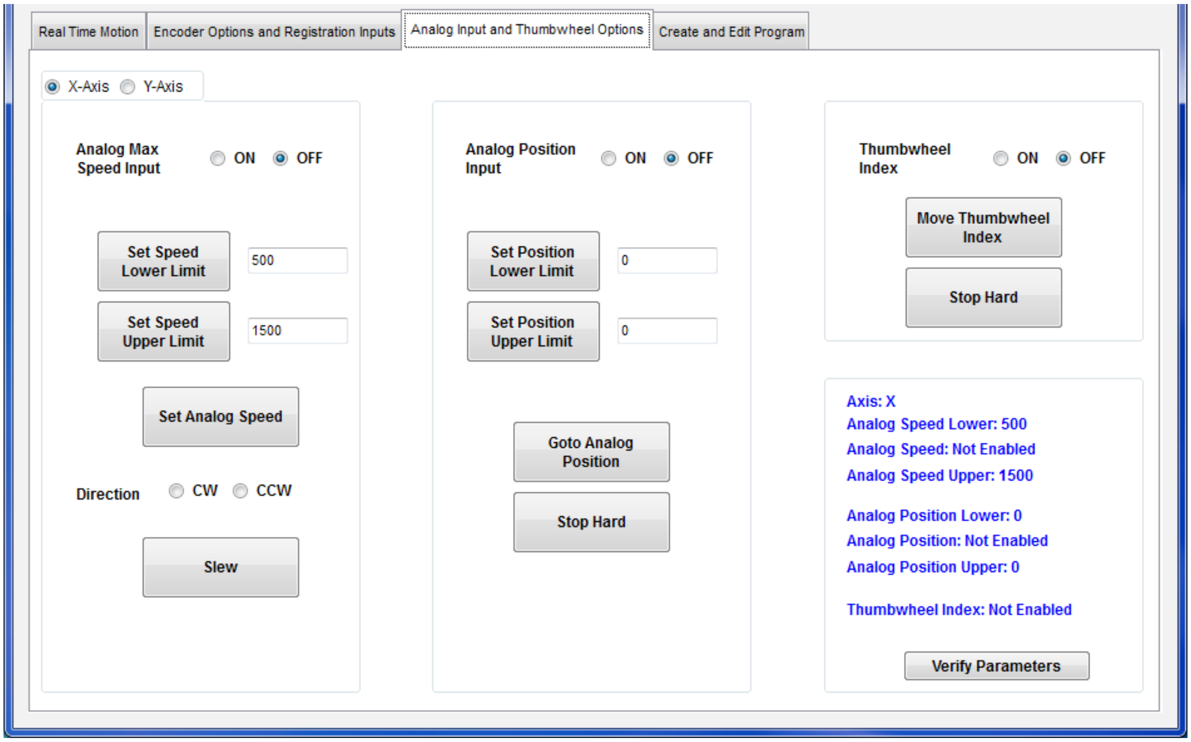
Index on the Fly ○ ON ● OFF

Set Registration Index 0

Axis: X
Encoder Delay: 0
Motor Ratio: 1
Encoder Retries: 0
Encoder Window: 0
Step Position: 0
Encoder Postion: 0
1st Output Position: 0
Steps Between Outputs: 0
# of Output Counts: 0
Registration Index: 0

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



● X-Axis ○ Y-Axis

Analog Max Speed Input ○ ON ● OFF

Set Speed Lower Limit 500

Set Speed Upper Limit 1500

Direction ○ CW ○ CCW

Analog Position Input ○ ON ● OFF

Set Position Lower Limit 0

Set Position Upper Limit 0

Thumbwheel Index ○ ON ● OFF

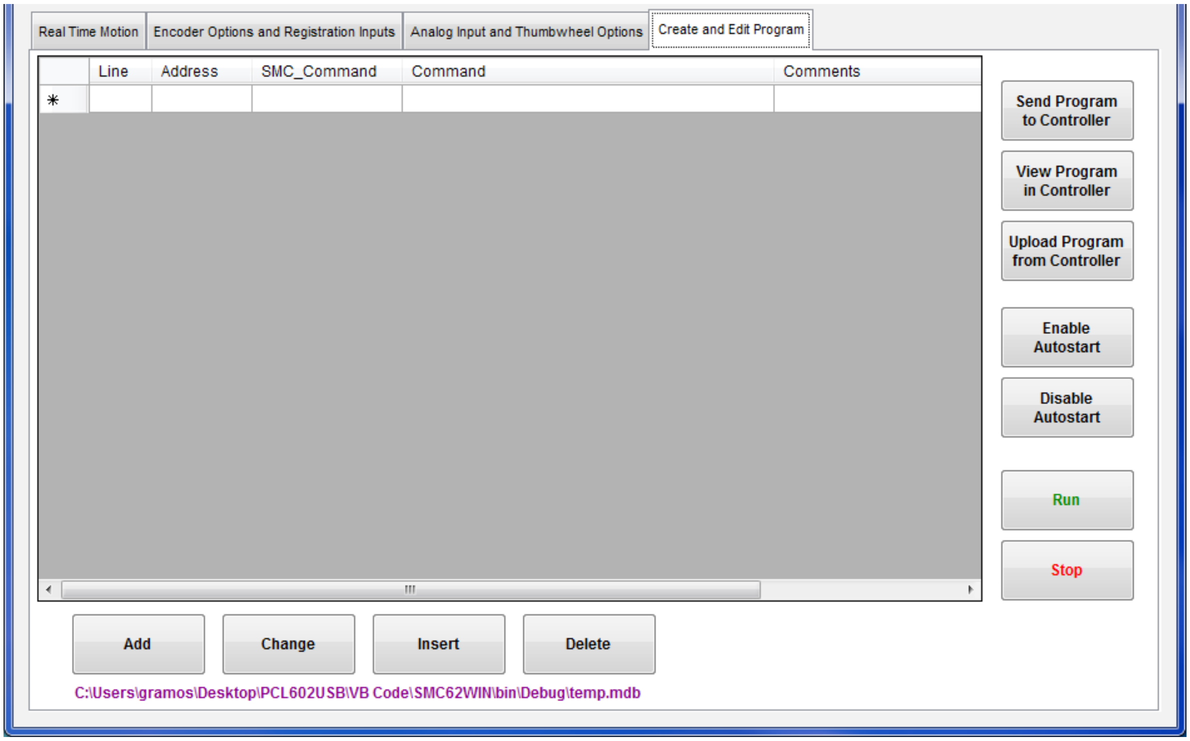
Axis: X
Analog Speed Lower: 500
Analog Speed: Not Enabled
Analog Speed Upper: 1500

Analog Position Lower: 0
Analog Position: Not Enabled
Analog Position Upper: 0

Thumbwheel Index: Not Enabled

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


Real Time Motion
Encoder Options and Registration Inputs
Analog Input and Thumbwheel Options
Create and Edit Program
Line
Address
SMC_Command
Command
Comments
Send Program to Controller
View Program in Controller
Upload Program from Controller
Enable Autostart
Disable Autostart
Run
Stop
Add
Change
Insert
Delete

C:\Users\gramos\Desktop\PCL602USB\VB Code\SMC62WIN\bin\Debug\temp.mdb

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.

PCL602 Memory Available

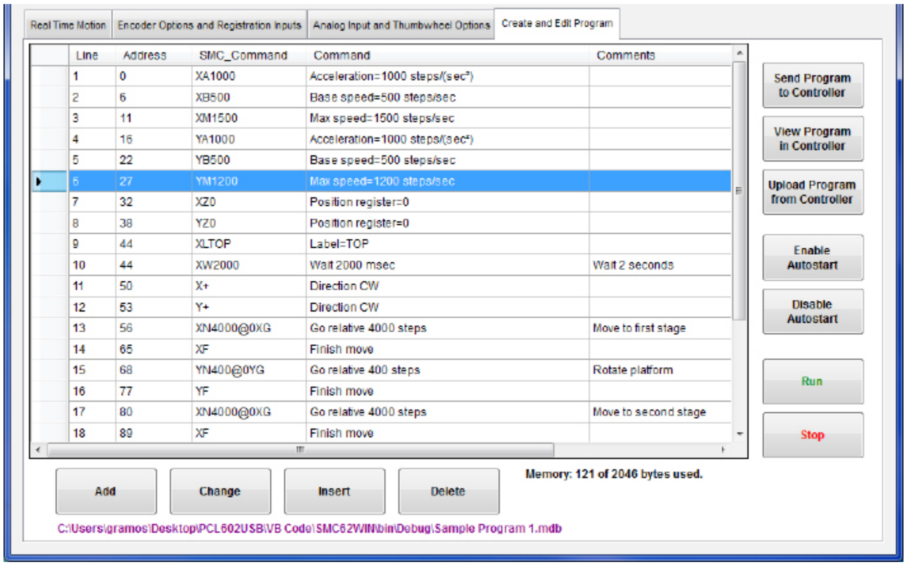
With the create and edit program tab sheet selected, the user can obtain the amount of available memory, located to the right of the Delete command button. The PCL602 has a maximum available memory of 2046 bytes - each instruction can use from 2 to 7 bytes.

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 SMC62WIN window. All programs created by the SMC62WIN 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.


Real Time Motion
Encoder Options and Registration Inputs
Analog Input and Thumbwheel Options
Create and Edit Program
Send Program to Controller
View Program in Controller
Upload Program from Controller
Enable Autostart
Disable Autostart
Run
Stop
Add
Change
Insert
Delete

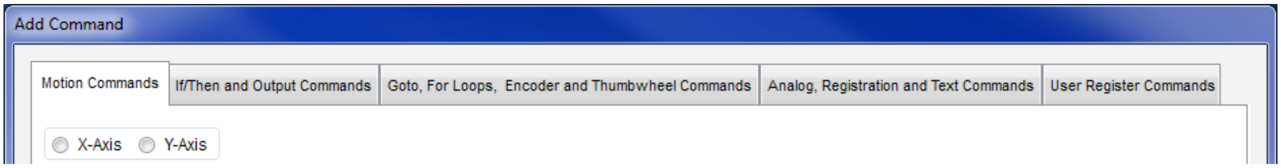
	Line	Address	SMC_Command	Command	Comments
	1	0	XA1000	Acceleration=1000 steps/(sec²)	
	2	6	XB500	Base speed=500 steps/sec	
	3	11	XM1500	Max speed=1500 steps/sec	
	4	16	YA1000	Acceleration=1000 steps/(sec²)	
	5	22	YB500	Base speed=500 steps/sec	
▶	6	27	YM1200	Max speed=1200 steps/sec	
	7	32	XZ0	Position register=0	
	8	38	YZ0	Position register=0	
	9	44	XLTOP	Label=TOP	
	10	44	XW2000	Wait 2000 msec	Wait 2 seconds
	11	50	X+	Direction CW	
	12	53	Y+	Direction CW	
	13	56	XN4000@0XG	Go relative 4000 steps	Move to first stage
	14	65	XF	Finish move	
	15	68	YN400@0YG	Go relative 400 steps	Rotate platform
	16	77	YF	Finish move	
	17	80	XN4000@0XG	Go relative 4000 steps	Move to second stage
	18	89	XF	Finish move	

Memory: 121 of 2046 bytes used.

C:\Users\gramos\Desktop\PCL602USB\VB Code\SMC62WIN\bin\Debug\Sample Program 1.mdb

Add/Change/Insert Commands

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

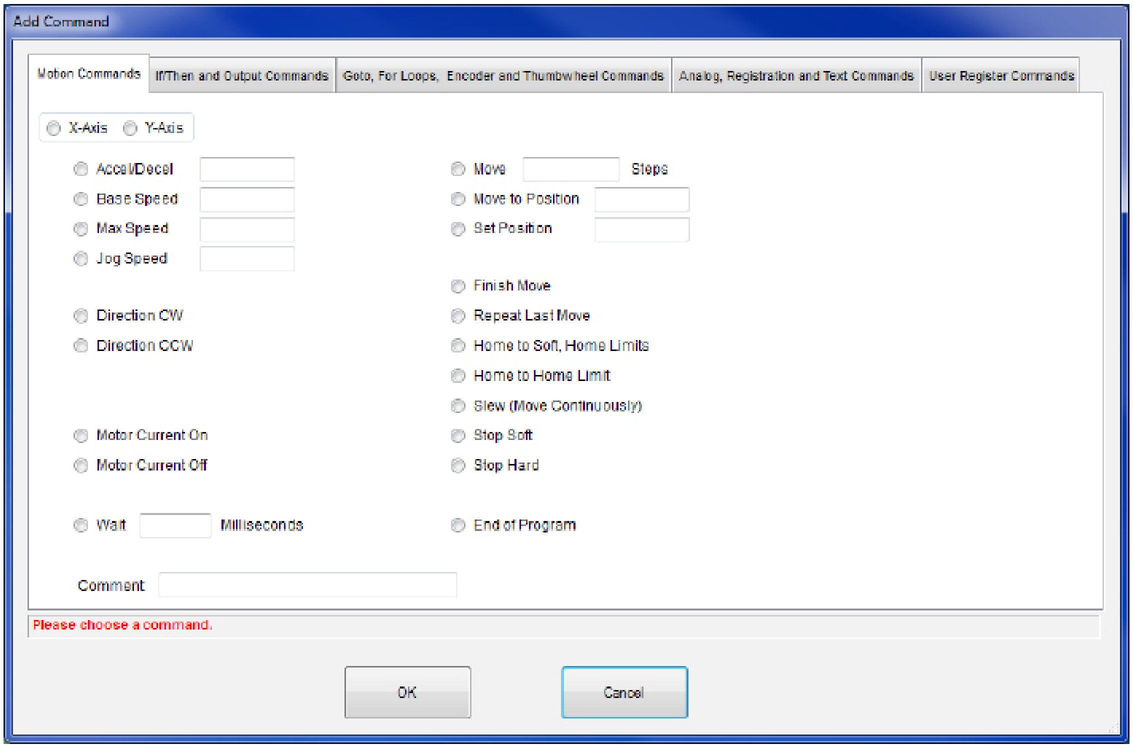

Add Command
Motion Commands
If/Then and Output Commands
Goto, For Loops, Encoder and Thumbwheel Commands
Analog, Registration and Text Commands
User Register Commands
○ X-Axis
○ Y-Axis

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/0 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, Regisration 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.
User Register Commands	Software section that allows the user to load, increment, decrement, and compare user registers that can be used to control the execcution of the program.
X-Axis / Y-Axis	Selects the axis that is to execute the current command being programmed.

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.

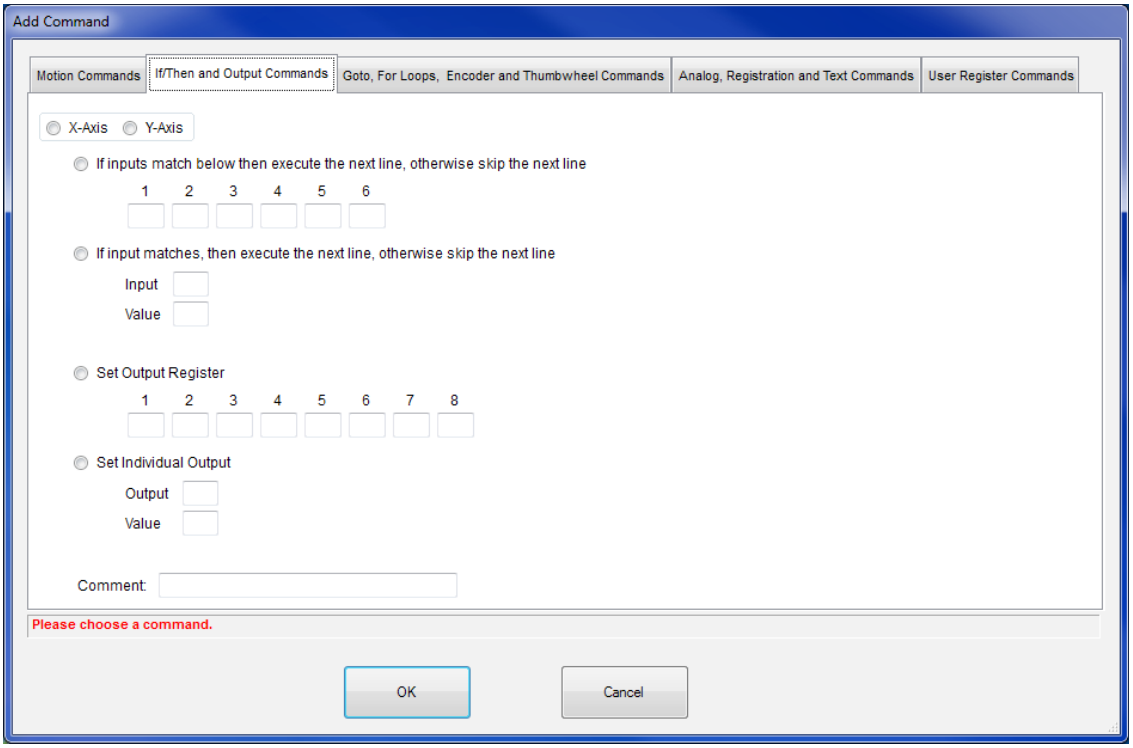
Add Tab Sheets - Motion Commands



Please choose a command.

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.

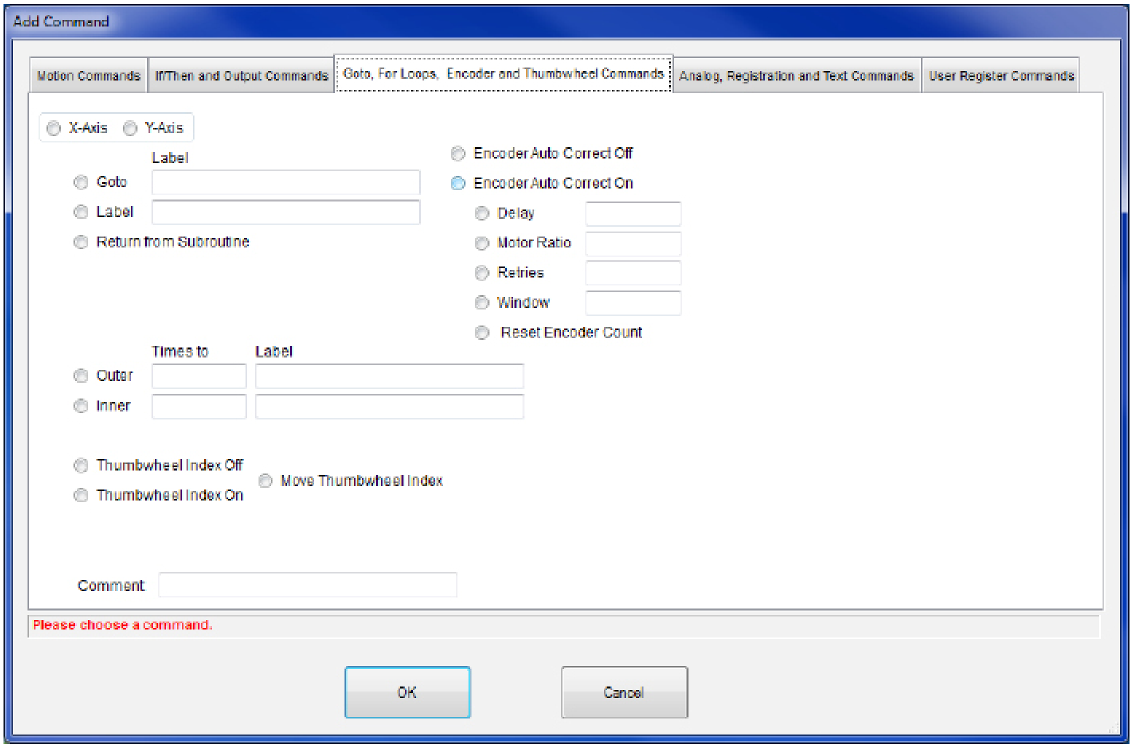
Add Tab Sheets - If/Then and Output Commands


Add Command
Motion Commands
If/Then and Output Commands
Goto, For Loops, Encoder and Thumbwheel Commands
Analog, Registration and Text Commands
User Register Commands
○ X-Axis ○ Y-Axis
○ If inputs match below then execute the next line, otherwise skip the next line
1 2 3 4 5 6
○ If input matches, then execute the next line, otherwise skip the next line
Input
Value
○ Set Output Register
1 2 3 4 5 6 7 8
○ Set Individual Output
Output
Value
Comment:
OK
Cancel

Please choose a command.

If inputs match below then execute the next line, otherwise skip the next line	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.
If input matches, then execute the next line, otherwise skip the next line	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.
Set Outputs	The outputs can be turned (on=1) or (off=0). These outputs can be used to trigger PLC operations, relays, solenoids, etc.

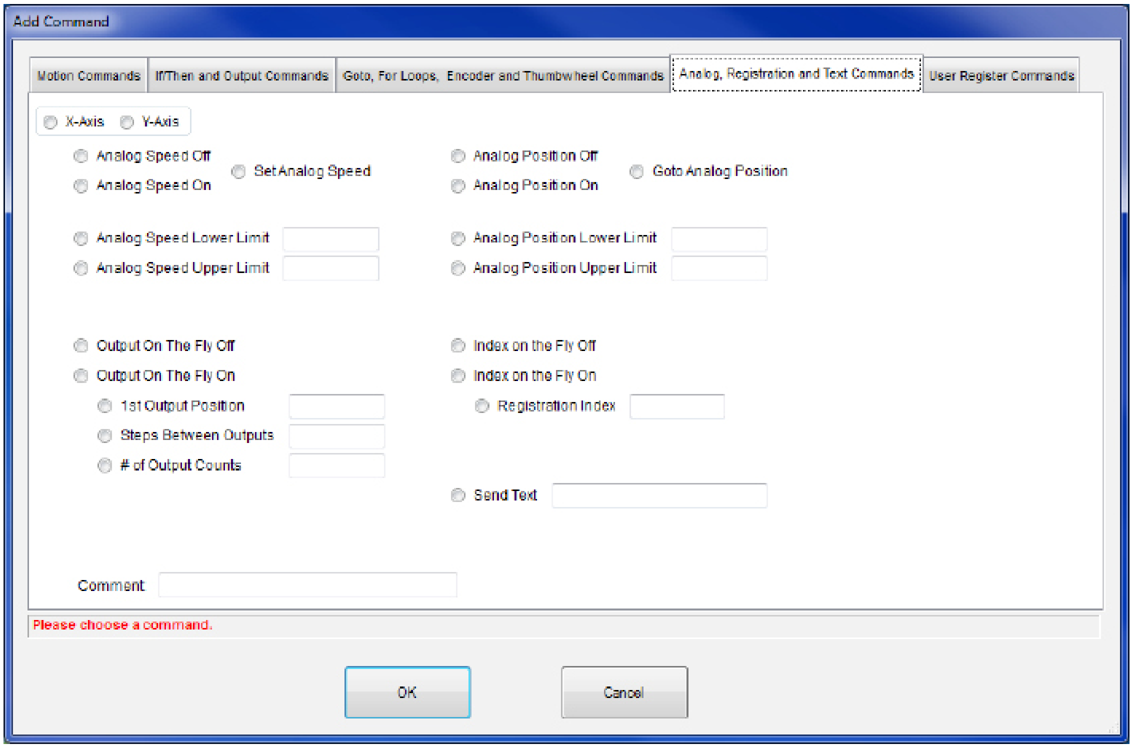
Add Tab Sheets - Goto, For Loops, Encoder and Thumbwheel Commands



Please choose a command.

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).

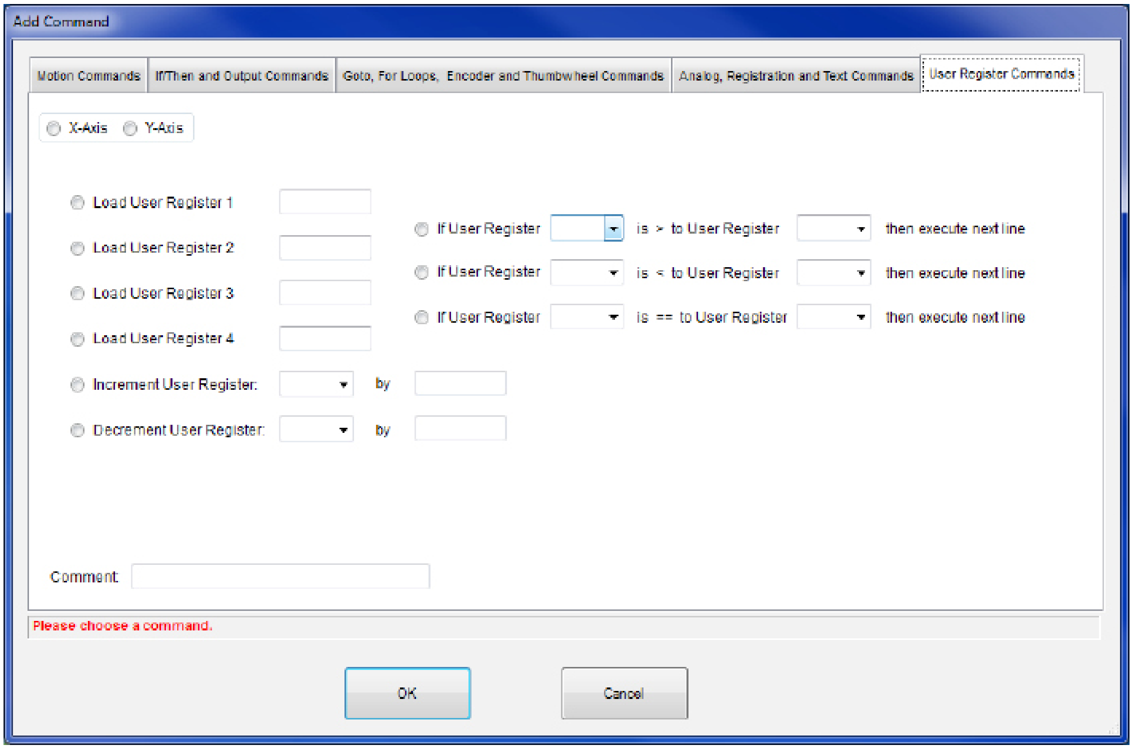
Add Tab Sheets - Analog, Registration and Text Commands



Please choose a command.

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)

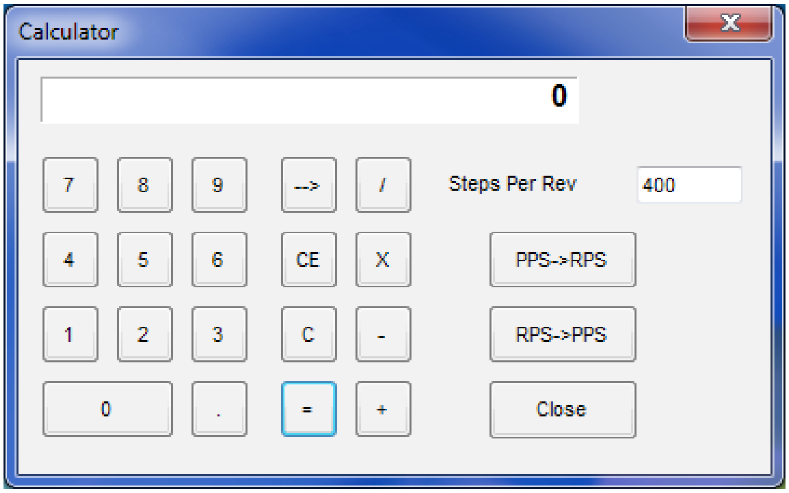
Add Tab Sheets - User Register Commands



Please choose a command.

Load User Register	Loads the given value into the specified user register.
Increment User Register	Increments the specified user register by the given value and stores the new value into that same user register.
Decrement User Register	Decrements the specified user register by the given value and stores the new value into the same user register.
If User Register A > User Register B	If the value in User Register A is greater than the value User Register B, then the next line of code is excuted; otherwise, the next line of code is skipped.
If User Register A < User Register B	If the value in User Register A is less than the value in User Register B, then the next line of code is executed; otherwise, the next line of code is skipped.
If User Register A == User Register B	If the value in User Register A is equal to the value in User Register B, then the next line of code is executed: otherwise, the next line of code is skipped.

Calculator



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 the Calculator.

Section 4: Direct Talk Mode

Direct mode is used to directly control motion for real time movements through serial communication. The PCL602 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 and Axis Selection

In order to select a unit, the @ command followed by the address of the unit must be sent. Then, in order to select the axis, the ‘X’ or ‘Y’ character must follow.
NOTE: There should be no spaces between any of the characters.

How to select a unit and axis:

```
@0X    (Unit 0, X-axis)                    @13X  (Unit 13, X-axis)
@0Y    (Unit 0, Y-axis)                    @13Y  (Unit 13, Y-axis)
```

How to get a response from a unit:

```
@0XS(Carriage Return)
@0XVO(Carriage Return)
```

Note: Some commands such as ‘$’ and ‘!’ are not axis dependent; therefore, the axis character can be omitted. 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:      @0X$(carriage return)
               incorrect:      @ 0 X $(carriage return)
```

Command Summary:

A - Acceleration/Deceleration
B - Base speed
C - Steps between outputs
D - 1st output on the fly position
EA - Encoder autocorrect enabled
ED - Encoder delay
EM - Encoder motor ratio
ER - Encoder retries
ET - Encoder reset
EW - Encoder window
G - Go number of steps
H - Home
I - Read inputs
J- Jog speed

M - Max speed
N - Number of steps
O - Set outputs
P - Absolute position
S - Go slew
T - Motor current enabled
V - Verify
Z - Position
! - Error codes register
$ - Version number
% - Verify axis number
‘ - Index on the fly enabled
(- Output on the fly enabled

+ - Clockwise direction
, - Stop soft
- - Counterclockwise direction
. - Stop hard
/ - Thumbwheel enabled
: - Analog position enabled
; - Analog speed enabled
[- Analog speed lower limit
] - Analog speed upper limit
^ - Number of outputs
{ - Analog position lower limit
} - Analog position upper limit
~ - Set address of PCL602

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 then the maximum speed and jog speed.

Range: 1 - 5,000

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 - 65,535

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 - 65,535

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 - 65,535

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 PCL602 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 PCL602 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 PCL602 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 PCL602 will send clocks until a home limit is reached, ramp down to base speed, change directions and run at base speed unit 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 complemented binary value of the inputs to the PC. The inputs will read as 0 when they are open or high; On the other hand, they will return a 1 when they are grounded. For example; if all inputs are active (grounded), the command will return a 63. If all inputs are inactive (open), the command will return a 0. Input 1 is the LSB and input 6 is the MSB.

Format 2: I[input]

Description: This command returns the binary or logic 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.

Range: 1 - 6

J - Jog speed

Format: J[value]

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

Range: 1 - 50,000

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 - 50,000

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 PCL602 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 - 8,388,607

Format 2: N

Description: This command reads the thumbwheel switches to set the number of clocks for the PCL602 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 PCL602 to send out following a G command. Motion is not activated by this command; it only sets the register. (N = P - Z)

Range: -8,388,607 to +8,388,607

Format 2: P

Description: This command uses the voltage on input 1 to calculate and set the number of clocks for the PCL602 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 PCL602. 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: -8,388,607 to +8,388,607

! - Error codes register

Format: !

Description: This command requests the PCL602 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 PCL602 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 PCL602 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 PCL602 controller will continue to the set position and stop. In a slew type motion the PCL602 controller will ramp down and stop. In a home type motion the PCL602 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. **Note** that when this command is sent without an axis (e.g. “@0.”), the device interprets it as an **emergency stop command**, and, therefore, it stops motion on both axes.

/ - 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 - 50,000

] - 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 - 50,000

^ - 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 - 65,535

} - 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 - 65,535

~ - 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 PCL602 controller.

Range: 0 - 99

Section 5: Troubleshooting

Problem:

Cannot establish communications with the PCL602.

Possible Solutions:

1) Make sure the PCL602 controller has power. Is the Green LED on?
2) Check the USB connections.
3) Check for loose cable connections either on the PCL602 controller or USB 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 PCL602 controller.
8) 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 PCL602 controller.

Possible Solutions:

1) Is the PCL602 controller connected to the appropriate power supply?
2) Check for any blown fuses in line with the PCL602 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 PCL602 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 PCL602 to clear an error.
3) Another way to clear an error is by using either the SMC62WIN software or the direct mode command instructions set.
4) The SMC62WIN 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.

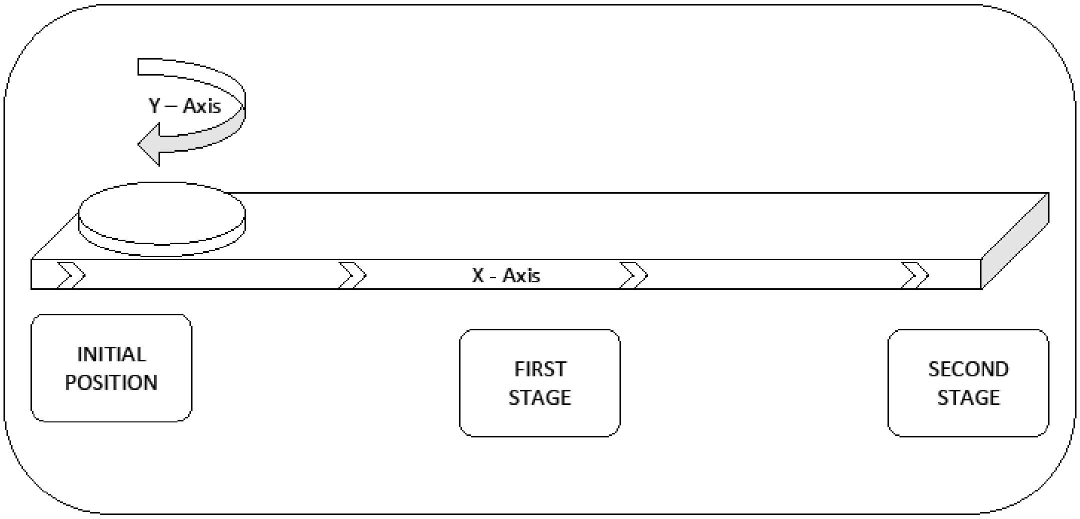
Error Codes

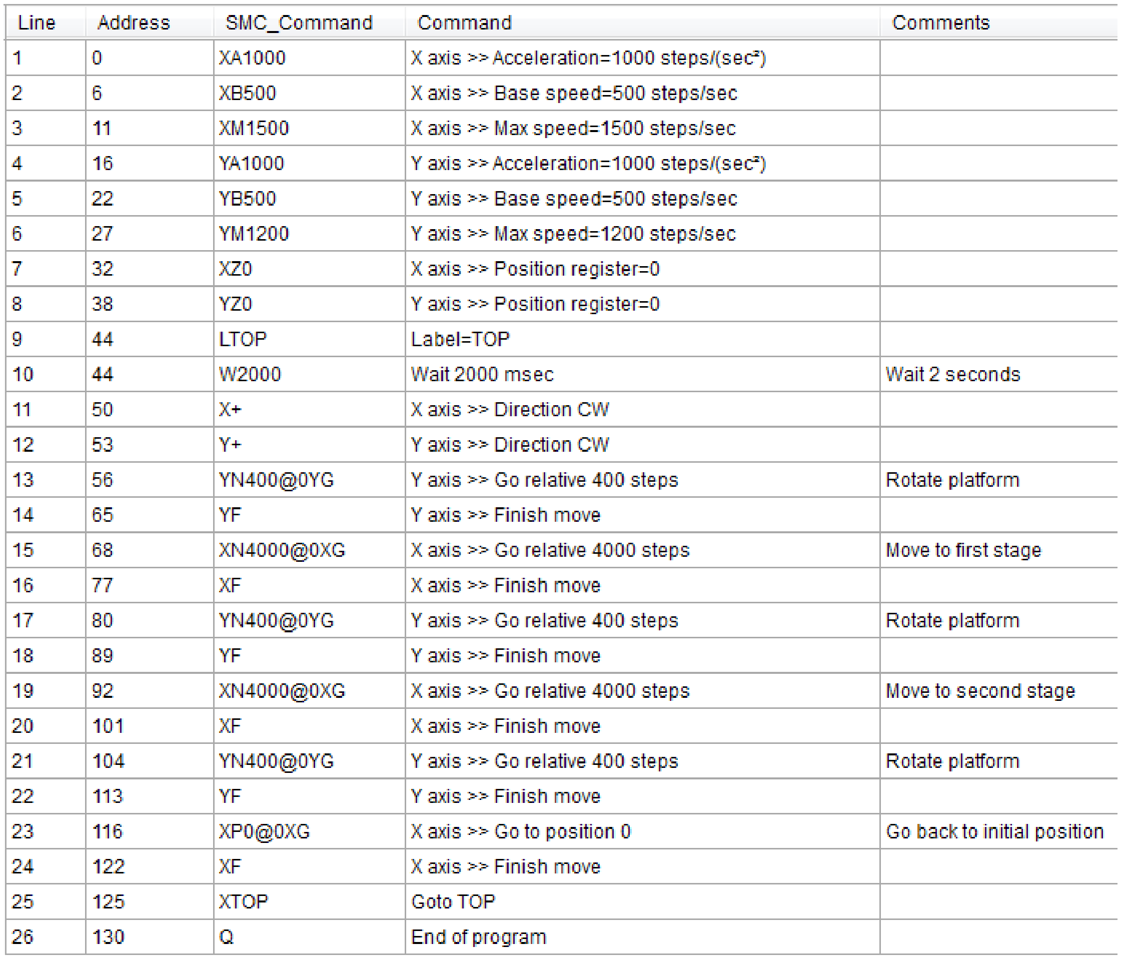
Error Code	Type	Description
1	Receive Overflow Error	The serial communications had a receiving error. This is an internal error caused by the computer.
2	Encoder Error 1	The encoder needed to correct the index.
4	Encoder Error 2	The encoder could not finish the index. Output 8 is set with this error.
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 a rotating platform to three sequential positions. Once at the specified position, the system shall rotate the platform one full revolution, which is equivalent to 400 steps in this sample, at a max speed of 1200 steps/s.

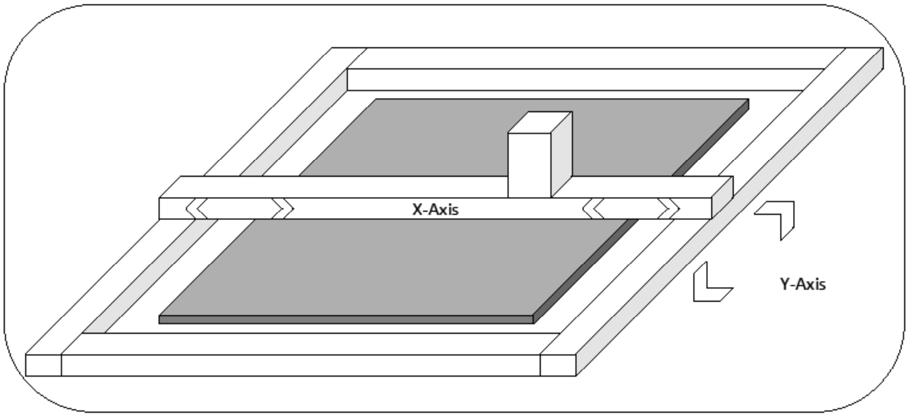


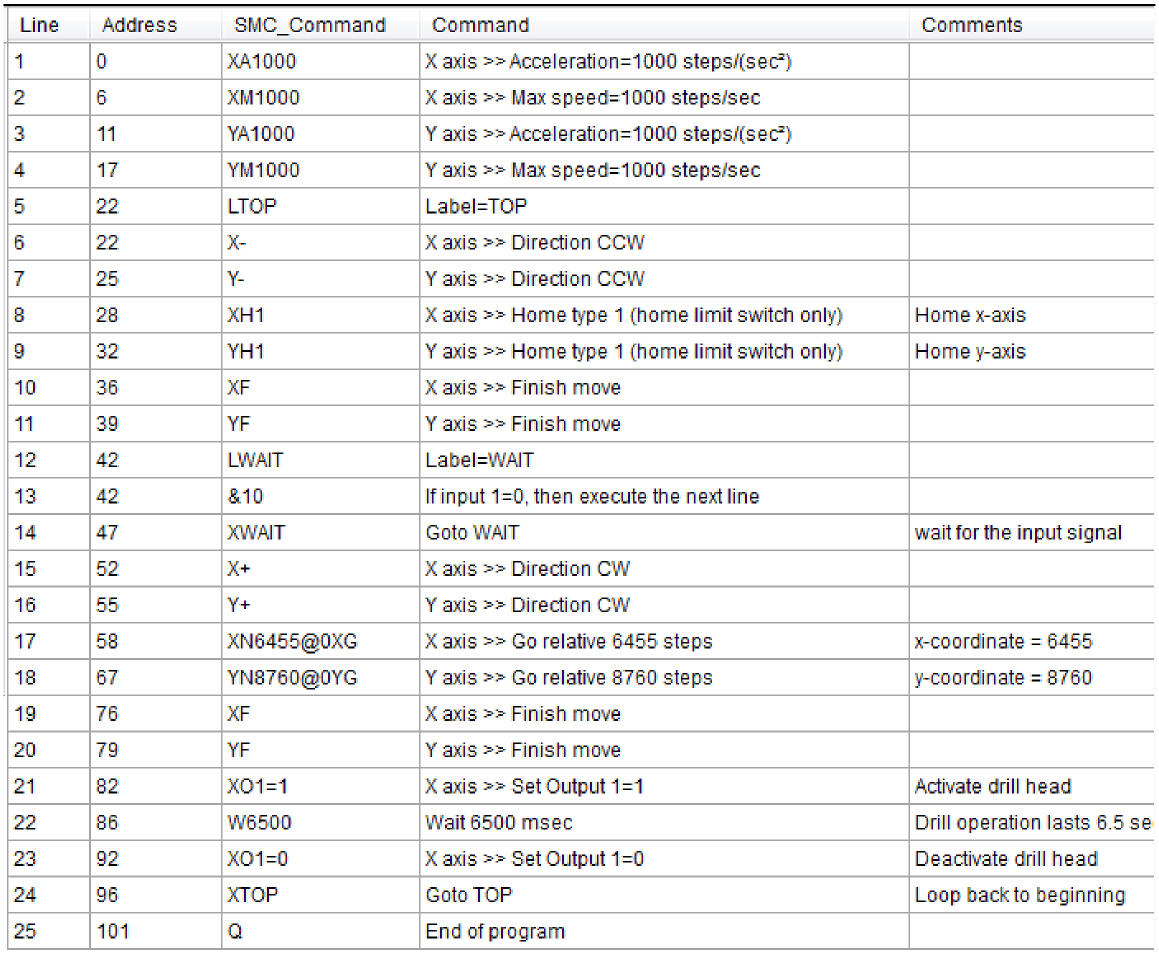


Line	Address	SMC_Command	Command	Comments
1	0	XA1000	X axis >> Acceleration=1000 steps/(sec^2)	
2	6	XB500	X axis >> Base speed=500 steps/sec	
3	11	XM1500	X axis >> Max speed=1500 steps/sec	
4	16	YA1000	Y axis >> Acceleration=1000 steps/(sec^2)	
5	22	YB500	Y axis >> Base speed=500 steps/sec	
6	27	YM1200	Y axis >> Max speed=1200 steps/sec	
7	32	XZ0	X axis >> Position register=0	
8	38	YZ0	Y axis >> Position register=0	
9	44	LTOP	Label=TOP	
10	44	W2000	Wait 2000 msec	Wait 2 seconds
11	50	X+	X axis >> Direction CW	
12	53	Y+	Y axis >> Direction CW	
13	56	YN400@0YG	Y axis >> Go relative 400 steps	Rotate platform
14	65	YF	Y axis >> Finish move	
15	68	XN4000@0XG	X axis >> Go relative 4000 steps	Move to first stage
16	77	XF	X axis >> Finish move	
17	80	YN400@0YG	Y axis >> Go relative 400 steps	Rotate platform
18	89	YF	Y axis >> Finish move	
19	92	XN4000@0XG	X axis >> Go relative 4000 steps	Move to second stage
20	101	XF	X axis >> Finish move	
21	104	YN400@0YG	Y axis >> Go relative 400 steps	Rotate platform
22	113	YF	Y axis >> Finish move	
23	116	XP0@0XG	X axis >> Go to position 0	Go back to initial position
24	122	XF	X axis >> Finish move	
25	125	XTOP	Goto TOP	
26	130	Q	End of program	

Sample Program 2:

Sample Program 2 illustrates the implementation of a two-dimensional motion system with a range of 10,000 steps on either axis. The PCL602 will be programmed to home upon power. After receiving a pulse on one of its inputs, it will go to a position specified in the program where it will activate one of its outputs to let the drill head of the system know that it must perform its operation. Once the drill operation is complete, it will home back to its (0,0) coordinate and wait for a pulse on one of its inputs.

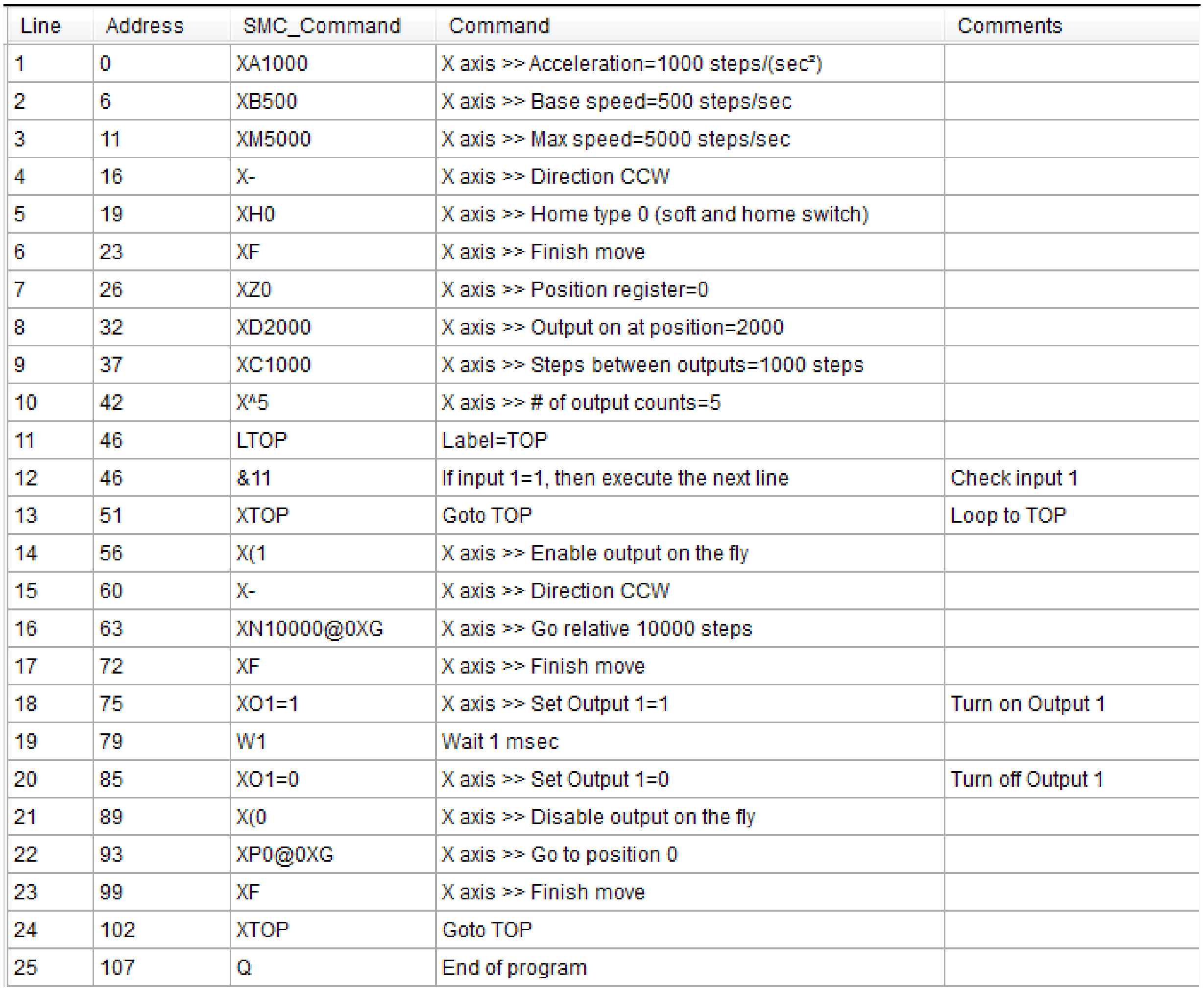




Line	Address	SMC_Command	Command	Comments
1	0	XA1000	X axis >> Acceleration=1000 steps/(sec^2)	
2	6	XM1000	X axis >> Max speed=1000 steps/sec	
3	11	YA1000	Y axis >> Acceleration=1000 steps/(sec^2)	
4	17	YM1000	Y axis >> Max speed=1000 steps/sec	
5	22	LTOP	Label=TOP	
6	22	X-	X axis >> Direction CCW	
7	25	Y-	Y axis >> Direction CCW	
8	28	XH1	X axis >> Home type 1 (home limit switch only)	Home x-axis
9	32	YH1	Y axis >> Home type 1 (home limit switch only)	Home y-axis
10	36	XF	X axis >> Finish move	
11	39	YF	Y axis >> Finish move	
12	42	LWAIT	Label=WAIT	
13	42	&10	If input 1=0, then execute the next line	
14	47	XWAIT	Goto WAIT	wait for the input signal
15	52	X+	X axis >> Direction CW	
16	55	Y+	Y axis >> Direction CW	
17	58	XN6455@0XG	X axis >> Go relative 6455 steps	x-coordinate = 6455
18	67	YN8760@0YG	Y axis >> Go relative 8760 steps	y-coordinate = 8760
19	76	XF	X axis >> Finish move	
20	79	YF	Y axis >> Finish move	
21	82	XO1=1	X axis >> Set Output 1=1	Activate drill head
22	86	W6500	Wait 6500 msec	Drill operation lasts 6.5 se
23	92	XO1=0	X axis >> Set Output 1=0	Deactivate drill head
24	96	XTOP	Goto TOP	Loop back to beginning
25	101	Q	End of program	

Sample Program 3:

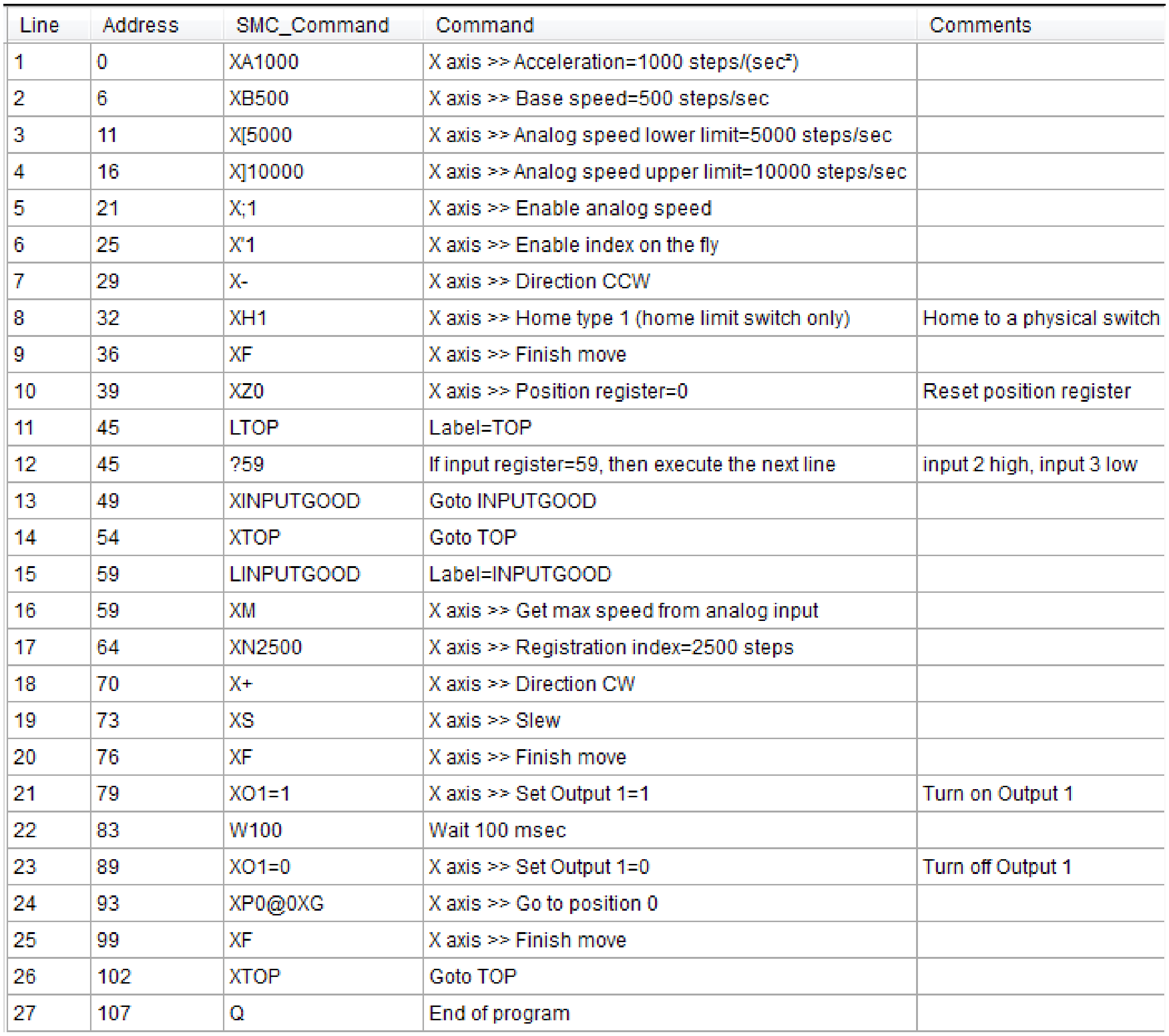
Sample Program 3 illustrates the setup and operation of the output on the fly function, and the use of the if/then statement. In this sample program, the PCL602 is to be used as a single-axis controller with the X-axis being the only axis used. The system is first homed using home type 0, waits for input 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	Address	SMC_Command	Command	Comments
1	0	XA1000	X axis >> Acceleration=1000 steps/(sec^2)	
2	6	XB500	X axis >> Base speed=500 steps/sec	
3	11	XM5000	X axis >> Max speed=5000 steps/sec	
4	16	X-	X axis >> Direction CCW	
5	19	XH0	X axis >> Home type 0 (soft and home switch)	
6	23	XF	X axis >> Finish move	
7	26	XZ0	X axis >> Position register=0	
8	32	XD2000	X axis >> Output on at position=2000	
9	37	XC1000	X axis >> Steps between outputs=1000 steps	
10	42	X^5	X axis >> # of output counts=5	
11	46	LTOP	Label=TOP	
12	46	&11	If input 1=1, then execute the next line	Check input 1
13	51	XTOP	Goto TOP	Loop to TOP
14	56	X(1	X axis >> Enable output on the fly	
15	60	X-	X axis >> Direction CCW	
16	63	XN10000@0XG	X axis >> Go relative 10000 steps	
17	72	XF	X axis >> Finish move	
18	75	XO1=1	X axis >> Set Output 1=1	Turn on Output 1
19	79	W1	Wait 1 msec	
20	85	XO1=0	X axis >> Set Output 1=0	Turn off Output 1
21	89	X(0	X axis >> Disable output on the fly	
22	93	XP0@0XG	X axis >> Go to position 0	
23	99	XF	X axis >> Finish move	
24	102	XTOP	Goto TOP	
25	107	Q	End of program	

Sample Program 4:

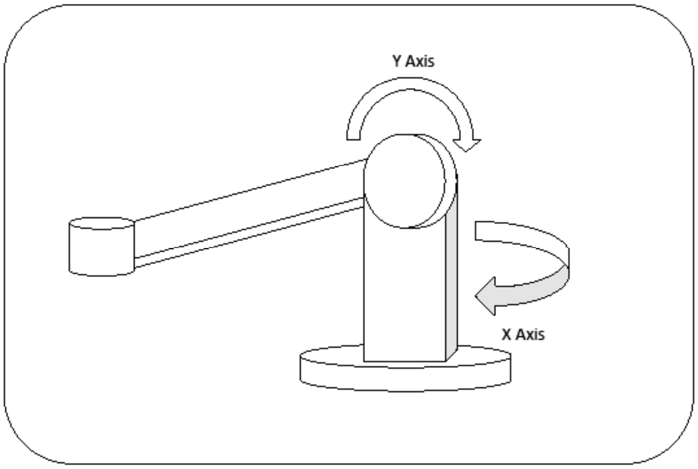
Sample Program 4 illustrates the setup of the analog speed function and the use of “indexing-on-the-fly.” In this sample program, only the X-axis in the PCL602 is used. The system is first homed using home type 1. The next step is to wait for the input register to read 110111 (input 2 must be high while 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.

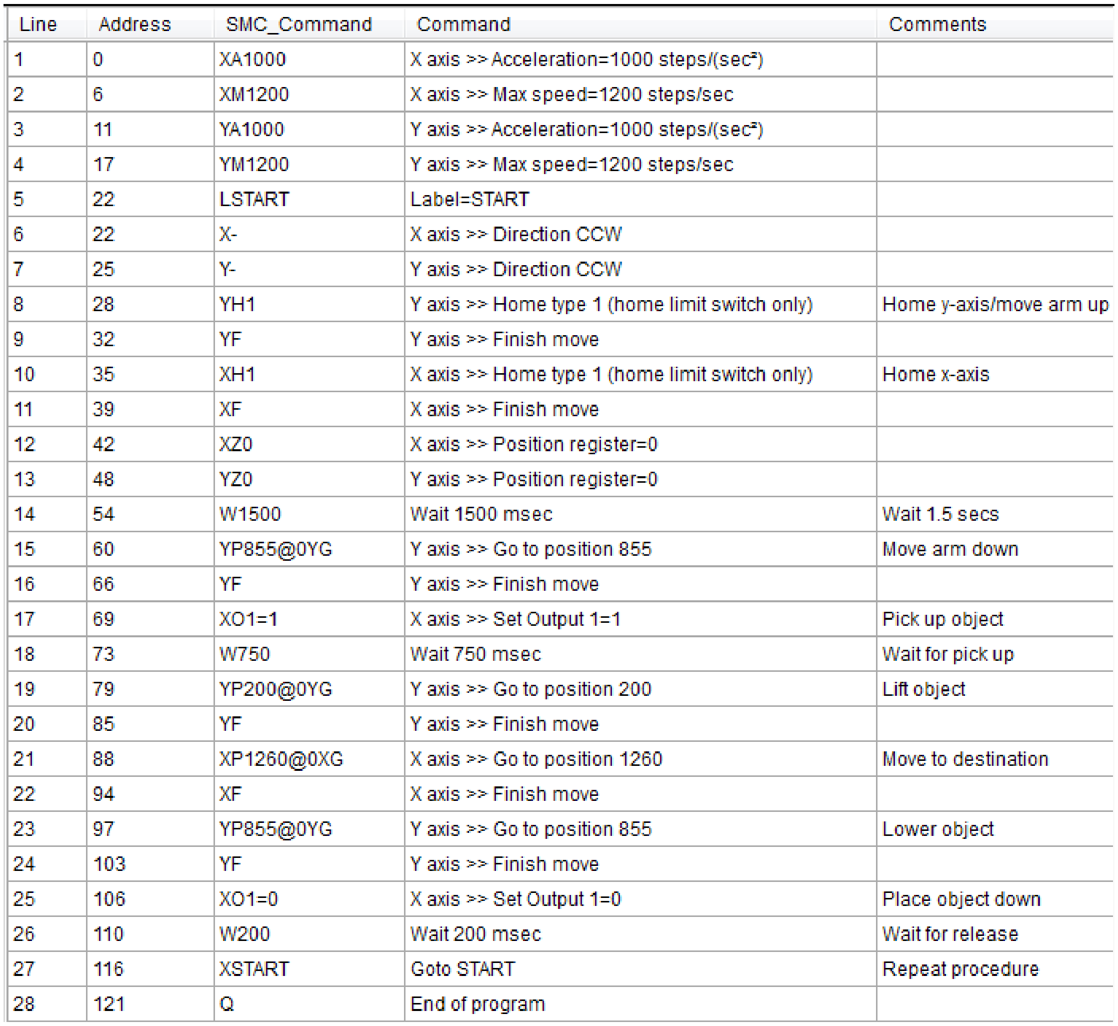


Line	Address	SMC_Command	Command	Comments
1	0	XA1000	X axis >> Acceleration=1000 steps/(sec²)	
2	6	XB500	X axis >> Base speed=500 steps/sec	
3	11	X[5000	X axis >> Analog speed lower limit=5000 steps/sec	
4	16	X]10000	X axis >> Analog speed upper limit=10000 steps/sec	
5	21	X;1	X axis >> Enable analog speed	
6	25	X'1	X axis >> Enable index on the fly	
7	29	X-	X axis >> Direction CCW	
8	32	XH1	X axis >> Home type 1 (home limit switch only)	Home to a physical switch
9	36	XF	X axis >> Finish move	
10	39	XZ0	X axis >> Position register=0	Reset position register
11	45	LTOP	Label=TOP	
12	45	?59	If input register=59, then execute the next line	input 2 high, input 3 low
13	49	XINPUTGOOD	Goto INPUTGOOD	
14	54	XTOP	Goto TOP	
15	59	LINPUTGOOD	Label=INPUTGOOD	
16	59	XM	X axis >> Get max speed from analog input	
17	64	XN2500	X axis >> Registration index=2500 steps	
18	70	X+	X axis >> Direction CW	
19	73	XS	X axis >> Slew	
20	76	XF	X axis >> Finish move	
21	79	XO1=1	X axis >> Set Output 1=1	Turn on Output 1
22	83	W100	Wait 100 msec	
23	89	XO1=0	X axis >> Set Output 1=0	Turn off Output 1
24	93	XP0@0XG	X axis >> Go to position 0	
25	99	XF	X axis >> Finish move	
26	102	XTOP	Goto TOP	
27	107	Q	End of program	

Sample Program 5:

Sample Program 5 illustrates a dual-axis application using the PCL602 to control a pick and place arm. The PCL602 will first take both axes to their respective origins by using home switches. Then the controller will move the arm to the pick-up position, and then it will set one of its outputs to activate the pick-up mechanism at the end of the arm. The controller will lift the arm and move it to the desired location. The output will be set to zero to deactivate the pick-up mechanism, thus placing the item at that location.





Line	Address	SMC_Command	Command	Comments
1	0	XA1000	X axis >> Acceleration=1000 steps/(sec^2)	
2	6	XM1200	X axis >> Max speed=1200 steps/sec	
3	11	YA1000	Y axis >> Acceleration=1000 steps/(sec^2)	
4	17	YM1200	Y axis >> Max speed=1200 steps/sec	
5	22	LSTART	Label=START	
6	22	X-	X axis >> Direction CCW	
7	25	Y-	Y axis >> Direction CCW	
8	28	YH1	Y axis >> Home type 1 (home limit switch only)	Home y-axis/move arm up
9	32	YF	Y axis >> Finish move	
10	35	XH1	X axis >> Home type 1 (home limit switch only)	Home x-axis
11	39	XF	X axis >> Finish move	
12	42	XZ0	X axis >> Position register=0	
13	48	YZ0	Y axis >> Position register=0	
14	54	W1500	Wait 1500 msec	Wait 1.5 secs
15	60	YP855@0YG	Y axis >> Go to position 855	Move arm down
16	66	YF	Y axis >> Finish move	
17	69	XO1=1	X axis >> Set Output 1=1	Pick up object
18	73	W750	Wait 750 msec	Wait for pick up
19	79	YP200@0YG	Y axis >> Go to position 200	Lift object
20	85	YF	Y axis >> Finish move	
21	88	XP1260@0XG	X axis >> Go to position 1260	Move to destination
22	94	XF	X axis >> Finish move	
23	97	YP855@0YG	Y axis >> Go to position 855	Lower object
24	103	YF	Y axis >> Finish move	
25	106	XO1=0	X axis >> Set Output 1=0	Place object down
26	110	W200	Wait 200 msec	Wait for release
27	116	XSTART	Goto START	Repeat procedure
28	121	Q	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.

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, 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.