MANUAL FOR MODELS

SMC40 SERIES INDEXER

December 3, 1999 *#L010098*

.

TABLE OF CONTENTS

SECTION 1 - INTRODUCTION

This manual is intended to help the user apply the SMC40 family of products in motion control applications. Familiarity with computers, programmable logic controllers (PLC's), or terminals is helpful, but is not essential.

The SMC40 family of products is based on Anaheim Automation's advanced step motor controller integrated circuit. This high speed microprocessor uses a simple language that can access over 125 different commands. These commands deal with motion parameters, encoders, inputs and outputs, variables, math functions, and much more. This functionality produces a very powerful step motor indexer that is able to handle even the most complicated machine control. Expansion Modules have been added for this series to include easy use of external thumbwheel modules, external terminals, and additional inputs.

The SMC40 has two modes of operation: Direct Mode and Stored Program Mode. In Stored Program Mode a whole program is stored in the SMC40's internal memory (16,000 bytes). Once a program has been sent to memory, it is non-volatile meaning if the power is turned off, the program will still remain in the memory. In Direct Mode, commands are sent to the unit via a serial port and are executed one command at a time.

A step motor is essentially a digital device. If you give the step motor driver 10 clock pulses the motor moves 10 steps. Sometimes a closed-loop system is needed to verify that the motor indeed moved 10 steps. The SMC40 will accept encoder inputs to form a closed-loop system.

The SMC40 is designed to communicate over a RS232C or RS422 bi-directional serial data bus. The RS422 serial bus is better suited for industrial environments susceptible to electrically noisy conditions. RS422 can reliably travel to a distance of 4000 feet. The RS232C line can only be used to a distance of 50 feet in a noise free environment.

Almost all computers have, or can be equipped with, an RS232 port. If you wish to send your RS232C signal over 50 feet, Anaheim Automation manufactures a RS232C to RS422 Data Converter (Model DC1709).

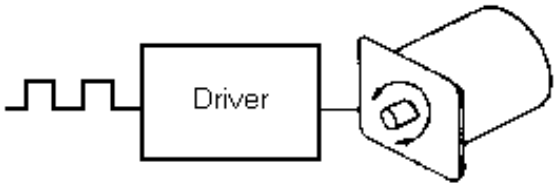
The SMC40 provides programming of acceleration, base speed (start up speed), running speed, and the number of steps to be taken in both relative and absolute positioning modes. On absolute positioning moves, the SMC40 automatically determines the proper direction to go and the number of steps to take. The relative positioning will move a number of steps in the direction that the user defines. The SMC40 has a high level command set including: looping, conditional statements, time delays, power down motor, encoder feedback, maskable I/O and open-collector (NPN) outputs.

Hard, Soft, and Home Limit Switch inputs are provided for each axis. These features are generally required in most machine control designs. 96 Inputs and 24 Outputs are provided per unit. These I/O may be used for monitoring and controlling machine operation and/or interaxis coordination. These I/O are accessible independent of the busy state of the axis controls. The Inputs are TTL/CMOS compatible. The Outputs are current sinking, open collector darlingtons.

The SMC40 has a built-in programmable reset circuit so that all axes in the daisychain may be reset. The outputs are reset to the off state when the board is reset. Reset is automatic on power-up or with a "break" signal on the RS232 or RS422 input. Windows software is provided when you purchase the unit. This software allows you to write and change programs that are to be stored in the SMC40 for autostart use. The software also allows you to save the programs onto your computer disk, and easily retrieve them when needed. The program can upload the stored program from the SMC40, allow you to make changes, and then download the program back to the SMC40.

STEP MOTOR DRIVER

A step motor driver is a device that takes input signals (usually Clock and Direction) and translates this information into phase currents in the motor. Each time the step motor driver receives a pulse, the step motor moves one step. If the driver receives 200 pulses, the motor moves 200 steps. The motor steps at the same frequency as the clock pulses.



TECHNICAL SUPPORT

If you have problems using any of the equipment covered by this manual, please read this manual completely to see if that will answer the questions you have. Be sure to look in the TROUBLESHOOTING section on page 51 of this manual. If you need assistance beyond what this manual can provide, contact your Local Distributor where you purchased the unit or the factory direct. Be sure to have this manual for reference. It is helpful to have the unit connected to a computer with the appropriate software loaded.

ORDERING INFORMATION

The table below lists a variety of SMC40 products available from Anaheim Automation. These products include those covered by this manual along supporting cables and devices. We are continually adding new products to our line, so please consult your nearest Authorized Anaheim Automation Distributor or Representative for information on the latest releases. We can also be found on the web, at: **http://www.anaheimautomation.com**

MODEL NUMBER	DESCRIPTION	
DPD72401	Single-Axis Intelligent Driver Pack,	1 - 7 Amp
DPD60401	Single-Axis Intelligent Microstep Driver Pack,	1 - 6 Amp
DPK72402	Dual-Axis Intelligent Driver Pack,	1 - 7 Amp
DPK72403	Triple-Axis Intelligent Driver Pack,	1 - 7 Amp
PCL401	Single-Axis Intelligent Standalone Controller	
PCL402	Dual-Axis Intelligent Standalone Controller	
PCL403	Triple-Axis Intelligent Standalone Controller	
SMC40M-24I	24 Input Expansion Module	
SMC40M-TWS7	Thumbwheel Expansion Module	
AA2M50	50 Pin Breakout Terminal Board	
AA2682	50 Pin Ribbon Cable Assembly, 2 Feet	
AA9FMC-6	Serial Cable, 6 Feet	
SMTNR2-1	Remote Panel Mount Programmer/Terminal	
TT1R2-1	Handheld Programmer/Terminal	

TABLE 1: Ordering Information

SECTION 2 - QUICK START

You will need:

Driver Pack (Single Axis)
Compatible Step Motor
Serial Cable
Computer with Windows 3.1 (or above)
Intelligent Indexer Software Package
Switch
5VDC LED or Equivalent (Optional)

10 QUICK STEPS – TUTORIAL PROGRAM

STEP 1 - Install the Software

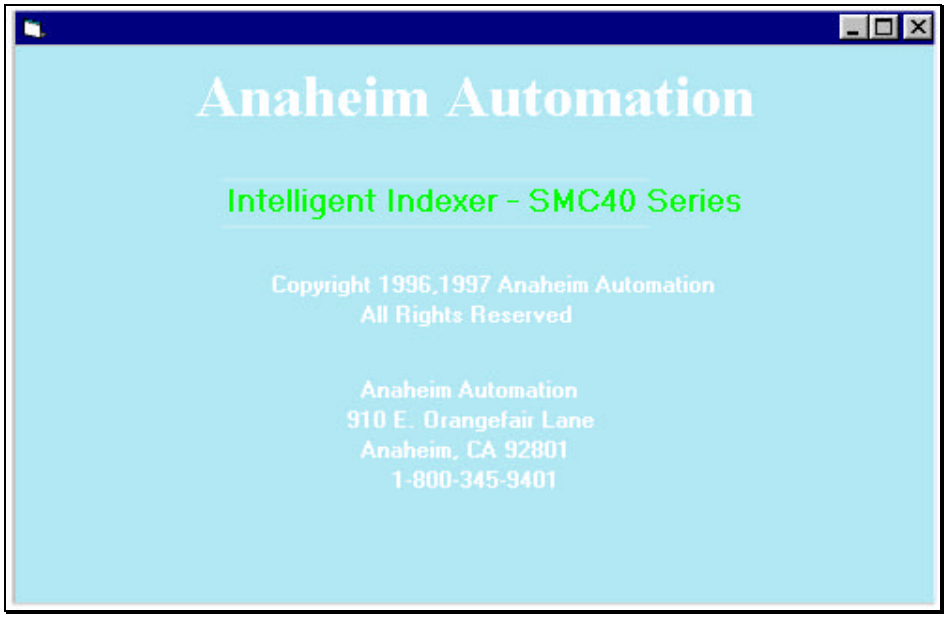
To install the SMC40 Software refer to Section 7 - Intelligent Software

STEP 2 - Setup the Software

Run the Intelligent Indexer Software by Double Clicking the **SMC40** Icon in Windows.

The Defaults are:

Com Port	Com1
Baud Rate	9600
Units	Steps
Number of Axes	1



STEP3 - Connect a Motor

Connect the step motor. Use Section 3, Motor Hook Up to assist. Set the Kick Current to match the motors rated phase current. The kick current potentiometer is located on the driver side labeled Kick Current Adjust. **Whenever you are connecting wires to the Driver Pack, be certain that the power is off or significant damage can occur**.

STEP 4 - Connect the Switch and (Optional) LED

This switch is used as a "start switch" to cause the Driver Pack to begin sequencing through the program. Connect one side of the switch to Pin #1 (Input #1) and the other side of the switch to Pin #5 (0 VDC).

The LED is optional. The LED demonstrates how an Output can be turned on and off. Connect the Anode side of the LED to Pin # 11 (+5 VDC) and the Cathode side of the LED to Pin #13. Note: A resistor must be placed in series with the LED to limit the current. A +5V LED has the series resistor internally connected.

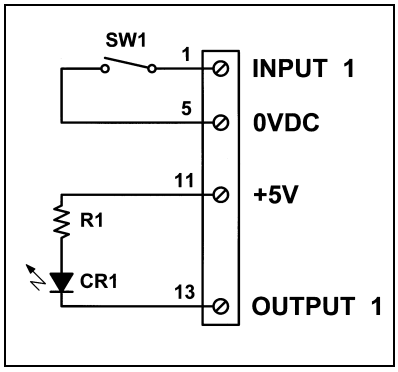


Figure 1: Connect the Switch and LED

Pin #	Description	Pin #	Description
1	Input #1	10	0 VDC
2	Input #2	11	+5 VDC
3	Input #3	12	Clamp
4	Input #4	13	Output #1
5	0 VDC	14	Output #2
6	Input #5	15	Output #3
7	Input #6	16	Output #4
8	Input #7	17	0 VDC
9	Input #8		

Table 3: Connector TB3 Pinout

STEP 5 - Connect the Serial Cable to the Driver Pack

To connect your computer to the Driver Pack you need to connect the serial ports (usually COM1 or COM2) to the units lower DB9 connector. This connector is labeled *RS232/RS422 IN* (P1). It is helpful to know which communications port (Com Port) is in use when we setup the software. In most computers, a 9 pin Male to Female cable will work.

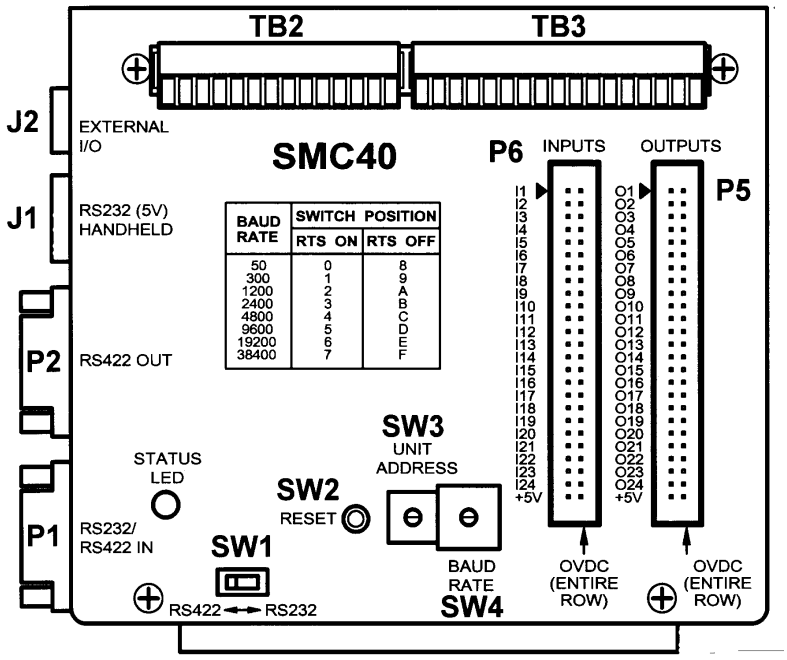
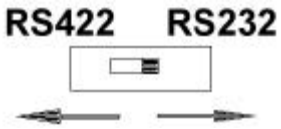


Figure 2: SMC40 Indexer

STEP 6a - Set up the Communication Slide Switch to RS232



Refer to Figure 2 (silkscreen) on the side of the Driver Pack for the slide switch placement

STEP 6b - Set up Unit Number

Refer to Figure 2 or silkscreen on the side of the Driver Pack for the rotary switch setting (SW3). Note the Default Unit Address 0 corresponds to Axis X.

STEP 6c - Set up the Baud Rate

To locate the Baud Rate Switch (SW4) refer to Figure 2. It is the silkscreened table on the Driver Pack. Note: the default is 9600 (position 5) on the Baud Rate Switch (SW4). Refer to Section 5, page 23 for further assistance when selecting the Baud Rate.

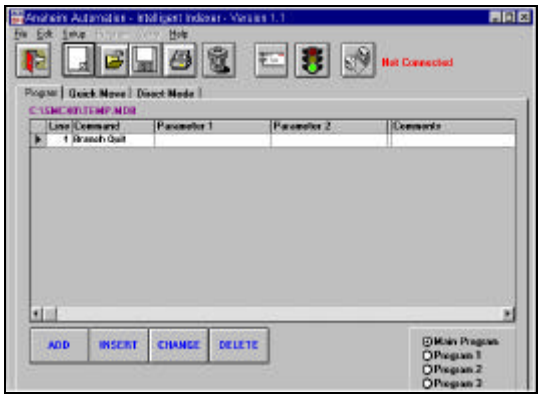
STEP 6d - Power-up the Driver Pack

Plug in the Driver Pack. Standard models DPD/DPF7240_ Series, DPFEN403, DPF11401 and DPD60401 are supplied with a detachable linecord which is plugged into a standard wall socket (115VAC, 60 Hz). The above models are available with a X250 suffix. These units are configured to accept a wide variety of voltages. You must configure the terminal block to handle the desired voltage level. (Refer to Section 4, page 26 for X250 Version).

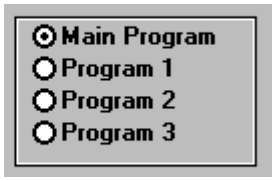
STEP 6e - Press the Connect Icon Button

If everything is connected properly, a green "Connected" indicator will be displayed on the top right hand of the screen.

STEP 7 - Write a Program

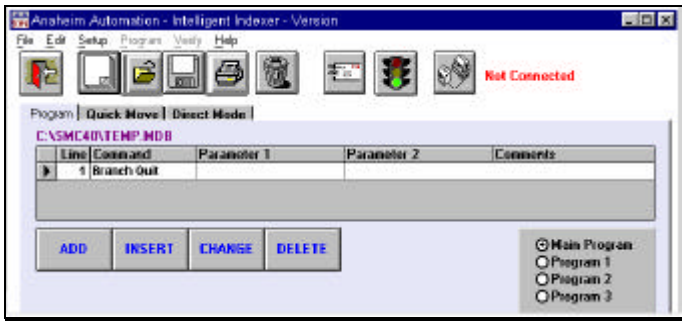


The Software automatically comes up with the file SMC*40.MDB* selected. This will be a blank program. When you save the program you will need to give it another name. We will save our program as TEST1.



There are four program areas that your program may reside in. You will be using the Main Program Area for this test program.

To begin programming place the cursor on the line you wish to enter the command.

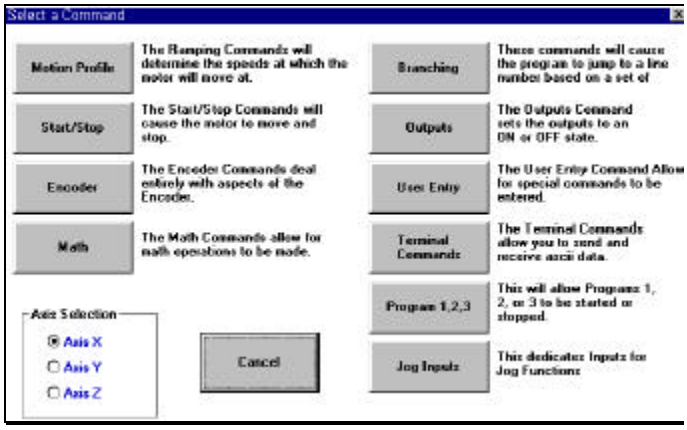


	Line	Command	Parameter 1
▸	1		

Select the Add button by clicking it once on the main programming screen.

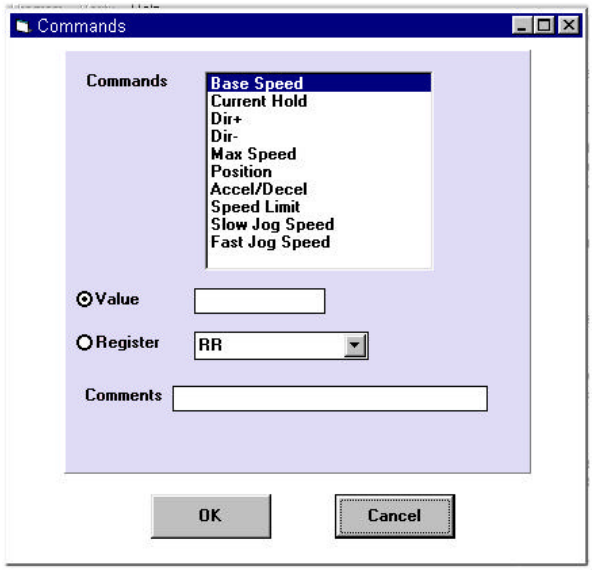
This will take you into the select command programming screen.

Select the **Motion Profile** Button.



Select the highlighted Base Speed and enter 500 for the Value. Click OK.

This will take you back out to the main programming screen. You will see that line 1 now reads Base Speed 500 on Line 1.



	Line	Command	Parameter 1
	1	Base Speed	500
▸	2		

Repeat the same process when entering the rest of the commands.

Turn to the next page to continue the test program.

Note: For Single-Axis units, an X will precede many of the commands.

The following is an overview of the program you will write:
The initialization of the parameters will be set up first. Base Speed will be set to 500 steps/sec, Max Speed to 1000 steps/sec, and Ramping (Accel / Decel) to 10000 steps/sec. Next, you will wait for Input #1 go to Low (0VDC). While waiting or pausing for the input to go low (0VDC), the motor current will be turned off. When Input #1 goes to 0VDC, motion will start. The motor will move in the Clockwise direction 400 steps. Output #1 will be set to 0VDC (*Refer to Section 8 on Command Descriptions for further assistance*), a wait delay of one second will occur and the output will be reset back OFF (+5VDC). This procedure will be looped four times. The program will repeat if Input #1 goes to 0VDC. *Place Branch Quit commands on all Four Programs, even if they are not utilized.*

	Line	**Location of Command**	Command	Value to be Entered	Parameter 1	Parameter 2
▸	1	Motion Profile	XBase Speed	Enter Value of 500	500	
	2	Motion Profile	XMax Speed	Enter Value of 1000	1000	
	3	Motion Profile	XAccel/Decel	Enter Value of 10000	10000	
	4	Branching \ Label	Label	Enter Label as *Pause*	Pause	
	5	Motion Profile	XCurrent Hold	Enter Value of 0	0	
	6	Branching \ If(bit)Then	If I1 = 0	Then Branch to Label *Start Motion* Else Then Branch to Label *Pause*	Then Start Motion	Else Pause
	7	Branching \ Label	Label	Start Motion	Start Motion	
	8	Branching \ For Loop \ Loop Top	Loop Top Start Motion	Click Loop Top Enter on Loop Top Label *Start Motion*. Enter 4 in Number of Times.	4 Times	
	9	Motion Profile	XDir+	Choose Dir+		
	10	Start/Stop	XGo Relative	Choose Go Relative	400	
	11	Outputs	Set Outputs	Click OUT1 ON	ON:1	OFF:
	12	Branching \ Wait Delay	Wait Delay	Enter Value of 1000	1000	
	13	Output	Set Outputs	Click OUT1 OFF	ON:	OFF:1
	14	Branching \ For Loop \ Loop Bottom	Loop Bottom Start Motion	Click Loop Bottom. Enter on Loop Bottom Label *Start Motion*.		
	15	Branching \ Goto \ Label	Goto	Enter Label as *Pause*	Pause	
	16	Branching \ Quit	Branch Quit	Click OK		

Refer to the *Command* Column in the program table above for programming assistance. Note that the *Location of Command* Column will help locate the commands needed. Note that Command Locations are in levels for example step1\ step2\ step3. The *Parameters* will be the **values** and **labels** necessary to complete the test program.

Note: If your program has problems and is showing errors:
1. Save your program by giving it a name.(Example: Test1)
2. Power down the Indexer momentarily, then Power Up.
3. *Reset* the Indexer at the Menu heading under *Setup*
4. Click the Connect Button (Upper Right most button).
5. Programs are Auto Saved so that you may resume where you left off.

STEP 8 - Download the Program (the Envelope Icon Button)

Select the Send Button (Envelope Icon Button). This sends all 4 programs (Main, Programs 1, 2, & 3 to the Driver Pack. The programs will reside in the nonvolatile memory, and will stay there until they are overwritten, or deleted by the software. Once the program is sent, Click OK on the prompt button Program Sent. *(Note: Place a Branch Quit Statement at the end of every program before sending.)*

STEP 9 - Run the Program

To run the program just sent, select the *Start Button* (Traffic Signal Icon Button) which signals Green for GO.

STEP 10 - Switch Closure

After the sample program is written, you will need to close the switch on Input 1 to view the programmed motion. The current to the step motor will be turned off whenever it is motionless. Once the switch has been closed, the motor will move 400 steps, then an output will be activated for 1 second. This will happen 4 times automatically, and then the indexer will wait for a switch closure before repeating the same cycle.

SECTION 3 - STEP MOTOR DRIVER

MODEL – DPD72401, DPK72402, & DPK72403 (BLD72 Driver)

BILEVEL DRIVE

The basic function of a step motor driver is to provide the rated motor phase current to the motor windings in the shortest possible time. The bilevel driver uses a high voltage to get a rapid rate of current rise in the motor windings in the least amount of time. When reaching the preset trip current, the driver turns off the high voltage and sustains the current from the low voltage supply.

HALF-STEP/FULL-STEP

Users have a choice of full-step operation or half-step operation. Full-step operation occurs by energizing two phases at a time, rotating a typical motor 1.8 degrees per step. Half-step operation occurs by alternately energizing one, and then two, phases at a time, rotating the motor 0.9 degrees per step. Full-step operation is suggested for applications that specifically require that mode, such as when retrofitting existing full-step systems.

MOTOR ON/OFF INPUT

The motor on/off input allows de-energizing a motor without disturbing the positioning logic. After re-energizing the motor, a routine can continue. This reduces motor heating and conserves power, especially in applications where motors are stopped for long periods and no holding torque is required. If holding torque is required (such as when lifting a load vertically), then this function should not be used. This output is internally connected to the Indexer. See Section 8 Command Descriptions for further information on Current Hold Command.

FAULT PROTECTION

There are 3 types of fault detection. When a fault is detected, the driver turns off the motor current and the red Fault LED indicates which type of fault occurred. (Located on the top of the driver pack.)

1	LED - Slow Blink	shorted wire in the motor or cable
2	LED - Fast Blink	open wire in the motor or cable
3	LED - ON Steady	ground fault (voltage shorted to 0V)

TABLE 3: FAULT LED

If the driver goes into a fault condition, the fault may be reset by turning the power OFF for at least 15 seconds or by pulling the RESET FAULT input (terminal 4) to a logic “0" for at least 100ms.

SETTING THE KICK CURRENT

The Kick Current should be set to the Motor’s Rated Unipolar Current. For example, a 34D309 is rated for 4.5A, so the Kick Current Potentiometer would be set somewhere between the 4A and 5A indication.

GROUNDING

The unit should be properly grounded. Shielded cable should be used to preserve signal integrity. Contact factory for grounding recommendations.

MOTOR HOOKUP

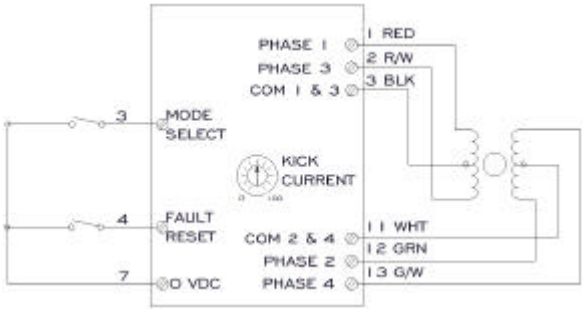
The DPD72401 Series Driver Packs can drive 6-lead and 8-lead step motors rated from 1 to 7 amps/phase (unipolar rating). It features a unipolar bilevel (dual voltage) drive technique with short/open circuit protection (with a Fault LED). This Driver Pack contains a 300 Watt fan cooled power supply.

MOTOR CONNECTIONS

Refer to Section 4 on Motor Connectors for a hookup diagram for Driver Pack applications. *All motor connections must be separated from input connections and all other possible sources of interference.*

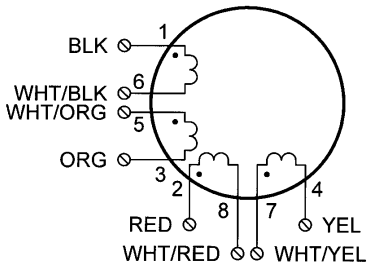
IMPORTANT NOTE: When wiring from the driver(s) to the step motor(s) that extends beyond 25 feet, it is important to consult with the factory.

WIRING DIAGRAM

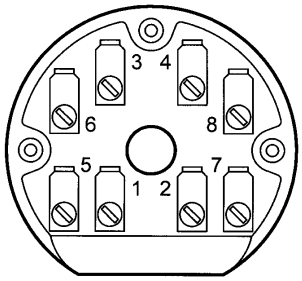


CONNECTION	LEAD NAME	LEAD COLOR	TERMINAL #
4 Lead Bipolar Series	A	Black (BLK)	1
	A	Orange (ORG)	3
	B	Red	2
	B	Yellow (YEL)	4
	None	WHT/BLK & WHT/ORG	6 & 5
	None	WHT/RED & WHT/YEL	8 & 7
4 Lead Bipolar Parallel	A	BLK & WHT/ORG	1 & 5
	A	ORG & WHT/BLK	3 & 6
	B	RED & WHT/YEL	2 & 7
	B	YEL & WHT/RED	4 & 8
6 Lead unipolar	Phase 1	Black (BLK)	1
	Phase 3	Orange (ORG)	3
	Phase 2	Red	2
	Phase 4	Yellow (YEL)	4
	Common 1 & 3	WHT/BLK & WHT/ORG	6 & 5
	Common 2 & 4	WHT/RED & WHT/YEL	8 & 7
Ground		Green/Yellow	Motor Frame

6-Lead Configuration (flying leads)	
Phase 1	RED
Phase 3	RED/WHITE
Common Phase 1 & Phase 3	BLACK
Phase 2	GREEN
Phase 4	GREEN/WHITE
Common Phase 2 & Phase 4	WHITE



8-Lead Configuration



Terminal Board NEMA 34 and 42

MODEL – DPD60401

The DPD60401 will deliver a peak current of 5.5 Amperes per phase at 65 Volts, providing outstanding motor performance. This advanced technology reduces ripple current while maintaining the 20kHz chopping frequency in the motor, causing less heat in both the motor and drive.

MODE SELECT SWITCHES

The MODE SELECT SWITCHES are used to select the divisor of operation from divide by 2 up to divide by 256. Microstep operations are recommended for those applications that specifically require this drive technique, such as retrofitting existing microstep systems. Microstep applications are mainly used for very slow speed operations that require high resolution or very smooth performance. This enhanced resolution can only be accountable for positioning accuracy of a step motor typically ±5% of one full step when dividing by 20 (step angle of .09°) or less. Anaheim Automation recommends the use of the bilevel drives that have a divide by 2 (halfstep mode) for operations ranging in speeds higher than 2Khz, since smoothness can easily be accomplished at this rate.

MICROSTEP SELECTION

The number of microsteps per step is selected by the dip switch(SW1). Table 6 shows the standard resolution values along with the associated settings for these switches. The standard waveforms are sinusoidal.

Resolution	Steps/ Rev	Switch 1	Switch 2	Switch 3	Switch 4
2	400	ON	ON	ON	ON
4	800	OFF	ON	ON	ON
8	1,600	ON	OFF	ON	ON
16	3,200	OFF	OFF	ON	ON
32	6,400	ON	ON	OFF	ON
64	12,800	OFF	ON	OFF	ON
128	25,600	ON	OFF	OFF	ON
256	51,200	OFF	OFF	OFF	ON
5	1,000	ON	ON	ON	OFF
10	2,000	OFF	ON	ON	OFF
25	5,000	ON	OFF	ON	OFF
50	10,000	OFF	OFF	ON	OFF
125	25,000	ON	ON	OFF	OFF
250	50,000	OFF	ON	OFF	Open

TABLE : Mode Selections

Note: *The Microstep Mode Select switches are located on the top cover of the DPD60401.*

ADJUSTING THE RUNNING CURRENT

The output current on the Microstep Driver Pack (DPD60401) is set by the Running Current Potentiometer. This resistance determines the per Phase RMS output current of the driver. Refer to table on Running Current Settings.

Running Current Potentiometer Setting	Current (Amps)
1	1.4
2	2
3	3
4	4
4	4.5
5	5
5.5	5.5

REDUCING OUTPUT CURRENT

The amount of current per Phase in the reduction mode is related to the value of the current adjustment potentiometer (*Running Current*) and the current reduction potentiometer (*Reduced Current*). When the current reduction circuit is activated, the Reduced Current resistance is paralleled with the Running Current resistance. This lowers the total resistance value, and thus lowers the per Phase output current by a percentage.

Pot	1.4 AMPS	2 AMPS	3 AMPS	4 AMPS	5 AMPS	5.5 AMPS
1	50%	42%	32%	26%	22%	21%
2	61%	53%	43%	36%	31%	30%
3	70%	64%	54%	46%	41%	39%
4	75%	70%	61%	54%	48%	46%
5	80%	75%	67%	59%	54%	52%
6	82%	78%	71%	64%	58%	56%
7	84%	81%	75%	68%	63%	61%
8	86%	84%	77%	71%	66%	64%
9	89%	88%	83%	78%	74%	72%

TABLE : Percentage of I(peak) settings on Reduced Current Potentiometer

Note: To obtain a proper setting, adjust the potentiometer arrow to the setting that corresponds to the rated current of the step motor in use. Use the resistance tables that match resistance to rated motor currents.

MOTOR DRIVER CONNECTIONS

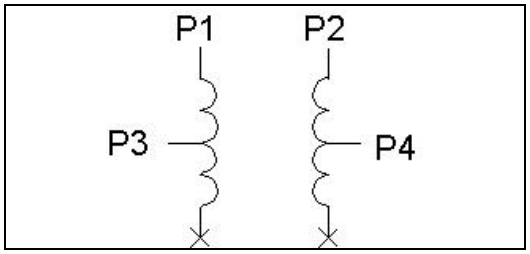
Wiring from the Driver Pack to the motor should be routed away from all other wiring. All electrical connections are made to screw-type terminals for secure and reliable connections. Refer to Section 5, Installation.

STEP MOTOR CONFIGURATIONS

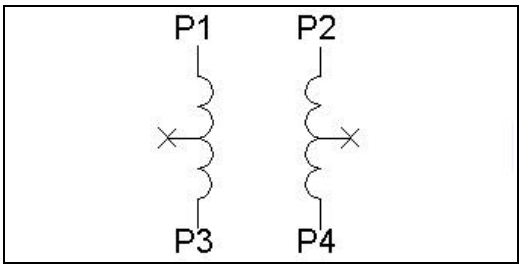
Step motors can be purchased in configurations of 4, 6 or 8 leads. Each configuration requires different current settings. Different lead configurations and the procedures to determine their output current are shown below .

6 Lead Motors

When configuring a 6 lead motor in a ***half-coil configuration*** (connected from one end of the coil to the center tap) use the specified per Phase (or unipolar) current rating to determine the current adjustment pot setting. This configuration will provide more torque at higher speeds.

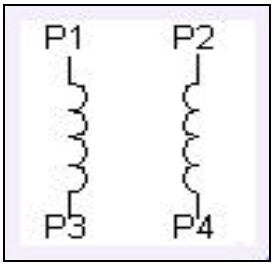


When configuring the motor in a ***series configuration*** (connected from end to end with the center tap floating) multiply the per Phase (or unipolar) current rating by 0.7. Use this result to determine the current adjustment pot setting.



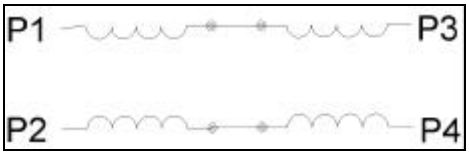
4 Lead Motors

Use the specified ***series*** motor current to determine the current adjustment resistor value. Four-lead motors are usually rated with their appropriate series current, as opposed to the *Phase Current* which is the rating for 6 and 8 lead motors.

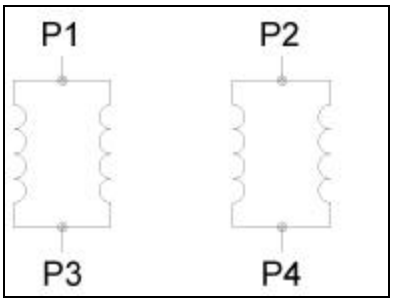


8 Lead Motors

Series Connection: When configuring the motor windings in series, multiply the per Phase (or unipolar) current rating by 0.7. Use this result to determine the current adjustment pot setting.



Parallel Connection: When configuring the motor windings in parallel, multiply the per Phase (or unipolar) current rating by 1.4. Use this result to determine the current adjustment Output Current pot setting.



WARNING! Step motors will run hot even when configured correctly. Damage may occur to the motor if a higher than specified current is used. Most specified motor currents are maximum values. Care should be taken to not exceed these ratings.

Note: A fault light on the DPD60401indicates a short occurred or the over temperature condition setting the fault light. Powering down momentarily will clear the fault.

SECTION 4- FEATURES

PRODUCT HIGHLIGHTS

- Speeds from 0.1 Hz to 2,500,000 Hz
- 24 Inputs Expandable to 96 Inputs
- 24 Outputs
- 16,000 Bytes of Stored Program Memory
- Up to 4 Programs can run Simultaneously (Multitasking)
- Math Functions
- Expandable Modules including Inputs and Thumbwheel Switches
- Encoder Feedback Capabilities
- Free Windows Based Software Included
- Limit Switches
- Addressable for up to 30 Axes
- Baud Rates up to 38,400

ENCODER FEEDBACK

Enhanced encoder feedback commands have been added to allow full control of the encoders. Refer to Section 8 on Encoder Commands for further instructions.

PLC TYPE FUNCTIONS

In the multitasking environment, four programs can run *simultaneously* which gives the unit a PLC like functionality. For instance, Program 1 can contain a program that will turn-on Output 1 when Input 1 goes Low. This will happen independent of everything else going on. This function is similar to a run on the ladder using Ladder Logic on a PLC, and is not found in most sequential programming systems.

MULTIPLE PROGRAMS

The unit can contain up to four programs. Any combination of these programs can be run simultaneously.

VARIABLES

Variables exist in the form of Bits and Registers.

REGISTERS

The Indexer uses a register based command structure. A register may be written to or read. A register is simply a name for a spot where some value is to be stored. Every register has a default value, can be modified, and can be read. For instance, the Base Speed has a default value of 99. This can be changed to another value, and it can also be read. The Base Speed Register is called the MB Register.

BITS

A Bit differs from a register in that its' value may only be a 1 or a 0. In the programming world, a bit being 0 is considered OFF, or False; a bit being 1 is considered ON, or True. All bits can be read, but only some can be written to. The Moving Bit can be read, but writing to that Bit has no meaning, so it is not allowed. In this section, each Bit is described in detail. Some Bits will be affected by commands, and others will not. Bits tell the status of a certain item. For example, the Moving Bit, or MV Bit will give the information if the axis is moving or not. If the axis is moving, the Bit Value will be a 1. If the axis is not moving, the Bit Value will be a 0. Bits are very useful in making information available to the user. Bits can also be used to make decisions, like - *If the Moving Bit is ON then activate output #1* which might turn on an LED to show that the motor is moving. Another example would be a conditional branch - *If the Moving Bit is ON then loop to line number 10*.

MOTOR INPUT BITS

HOME TYPE BIT
This BIT sets the type of homing to execute when the HOME(±) command is issued.

HARD LIMIT INPUT BIT
This BIT indicates whether the HARD LIMIT input is on or off.

HOME LIMIT INPUT BIT
This BIT indicates whether the HOME LIMIT input is on or off.

SOFT LIMIT INPUT BIT
This BIT indicates whether the SOFT LIMIT input is on or off.

MOTION STATUS BITS

MOVING BIT
This BIT indicates whether the motor is moving or stopped, thus it is an output and cannot be written to. A logic "1" indicates the motor is moving.

MOTION COMPLETE BIT
Reading this BIT will indicate whether the specified motor has completed its' motion. A logic "1" indicates that the motor has completed it's move.

MOTOR DIRECTION BIT
The DIRECTION BIT will indicate which direction the motor is moving in. A logic "1" indicates that the motor/axis in moving in the clockwise direction.

MOTOR ERROR CODE BIT
This Bit is enabled if the unit gets a motor error.

AT BASE SPEED BIT
This BIT is read as a 1 whenever the motor/axis is moving at the designated base speed.

AT SPEED BIT
The AT SPEED BIT indicates when the motor is moving at maximum speed.

AT RAMPING DOWN BIT
This BIT is read as 1 whenever the motor/axis is ramping down (decelerating).

AT RAMPING UP BIT
This BIT is read as a 1 whenever the motor/axis is ramping up (accelerating).

AT FAST JOG BIT
This BIT will be read as a 1 if the axis has been put into a fast jog cycle.

AT SLOW JOG BIT
This BIT will be read as a 1 if the axis has been put into a slow jog cycle.

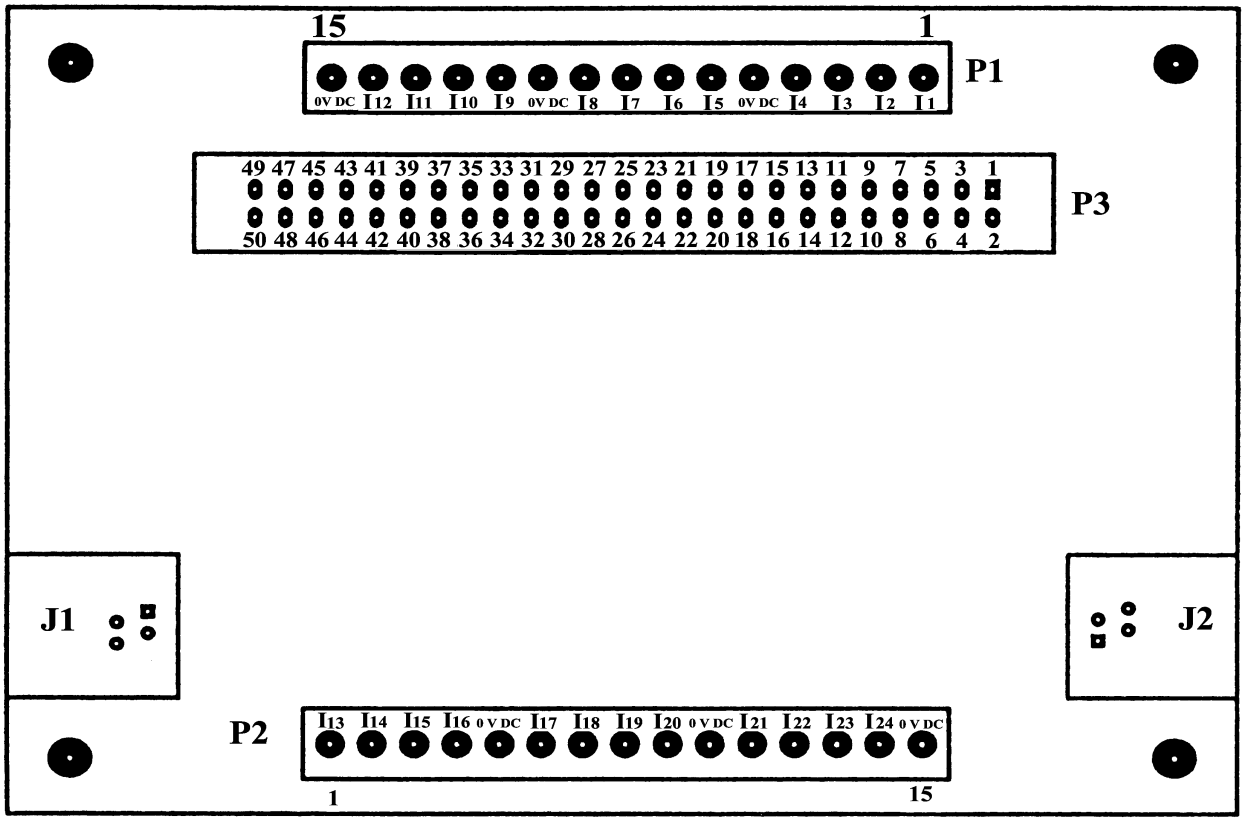
AT RETRY BIT
This BIT will be read as a 1 if the indexer is still retrying to correct for encoder/motor difference.

EXTERNAL MODULES

External Modules offer expansion capabilities for items that are not available in the Driver Pack and are for additional items such as additional Inputs, Thumbwheel Switch Modules and Remote Panel Mount Terminal. (Purchased Separately)

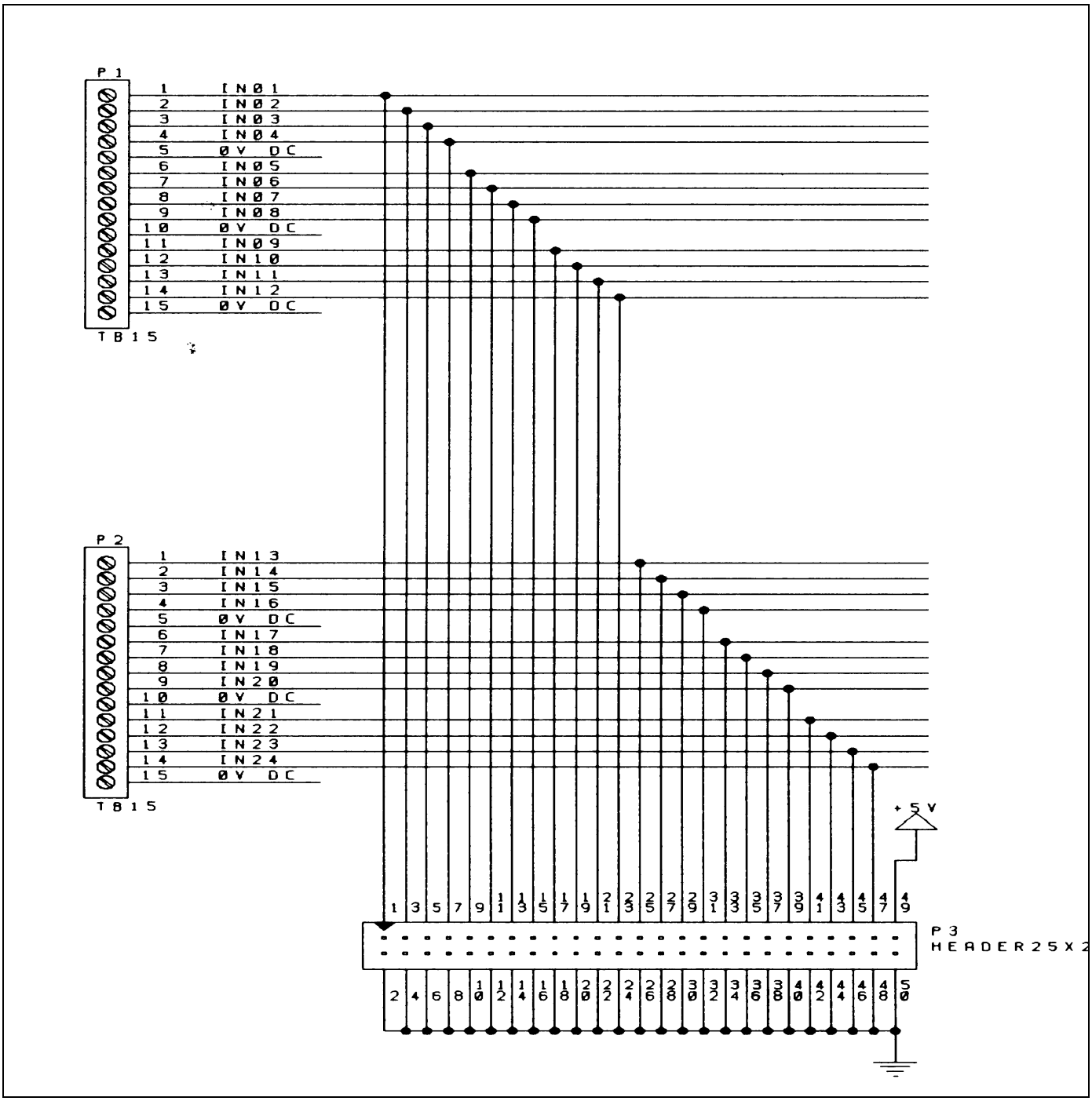
The SMC40M-24I is an expansion module for the SMC40 Programmable Indexer series. Each SMC40M-24I module adds 24 more inputs to the SMC40 - up to three SMC40M-24I modules can be daisy chained for a total of 96 inputs. The expansion modules are simply connected to the SMC40 through a standard telephone handset cable(on JP2) - no external power supply required. JP1 is the Input socket and JP2 is the Output socket for the next module. All 24 inputs are TTL/CMOS compatible and are easily accessible through two detachable terminal blocks. Or using the 50 pin header connector, the input cable can be connected to an Opto22® board for opto-isolation.

Using the Windows based “Intelligent Indexer” utility (that comes with the SMC40), the additional inputs are easily incorporated into SMC40 programs. The SMC40M-24I provides the SMC40 Programmable Indexer a versatile way of dealing with additional inputs for complex automated systems.



SMC40M-24I

SMC40M-24I INTERNAL WIRING



THUMBWHEEL SWITCH

To connect a SMC40M -TWS7seven decade thumbwheel switch, simply connect the cable that is included into the expansion input slot on the Driver Pack. The SMC40M -TWS7 expansion module is used for register manipulation or math functions. (Purchased Separately)

EXAMPLE 1:

	Line	Command	Parameter 1
	1	XMN	TA

Example 1 demonstrates the SMC40 reading the first thumbwheel module(A) value dialed. If three modules were in use, refer to them as A,B,C where the furthest that is multidropped is C.

EXAMPLE 2:

	Line	Command	Parameter 1
	1	Math	RR=TA
	2	Math	R1=RR*400
▸	3	Math	XMN=RR

Example 2 demonstrates the Result Register equal to first thumbwheel module(A) value dialed. A math function is performed with the value and set equal to another temporary register R1. Then the distance number is set equal to the temporary register
value R1.

Connect the SMC40M -TWS7 (maximum of three) to the SMC40 through a standard telephone handset cable (on J2) - no external power supply required. J1 is the Input socket and J2 is the Output socket for the next module. If used in conjunction with the SMC40M-24I Input Module, simply multidrop the thumbwheel switch module(s) along with the input module(s).

REMOTE PANEL MOUNT

The SMTNR2-1 is an ASCII terminal for use with the SMC40 Indexer (Purchased Separately). TheSMTRN2-1 features a 20-key keypad with tactile feedback and a liquid crystal alphanumeric display showing 4 lines of 20 characters. The terminal provides user selectable communications parameters, programmable function keys, and other features which make it ideal for industrial applications requiring flexibility and solid, reliable operation. When properly mounted between the terminal face and panel, the gasket provides a NEMA 4/12 rating. (Dimensions4.9 x 4.9 x 1.43)

The purpose of the remote panel mount can be to operate large scale programs without the need to introduce a PC or an alphanumeric terminal. This allows the operator to enter run time data.

Installation

(WARNING: Never Connect the RS232 Cable and SMTNR2-1to the SMC40 Indexer simultaneously.
A six pin modular connector is provided and used to connect to J1 of the SMC40 Indexer. Although this connector is physically similar to the popular modular telephone connector, the terminal is not compatible with telephone lines or signals. *Connection to a telephone line will damage the terminal and voids the warranty.*

Settings

Parameters	Preset	Options
Baud	9600	300-600-1200-2400-4800-9600
Data Bits 8	7-8	
Parity	None	Even-Odd-Mark-Space-None
Display PE	Enabled	Enable-Disabled
Repeat	Fast	Slow-Fast-Disable
Echo	Disabled	Enabled-Disabled
Handshake	Enabled	Enabled-Disabled
Self Test Disabled	Enabled-Disabled	

SECTION 5 - INSTALLATION

UNIT SELECTION

Each SMC40 can be set to 1 of 10 possible unit numbers. This can be changed by turning the unit address switch to the appropriate position. Refer to Section 2, Figure 1 for location assistance.

BAUD RATE SELECTION

The Baud Rate is the transfer rate of the serial communications. This is how fast the ASCII Data is sent over the transfer lines. The number specifies the number of bits that are sent per second. With a baud rate of 9600, 9600 bits of information are sent in one second. For standard communications (like the SMC40), there is one start bit, one stop bit, and 8 data bits. This means that for every ASCII Character 10 bits are sent, so for the 9600 Baud Rate, 960 ASCII Characters will be sent every second. The Baud Rate is selected by adjusting the Baud Rate Rotary Switch. This switch not only determines the baud rate, but also sets the parameter RTS, for communication with your computer. Table 6 shows the position of the switch for the corresponding baud rates. If you are not sure if your computer uses RTS, select RTS ON. Most computers will work with either RTS ON or OFF.

Baud Rate - the baud rate is set externally by a rotary switch.

BAUD RATE	SWITCH POSITION	
	RTS ON	RTS OFF
50	0	8
300	1	9
1200	2	A
2400	3	B
4800	4	C
9600	5	D
19200	6	E
38400	7	F

TABLE 5: Baud Rate Switch

INSTALLATION – MOUNTING OPTIONS

When installing the Driver Pack, make sure there is adequate space for ventilation. Airflow is necessary to maintain normal operating temperatures for the electronics inside. The Driver Pack should never reach a temperature over 60 degrees Celsius. *Do not* block or cover the vents on the Driver Pack. Two different methods of fastening the Driver Pack are available: bottom and top mounting.

Bottom Mount:

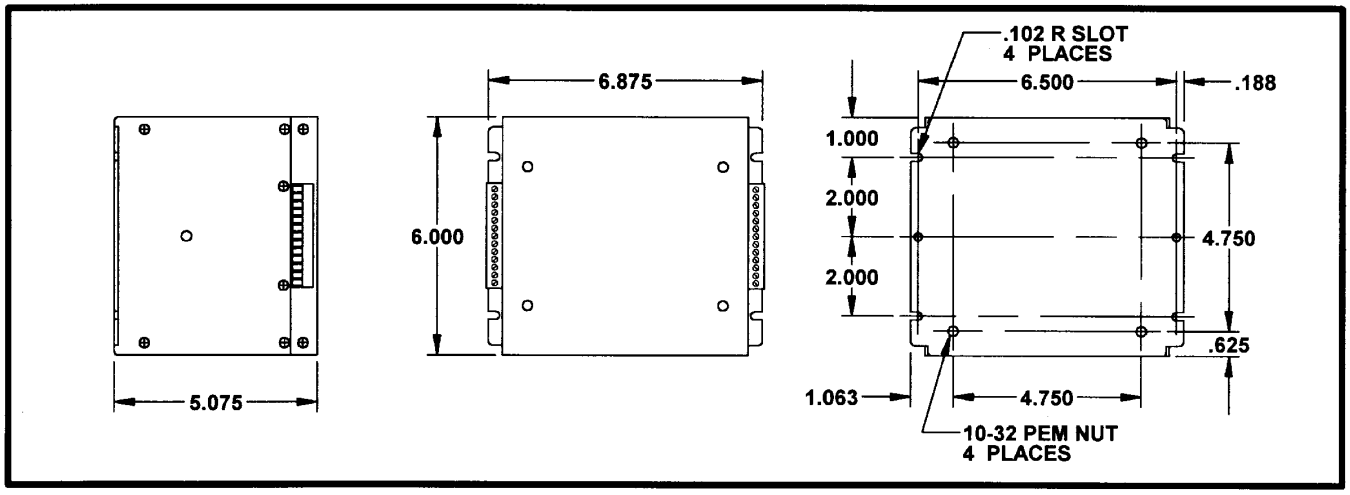
The Driver Pack can be fastened from behind the mounting surface by using four #10-32 x $^{3}/_{8}$" screws.

Top Mount:

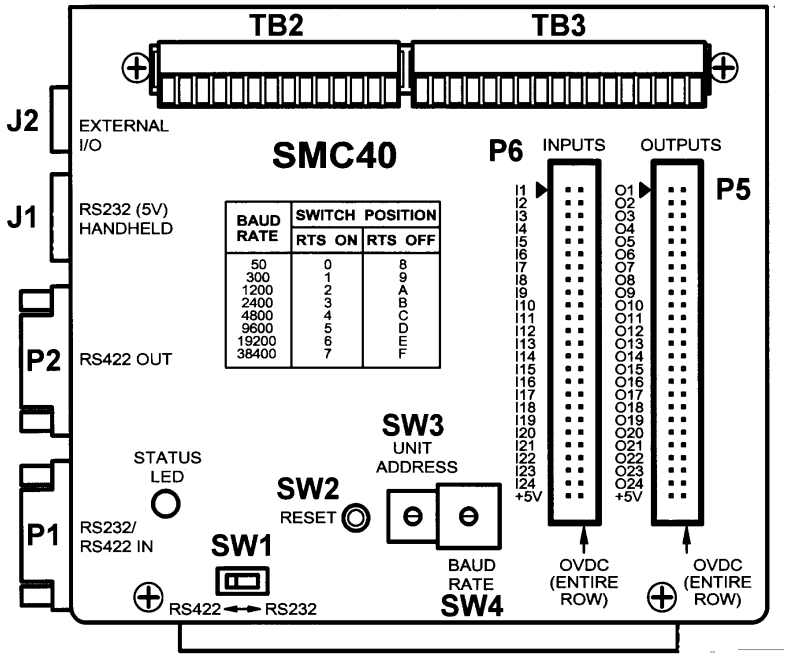
The Driver Pack has mounting flanges on the left and right sides which permits fastening from the top. Each flange consist of two slots and a hole. All four slots can be used to mount the Driver Pack from the front. For easy access mounting, use two slots on one side to slide the Driver Pack into place and then secure it using the hole on the opposite side.

DIMENSIONS - DRIVER PACKS

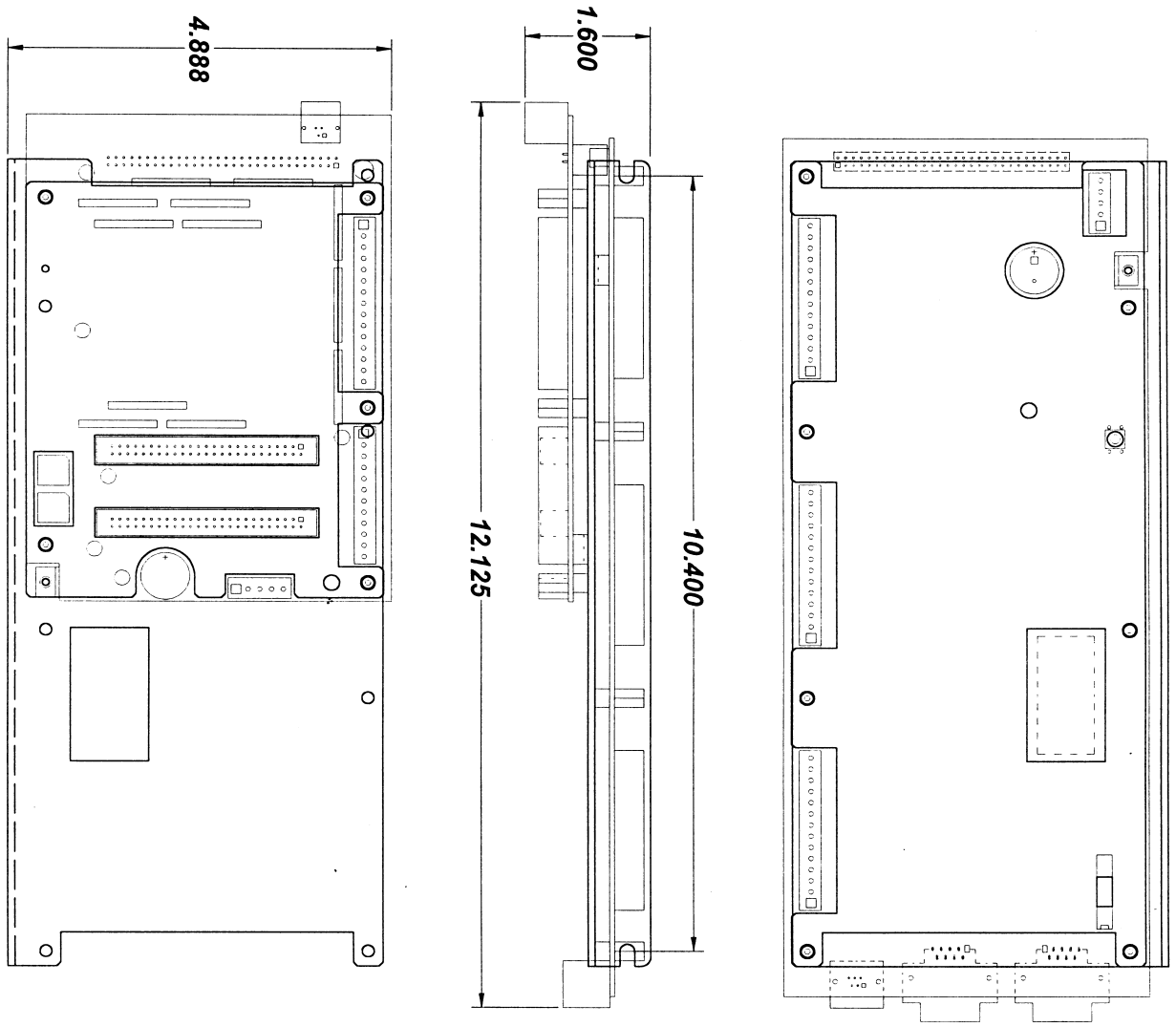
DPD SERIES



INDEXER TERMINAL BLOCK AND SWITCH LOCATION

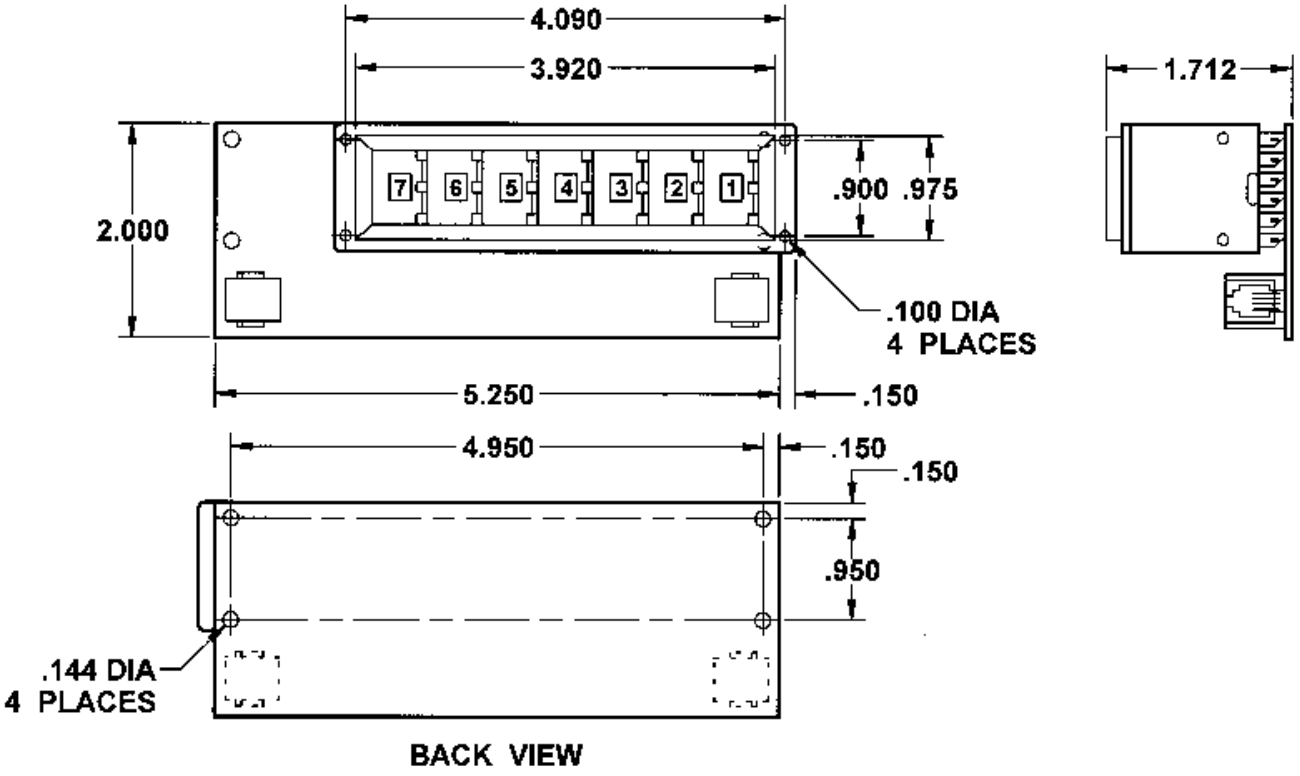


DIMENSIONS – PCL402 & PCL403

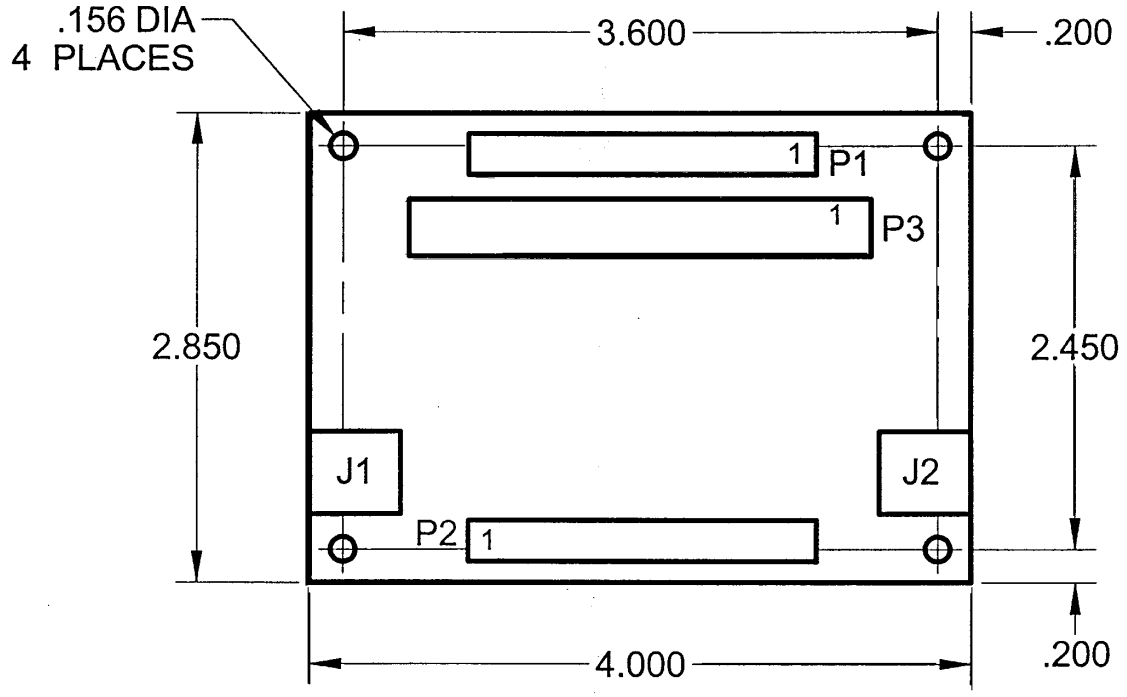


PCL402 & PCL403
OVERALL DIMENSIONS

DIMENSIONS - SMC40M - TWS7



DIMENSIONS - SMC40M - 24I



AC POWER CONNECTION AND FUSE

The fuse is located in a panel underneath the 115 VAC line cord socket. See Section 9, on page 44 Specifications, for the recommended part number.

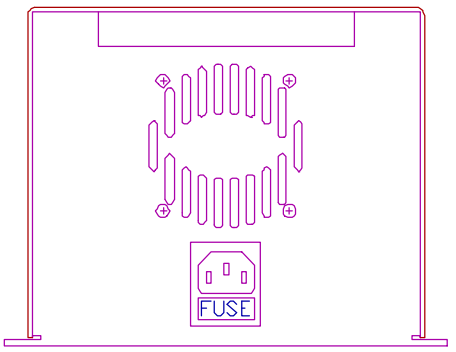
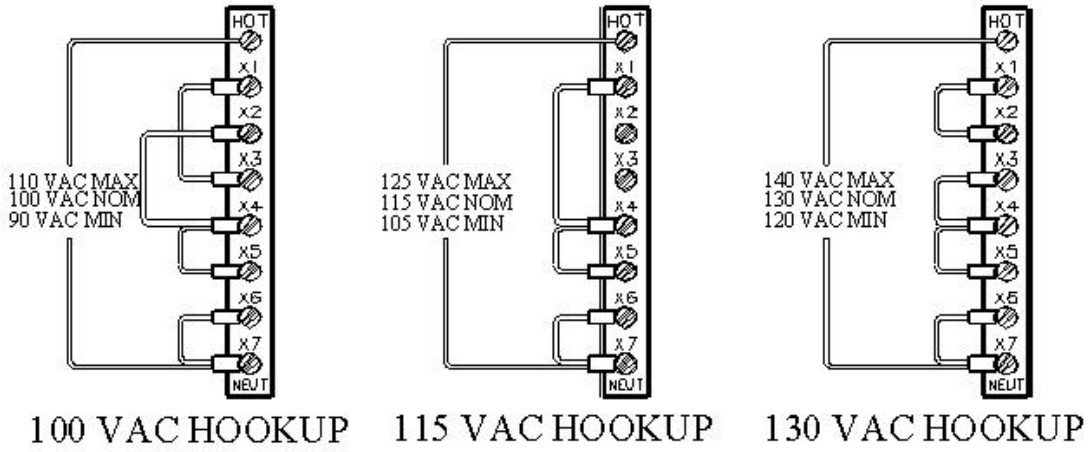
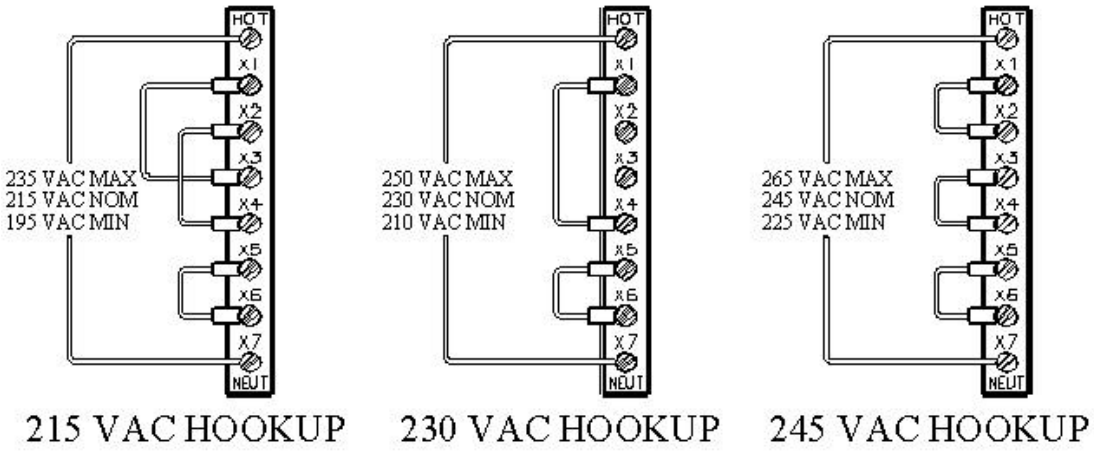


Figure 3

HOOKUP FOR X250 VERSION





SINGLE AXIS INDEXER CONNECTIONS

Pin #	Description
1	+5 VDC
2	Encoder X: Channel A
3	Encoder X: Channel B
4	Encoder X: Channel Z/Marker/Index
5	0 VDC
6	X: Hard-
7	X: Soft
8	X: Home
9	X: Hard+
10	0 VDC
11	***X: CLK (Available on PCL401 ONLY)***
12	***X: DIR (Available on PCL401 ONLY)***
13	***X: PWR (Available on PCL401 ONLY)***
14	***0 VDC (Available on PCL401 ONLY)***

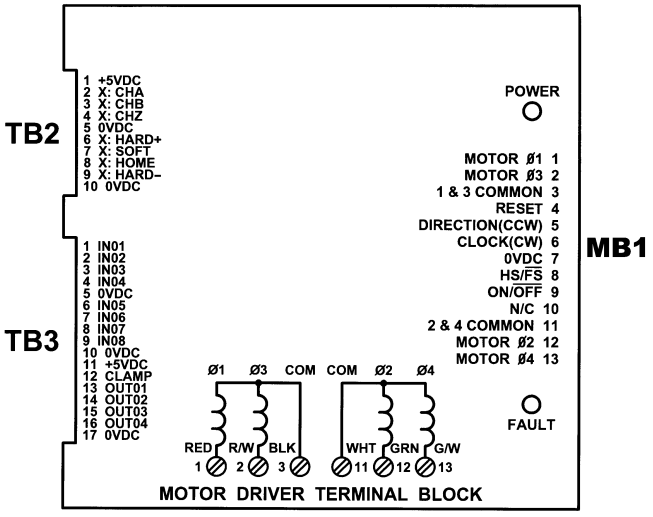
Table 6: Connector TB2 Pinout

Pin #	Description	Pin #	Description
1	Input #1	10	0 VDC
2	Input #2	11	+5 VDC
3	Input #3	12	Clamp
4	Input #4	13	Output #2
5	0 VDC	14	Output #3
6	Input #5	15	Output #4
7	Input #6	16	Output #5
8	Input #7	17	0 VDC
9	Input #8		

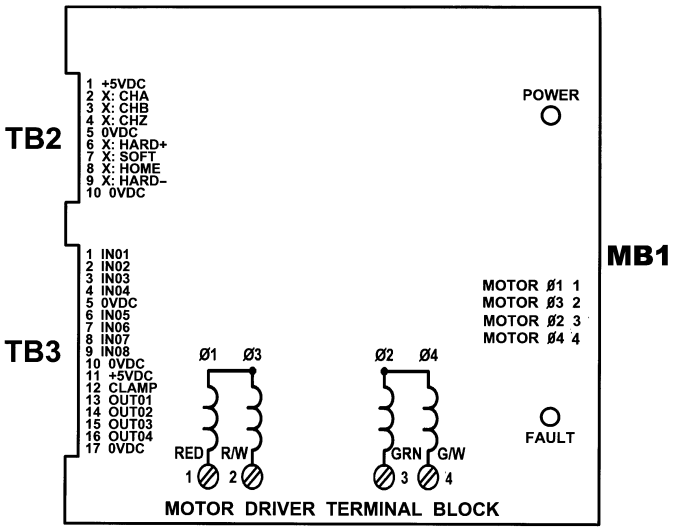
Table 7: Connector TB3 Pinout

Note: Both the Input and Output 50PIN Headers require a flat ribbon cable and the 50PIN Breakout Boards if all the I/O is required. The Anaheim Automation Catalog describes all SMC40 Accessories.

Single Axis DPD72401 Connectors
1 - 7 Amp Bilevel Driver



Single Axis DPD60401 Connectors
1 - 6 Amp Microstep Driver



PCL INDEXER CONNECTIONS

AXIS X, Y, Z

Pin #	Description
1	ISO +5 VDC
2	Encoder X: Channel A
3	Encoder X: Channel B
4	Encoder X: Channel Z/ Marker/ Index
5	ISO 0 VDC
6	X: Hard-
7	X: Soft
8	X: Home
9	X: Hard+
10	ISO 0 VDC
11	X: CLOCK OUTPUT
12	X: DIRECTION OUTPUT
13	X: POWER (ON / OFF)
14	X: 0VDC

Pin #	Description
1	ISO +5 VDC
2	Encoder Y: Channel A
3	Encoder Y: Channel B
4	Encoder Y: Channel Z/ Marker/ Index
5	ISO 0 VDC
6	Y: Hard-
7	Y: Soft
8	Y: Home
9	Y: Hard+
10	ISO 0 VDC
11	Y: CLOCK OUTPUT
12	Y: DIRECTION OUTPUT
13	Y: POWER (ON / OFF)
14	Y: 0VDC

Pin #	Description
1	ISO +5 VDC
2	Encoder Z: Channel A
3	Encoder Z: Channel B
4	Encoder Z: Channel Z/ Marker/ Index
5	ISO 0 VDC
6	Z: Hard-
7	Z: Soft
8	Z: Home
9	Z: Hard+
10	ISO 0 VDC
11	Z: CLOCK OUTPUT
12	Z: DIRECTION OUTPUT
13	Z: POWER (ON / OFF)
14	Z: 0VDC

POWER CONNECTOR

Pin #	Description
1	9 –12 AC1
2	9 – 12 AC2
3	+12VDC UNREG
4	+5VDC REG
5	0VDC

I/O CONNECTOR

Pin #	***IP 1 Connector*** Description
1	IN01/ HC /G1
2	IN02
3	IN03
4	IN04
5	ISO 0VDC
6	IN05
7	IN06
8	IN07
9	IN08
10	ISO 0VDC
11	IN09
12	IN10
13	IN11
14	IN12
15	ISO 0VDC

Pin #	***IP 2 Connector*** Description
1	+5VDC
2	CLAMP
3	OUT01
4	OUT02
5	OUT03
6	OUT04
7	0 VDC
8	OUT05
9	OUT06
10	OUT07
11	OUT08
12	0VDC

ISO POWER CONNECTOR

Pin #	Description
1	O VDC
2	+5 VDC
3	+12 VDC UNREG
4	9 – 12 VAC
5	9 – 12 VAC

NOTE: The ISO Power Connector is different from the Power Connector. Both sides of the PCL402 & PCL403 will need power.

INDEXER INPUTS (FLAT RIBBON HEADER)

PIN #	Description	PIN #	Description	PIN #	Description
1	Input 1	17	Input 9	33	Input 17
2	0 VDC	18	0 VDC	34	0 VDC
3	Input 2	19	Input 10	35	Input 18
4	0 VDC	20	0 VDC	36	0 VDC
5	Input 3	21	Input 11	37	Input 19
6	0 VDC	22	0 VDC	38	0 VDC
7	Input 4	23	Input 12	39	Input 20
8	0 VDC	24	0 VDC	40	0 VDC
9	Input 5	25	Input 13	41	Input 21
10	0 VDC	26	0 VDC	42	0 VDC
11	Input 6	27	Input 14	43	Input 22
12	0 VDC	28	0 VDC	44	0 VDC
13	Input 7	29	Input 15	45	Input 23
14	0 VDC	30	0 VDC	46	0 VDC
15	Input 8	31	Input 16	47	Input 24
16	0 VDC	32	0 VDC	48	0 VDC

Table 8: Connector Pinout

INDEXER OUTPUTS (FLAT RIBBON HEADER)

PIN #	Description	PIN #	Description	PIN #	Description
1	Output 1	17	Output 9	33	Output 17
2	0 VDC	18	0 VDC	34	0 VDC
3	Output 2	19	Output 10	35	Output 18
4	0 VDC	20	0 VDC	36	0 VDC
5	Output 3	21	Output 11	37	Output 19
6	0 VDC	22	0 VDC	38	0 VDC
7	Output 4	23	Output 12	39	Output 20
8	0 VDC	24	0 VDC	40	0 VDC
9	Output 5	25	Output 13	41	Output 21
10	0 VDC	26	0 VDC	42	0 VDC
11	Output 6	27	Output 14	43	Output 22
12	0 VDC	28	0 VDC	44	0 VDC
13	Output 7	29	Output 15	45	Output 23
14	0 VDC	30	0 VDC	46	0 VDC
15	Output 8	31	Output 16	47	Output 24
16	0 VDC	32	0 VDC	48	0 VDC

Table 9: Connector Pinout

Note: Both the Input and Output 50PIN Headers require a flat ribbon cable and the 50PIN Breakout Boards if all the I/O is required. The Anaheim Automation Catalog describes all SMC40 Accessories.

MOTOR CONNECTORS

TABLE 10A: Connector MB1 (2)

Pin #	Description
1	**Phase 1**
2	**Phase 3**
3	**Common Phase 1 & 3**
4	Reset Fault Input
5	Direction (internally connected)
6	Clock (internally connected)
7	0 VDC
8	On/Off (internally connected)
9	Halfstep/ Fullstep
10	No Connection
11	**Common Phase 2 & 4**
12	**Phase 2**
13	**Phase 4**

Pinout for BLD72 Driver Series
(1 – 7 Amps Bilevel)

Pin #	Description
1	Phase 1
2	Phase 3
3	Common Phase 1 & 3
4	Common Phase 2 & 4
5	Phase 2
6	Phase 4

Pinout for DPF11 Series
(2 - 12.5 Amps Bilevel)

Table 2D: Connector MB1

Pin #	Description
1	Phase 1 A
2	Phase 3 /A
3	Phase 2 B
4	Phase 4 /B

Pinout for DPD60401 Series
(1 - 6 Amps Bilevel)

Figure 4: Six Lead Motor Connection

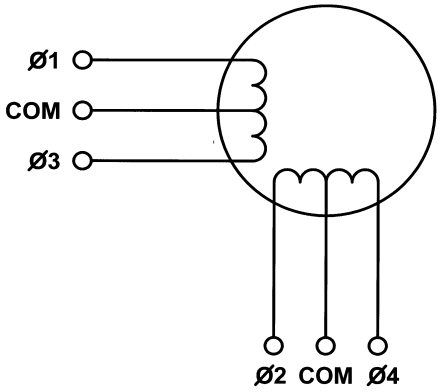
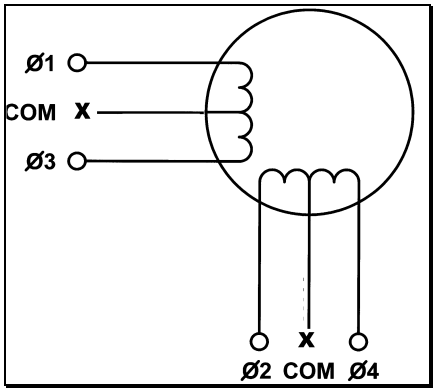
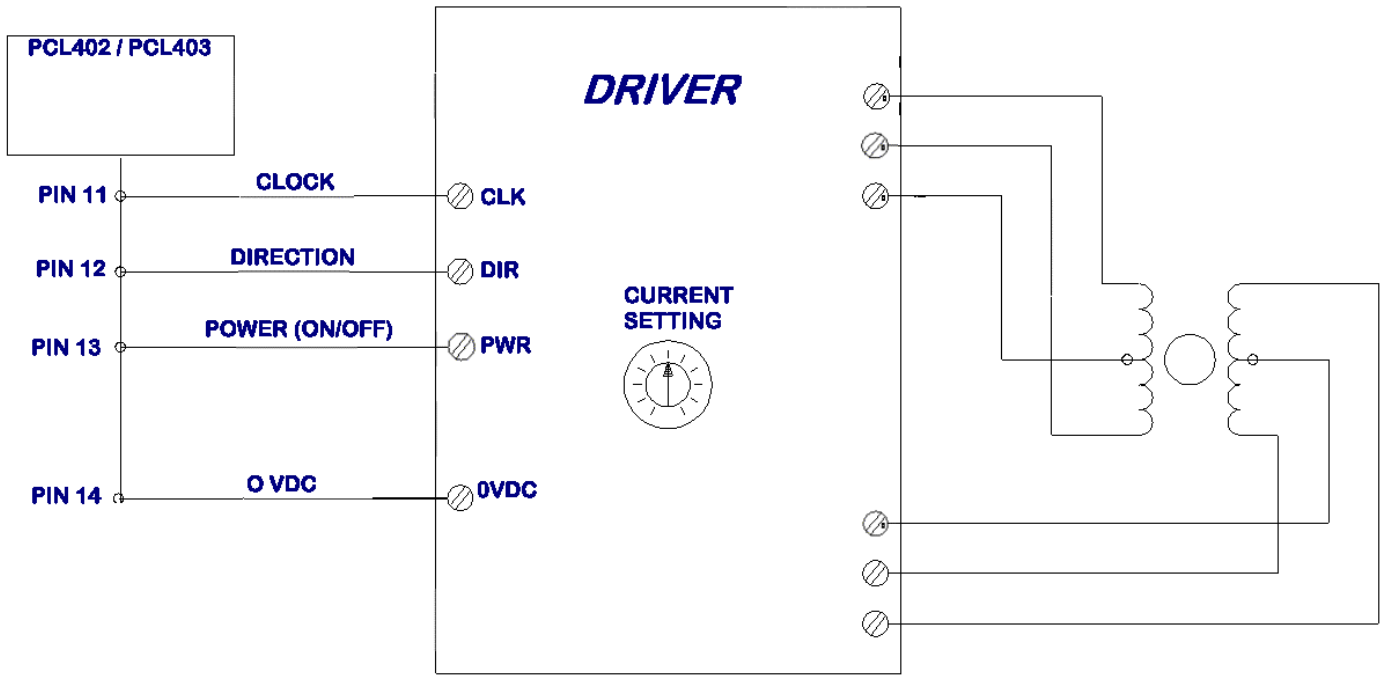


Figure 5: Four Lead Motor Connection



PCL HOOKUP DIAGRAM



Note: Contact the factory direct if you have questions regarding the hookup procedures.

SECTION 6 - COMMUNICATIONS

TALKING TO THE INDEXER

Anaheim Automation's SMC40 Programmable Indexers communicate by using the RS232C or RS422 standards. Most computers contain at least one RS232 serial port. Some industrial computers have a RS422 serial port. To communicate with the SMC40, use connector P1. P1 is used for either R3232 or RS422, and is set by sliding the two switches to the appropriate direction. P1 is a DB9 Female. To communicate with subsequent axes, use P2, the RS422 output port. P2 is a DB9 Male, and is always set for RS422, regardless of the switch setting for P1. The switches affect only the Input Port P1. The difference between the two types of communications is discussed below.

RS232

This 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. For the four signals, RD, TD, CTS and RTS to be transmitted, RS232C requires five wires. The signal line maintains levels of +5VDC to +15VDC (LOW LOGIC INPUT) and -5VDC to -15VDC (HIGH LOGIC INPUT). The receiver for the RS232 looks for a voltage potential of +3 to +25 volts for a logic LOW, and -3 to -25 volts for a logic HIGH. For a valid logic level, the voltage must be +/-3 volts. RS232 works well at 9,600 baud over 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 control wiring separated from motor cable/wiring.

RS422

To talk to the SMC40 in RS422 set the switch to RS422, and use P2. The RS422 serial communication standard is differential. This means that from each signal, there are two wires. For the four signals transmitted there needs to be nine wires including the ground reference. The signal line maintains a voltage level of up to +12 volts on either line. The polarity of the line switches is used to obtain the logic levels. For example, if RD+ is more positive than RD- then it is a logic HIGH. If RD- is more positive than RD+, then it is a logic LOW. For a valid logic level, the voltage difference between RD+ and RD- needs to be greater than 200 millivolts. RS422 is unsusceptible to noise due to the differential lines. RS422 is specified at a maximum of 9600 Baud at up to 4000 feet.

RS232C 25 PIN CONNECTOR	FUNCTION	RS232 FUNCTION
1	CG	Chassis Ground
2	TD	Transmit Data
3	RD	Receive Data
4	RTS	RTS - Request To Send
5	CTS	CTS - Clear To Send
7	O VDC	SG - Signal Ground

TABLE 12: RS422 Connector Pinout

RS422 9 Pin Connector	FUNCTION	DESCRIPTION
1	SG	signal ground
2	CTS+	clear to send
3	CTS-	clear to send-
4	TD+	transmit data+
5	TD-	transmit data-
6	RTS+	request to send+
7	RTS-	request to send-
8	RD+	receive data+
9	RD-	receive data-

TABLE 13: RS422 Connector Pinout

HANDSHAKING SIGNALS

There are two "handshaking" signals: RTS (Request to Send) and CTS (Clear to Send). Some devices use these handshaking signals and others do not. It is important to know if your device supports certain handshake signals. Anaheim Automation Indexers support both of these signals.

DTE VS DCE

(THE COMPUTER IS THE DTE.....THE INDEXER IS THE DCE)

Signal	9 PIN Connector	DIRECTION	FUNCTION
SG	5	0 VDC	Signal Ground
TD	3	DTE to DCE	transmitted data
RD	2	DCE to DTE	received data
RTS	7	DTE to DCE	request to send (DTE ready)
CTS	8	DCE to DTE	clear to send (DCE ready)

TABLE 14: Pin Description for RS232 with a DB9

There are two types of devices defined. The first is called DTE (data terminal equipment). Examples of this would be a terminal, or an IBM Compatible Computer. The second type of device is a DCE (data communication equipment). Examples of this would be a modem or an Anaheim Automation Indexer such as the SMC40. DTE's have input pins of one type corresponding to output pins on the DCE's.

NOTE: THE SIGNAL NAMES ARE FROM THE POINT OF VIEW OF THE DTE COMPUTER. FOR EXAMPLE, PIN 3 IS CALLED TD (TRANSMIT DATA) BY BOTH SIDES, EVEN THOUGH THE DTE (COMPUTER) SENDS IT AND THE DCE (SMC40) RECEIVES IT.

With a DB9, a DTE (such as a computer) transmits on pin 3 and receives on pin 2.
With a DB9, a DCE (such as a SMC40) transmits on pin 2 and receives on pin 3.

"A MANNER OF SPEAKING"

The communication signals supported by Anaheim Automation Indexers are: **RECEIVE, TRANSMIT, CLEAR TO SEND (BUSY), AND REQUEST TO SEND**.

The method in which the Computer and the Indexer communicate is as follows: When the computer wants to send some information, it looks at the CTS (Clear To Send) line. This will inform the computer if the Indexer is ready to receive information. If a logic LOW is read (meaning it is clear to send), the computer will send information on pin 3, in which the Indexer will receive on pin 3.

When the Indexer receives data that requires some computational time, it will pull the CTS HIGH meaning it is not clear to send data.

When the Indexer is ready to send something to the Computer it looks at the RTS signal which will inform the Indexer that the Computer is busy. If the RTS is low then the Indexer will send information on pin 2, which will be received by the Computer on pin 2 also.

RTS DEFINED

On the SMC40, there is an option to either enable, or disable the RTS. If RTS is enabled, then the above description applies. If RTS is disabled, then when the SMC40 wants to send information to the Computer, it will send it without looking at the RTS line. This is used when the computer does not support the RTS line.

CTS DEFINED

The CTS line must always be supported. No information should be sent to any indexer unless the CTS line is low. Otherwise the data sent may be lost, and the indexer could possibly stop communicating.

NOTE: THE SIGNAL NAMES ONLY MAKE SENSE FROM THE POINT OF VIEW OF THE DTE. FOR EXAMPLE, PIN 3 IS CALLED TD (TRANSMIT DATA) BY BOTH SIDES, EVEN THOUGH THE DTE SENDS IT AND THE DCE RECEIVES IT.

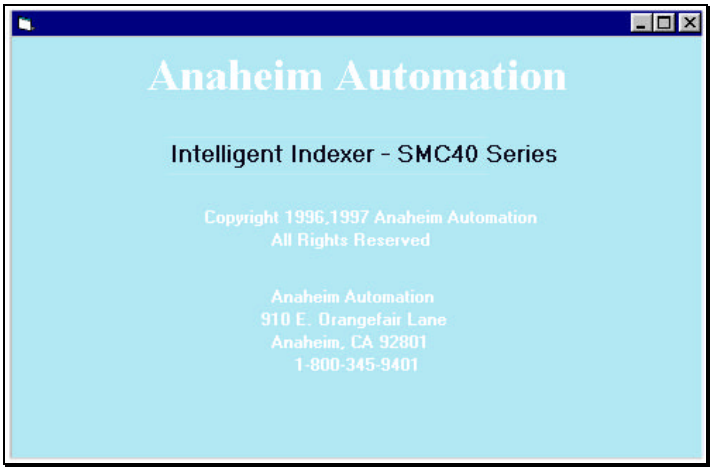
Pin #	Name	Description
1	CG	Chassis Ground
2	TD	Transmit Data
3	RD	Receive Data
4	RTS	Request To Send
5	CTS	Clear To Send
7	0 VDC	Signal Ground

TABLE 15: RS232C 25 Pin Connection (computer port)

SECTION 7-INTELLIGENT SOFTWARE

DESCRIPTION

Included with a SMC40 purchase Driver Pack is a Windows software package that is used to program the SMC40 indexer. This software will allow for easy programming of the Driver Pack. The SMC40 software runs on Microsoft Windows 3.1 (*Version 1.12*) or Windows 95/ 98/ NT (*Version 1.13*).



INSTALLATION

Windows 3.1

To install the software you need to go into Windows. From the Program Manager select the menu **File**, and then **Run**. Put the Intelligent Indexer Disk in the appropriate disk drive and select Setup.

It might look like: ***A:\setup .exe***

Follow the instructions on the screen.

Windows 95/ 98/ NT

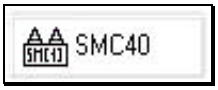
Put the Intelligent Indexer Disk in the appropriate disk drive.
Go to the Start Button, and Select **Run**. Choose **A:\setup.exe** and follow the instructions on the screen.

SOFTWARE DEFAULTS

Run the Intelligent Indexer Software by Double Clicking the SMC**40** Program File in Windows.

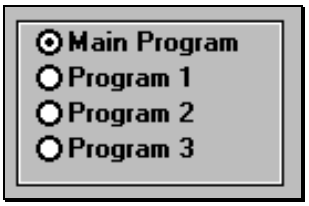
The Defaults are:

Com Port	Com1
Baud Rate	9600
Units	Steps
Number of Axes	1



THE FOUR PROGRAMS

There are four different program areas available. These are the Main Program, Program 1, Program 2, and Program 3. When running the Driver Pack, these programs can run simultaneously. They have access to the same set of registers, and can affect the other program with some of the appropriate commands. This allows a programmer to write sophisticated routines that would not be possible with other similar products.



MAIN PROGRAM

This is area where most programs will reside. When a program is started by the Menu Item *START*, or powered up after setting the Autostart Flag, the Main Program will start running. At this point, the other 3 program areas will be idle. There are commands that can start the other three programs. The program can be 1000 lines or longer, depending on the type of commands used and the space used in the other programs.

PROGRAM 1, 2 and 3

These are the secondary programs that will run concurrently with the Main Program.
The user must enter a Branch Quit statement in all Programs 1,2, & 3 before sending/ compiling the Main Program.

MULTITASKING

The word multitasking in this context means that four programs can run at the same time. This use of the multitasking environment will allow PLC-like functions to be programmed into this unit. One example would be to turn an output on for one second whenever an input goes on. To do this with standard sequential programming, the programmer would constantly have to check for that input, while the program is still doing all that it has to do as well. This can often overburden the program so that another module has to be purchased to do that function. With the multiple programs available in this unit, it is like having four separate modules all wrapped up into 1. With this unit, that program can be put into Program 1, 2, or 3. This off-loads the Main Program from having to continuously check for that condition.

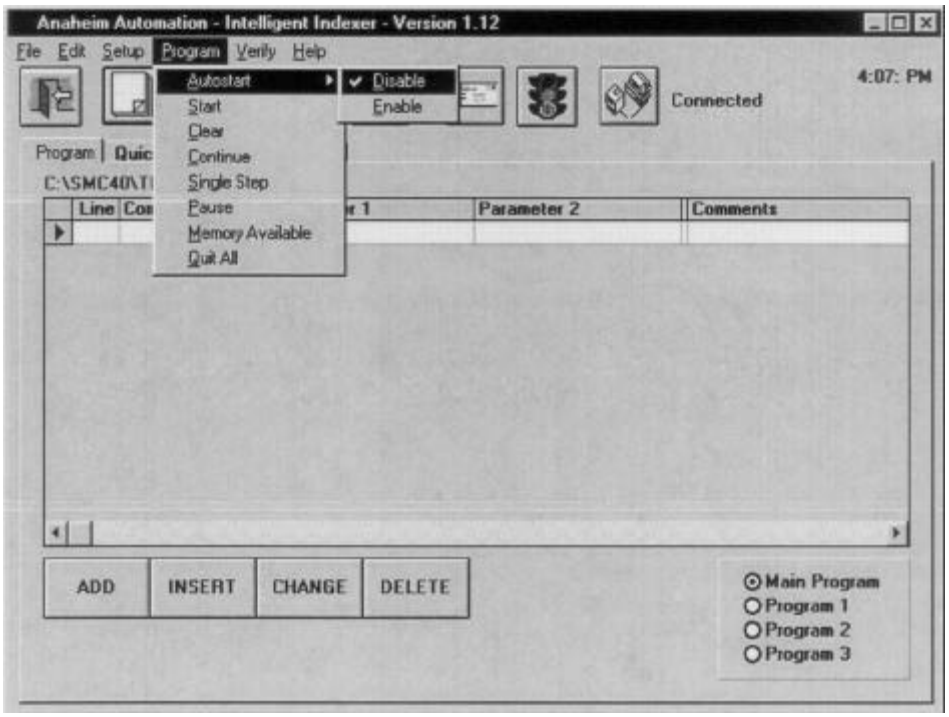
ADDING, INSERTING, CHANGING OR DELETING A COMMAND

To add a command to the program, select the appropriate choice from the main screen –
ADD, INSERT, CHANGE or DELETE.

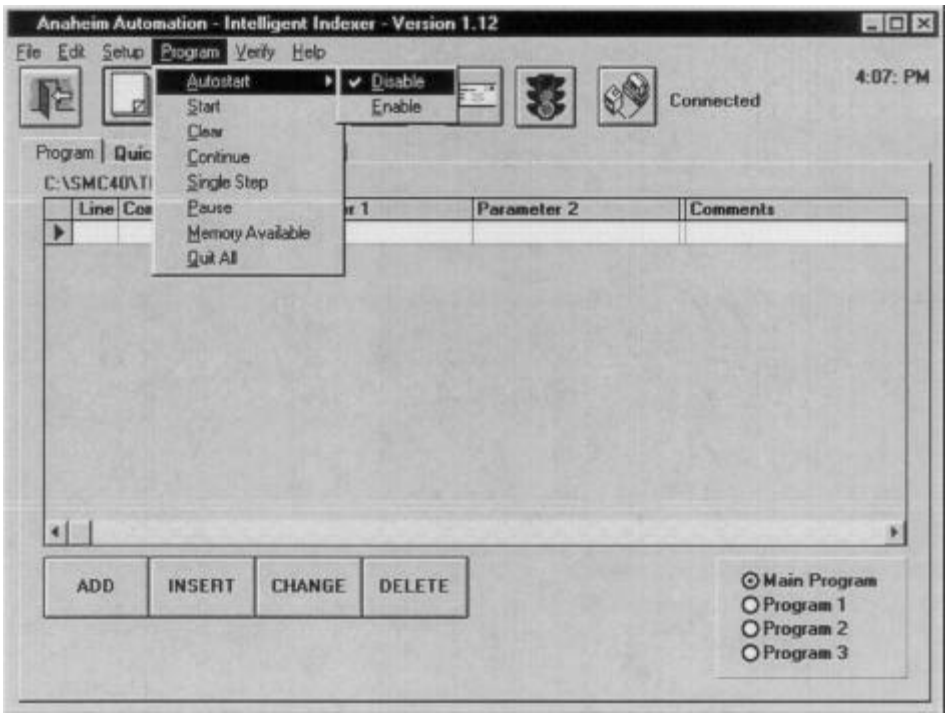
ADD:	This command allows the user to place additional commands at the bottom of the program.
INSERT:	This command allows the user to insert additional commands at the cursor location.
CHANGE:	This command allows the user to change commands at the cursor location.
DELETE:	This command allows the user to delete commands at the cursor location.

AUTOSTARTING

The programs that have been tested and ready for execution can be downloaded to the SMC40 Indexer and Autostart ***Enabled.*** This feature will now allow the SMC40 Indexer to be disconnected from the PC and the program stored can now start independent of the PC.



The program can also be Autostart ***Disabled*** when the program needs editing. Please reconnect the PC to SMC40 Indexer and note that the check mark to the left of the Disable Command of the SMC40 Menu Commands will appear when the Disable Command has been clicked/ selected ***twice***.



SECTION 8 - COMMAND DESCRIPTIONS

This section covers the broad range of commands available in the SMC40. Commands are grouped together with other similar Commands. They include Branching, Start/Stop, Motion Parameters, Outputs, User Entry, Encoder Commands, Math, Terminal Commands, Jog Inputs and Program 1,2,3. After clicking the Insert, Add or Change Button, the Select a Command screen comes up to select commands. Select the Button of your choice and it will offer you a selection of commands from that group.

Below is a list of Command Groups, and the commands that correspond to those groups.

Branching	
	Label
	Goto
	If (bit) Then
	If (Reg) Then
	Gosub
	Return
	Quit
	Wait Delay
	For Loop
Start/Stop	
	Go Absolute
	Go Relative
	Home
	Slew
	Stop Hard
	Stop Hard
	Wait Motor
Motion Parameters	
	Base Speed
	Current Hold
	Dir+
	Dir-
	Max Speed
	Position
	Accel/ Decel
	Speed Limit
	Slow Jog Speed
	Fast Jog Speed

Outputs	
	Set Outputs
User Entry	
	User Entry
Encoder Commands	
	Encoder Position
	Encoder Autocorrection
	Encoder Delay
	Encoder Retries
	Encoder Window
	Encoder Motor Ratio
Math	
	+ Addition of Registers
	- Subtraction of Registers
	/ Division of Registers
	* Multiplication of Registers
Terminal Commands	
	Write Text
	Write Value
	Write ASCII
	Write Value
Jog Inputs	
	Jog+
	Jog-
	Fast Jog
Program 1,2,3	
	Start Program 1
	Start Program 2
	Start Program 3
	Stop Program 1
	Stop Program 2
	Stop Program 3

BRANCHING COMMANDS

Branching Commands cause the program to go to a specific line number or label. This sometimes will occur if a condition exists, like an input being activated. The most common use will be to have a machine operator activate a switch before the machine begins operation. Another use of these types of commands would be for a program to continuously go from the bottom of the program back to the top of the program. The Branching Commands cause the program to 'branch' to another part of the program based on a set of conditions, like a switch being pressed, a register value equal to a number, a bit set, or many other conditions.

GOSUB COMMAND

When the Gosub Command is implemented, the program will go to the specified line and then the Return Command will send it back to the line below the original Gosub Command. *This must be used in conjunction with the Return Command.*

This sample will cause the next instruction to go to the label Function 2, execute the lines 68 through 74, and then return to line 15..

Line	Command	Parameter 1	Parameter 2	Comments
14	Gosub	FUNCTION2		
*				
67	Label	FUNCTION2		
*				
75	Return			Returns to Line 15

GOTO COMMAND

The Goto Command causes the program to go directly to the specified line. This line can be described by a Line Number, a Register Value, or a Label. When using a Register Value, you must make sure that there is a value stored in that Register. If there is a value, for example 10, then this command will cause the program to jump to line 10 for the next command.

This sample will cause the next instruction to go to the Label Top.

Line	Command	Parameter 1	Parameter 2	Comments
14	Goto	TOP		

IF BIT COMMAND

The IF BIT COMMAND is a conditional statement used to execute another command based on whether or not a certain Input condition is met.

Line	Command	Parameter 1	Parameter 2	Comments
1	If I1 = 0	Then TOP	Else CONTINUE	

IF REGISTER COMMAND

The IF REGISTER COMMAND is a conditional statement used to execute another command based on whether or not a certain Register condition is met.

Line	Command	Parameter 1	Parameter 2	Comments
1	If R1 = 0	Then TOP	Else CONTINUE	

LABEL COMMAND

The LABEL command is simply used to label a line. As mentioned before, labels are commonly used with branching statements.

Line	Command	Parameter 1	Parameter 2	Comments
1	Label	TOP		

BRANCH QUIT COMMAND

The BRANCH QUIT command will cause the program to stop execution of the associated program (Main Program, Program 1, 2, or 3).

Line	Command	Parameter 1	Parameter 2	Comments
1	Branch Quit			

RETURN COMMAND

The Return Command will cause the program to return to the line below the last Gosub Command that was issued. The return command *must* be used in conjunction with the Gosub Command or an error will occur.

Line	Command	Parameter 1	Parameter 2	Comments
1	Return			

WAIT DELAY COMMAND

This Command sets the Delay Register, and waits for the time specified to expire before continuing to the next line in the program. The Value is in 1/1000 of a second, meaning that a value of 1000 is 1 seconds. The below example would begin moving 1000 steps. After the start of the motion, 1 seconds would need to expire before the output is turned on. Value(s) of 1 - 60000 equivalent to .001seconds - 60seconds is the usable delay range per line.

Line	Command	Parameter 1	Parameter 2	Comments
1	Go Relative	1000		
2	Wait Delay	1000		
3	Set Outputs	ON:1,		

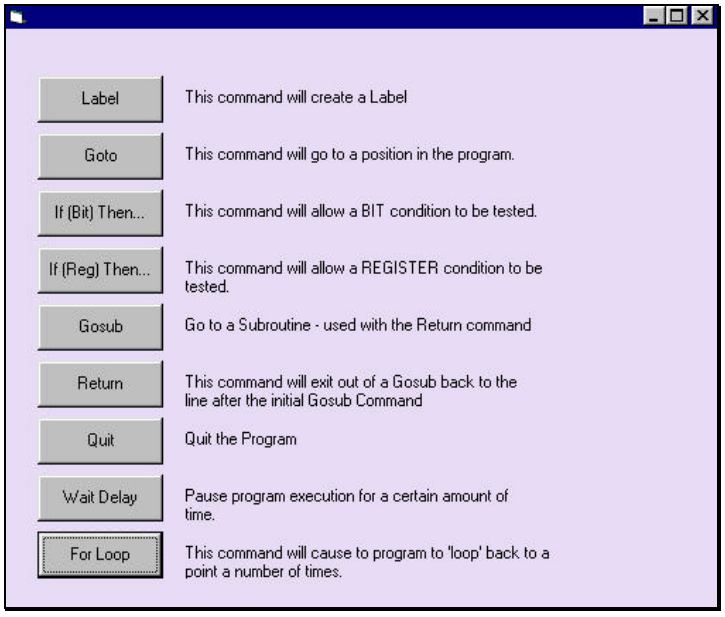
FOR LOOP

The For Loop Command is separated into two sections where the Top Loop Identifies the top section of the looping command for a pre-selected number of times.

Line	Command	Parameter 1	Parameter 2	Comments
1	Loop Top Start	4 Times		

The Bottom Loop Identifies the bottom section of the looping command in order to close the loop.

Line	Command	Parameter 1	Parameter 2	Comments
10	Loop Bottom Start			



START/ STOP COMMANDS

This grouping of command pertains to functions that will Start and Stop the Motion. All axis motion begins at the specified Base Speed and then ramps up to Max Speed. If the motion is definite, the axis will then ramp from Max Speed down to Base Speed to complete the move. Setting Motor Base, Max, and Ramping Speeds is discussed in the Motor Parameter Command Section.

GO ABSOLUTE COMMAND

This Command is a move-to-position command meaning that it will move to the position specified. For instance, in the below example it will cause the motor to go to position 0. If the Motor Position is at 2500, the Go Absolute Command of 0 will move to the position and the Direction Register would be 0 (CCW Direction).

Line	Command	Parameter 1	Parameter 2	Comments
1	X Go Absolute	0		

GO RELATIVE COMMAND

This Command is a move-a-distance command meaning that it will move the distance set in the Go Relative Command. For instance, in the example below it will cause the motor to go 1000 steps in the CCW direction

Line	Command	Parameter 1	Parameter 2	Comments
1	X Dir -			
2	X Go Relative	1000		

HOME COMMAND

This Command will cause the motor to move continuously in the direction specified by a Direction Command until a Home limit switch is hit. (Refer to the HOME TYPE COMMAND for Home Types available.)

Line	Command	Parameter 1	Parameter 2	Comments
1	X Dir -			
2	X Home	0		

HOME TYPE COMMAND

This Command specifies the type of Homing that will be executed when a Home Command is executed. There are 3 types of Homing.

Home Type 0

This will rotate continuously in the appropriate direction and speed until a Soft Limit Switch is active. It will then Ramp Down to Base Speed, and then Stop when the Home Limit Switch is active.

Home Type 1

This will rotate continuously in the appropriate direction and speed until a Home Limit Switch is *active*. It will then Ramp Down to Base Speed, Stop and instantly reverse direction. It will then continue in this reversed direction and then Stop when the Home Limit Switch is once again *inactive*. This will stop on the 'negative edge' of the switch only, so the switch can be passed on ramp down and then hit again in the reverse direction stopping after it comes *off* the switch. This is most useful for taking backlash out of a leadscrew.

Home Type 2

This will rotate continuously in the appropriate direction and at *Base Speed* until a Marker Pulse is active. This is typically called Channel Z from an encoder. Although the intent of this Homing is for use with an encoder, it is not limited by it. A simple switch will work as well.

SLEW COMMAND

This Command will cause the motor to move continuously in the direction specified by a Direction Command.

Line	Command	Parameter 1	Parameter 2	Comments
1	X Dir -			
2	X Slew			

STOP HARD COMMAND

This Command will cause the motion to stop immediately without any ramping down. If the Motor was moving at speeds higher than what it can reliably start and stop at, it is likely that the Motor will not be at the position that the Position Register reads.

	Command	Parameter 1	Parameter 2	Comments
1	X Stop Hard			

STOP SOFT COMMAND

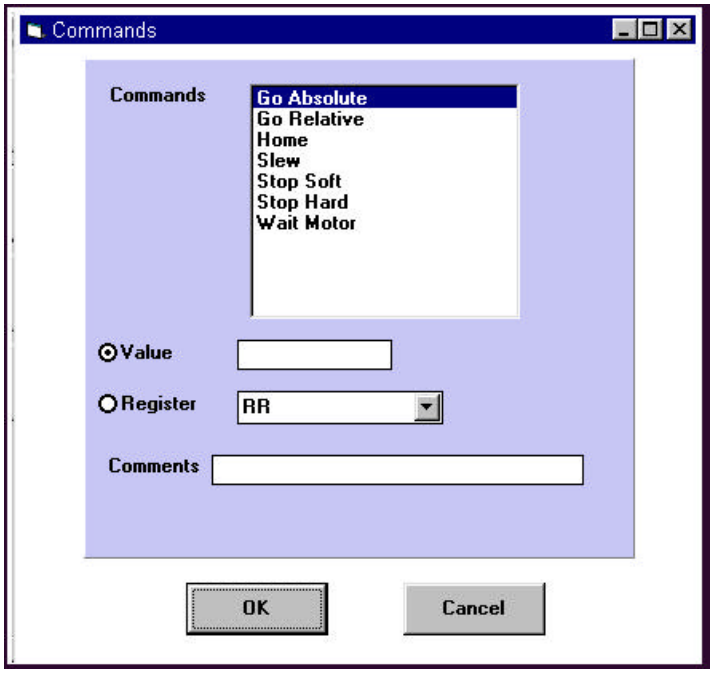
This Command will cause the motion to stop immediately *after* ramping down. This causes the Motor to ramp down to Base Speed before stopping. This gentler way of stopping is not likely to cause positioning errors like the Stop Hard Command would.

	Command	Parameter 1	Parameter 2	Comments
1	X Stop Soft			

WAIT MOTOR COMMAND

This Command waits for the Motor to stop moving before continuing to the next line in the program.

Line	Command	Parameter 1	Parameter 2	Comments
1	Go Relative	1000		
2	Wait Motor			
3	Set Outputs	ON:1,		



MOTION PROFILE

The Motor Commands are those commands that affect the motion of the Motor. These commands will set the position that the motor is at, the distance it is to travel, the speed at which it will travel, and how long current will remain in the motor after it has stopped.

BASE SPEED COMMAND (1 to 2,500,000)

This Command will determine the speed at which the Motor *starts* at. Step Motors can go from rest immediately to a speed in the motors start/stop region which varies from motor to motor. This is similar to a Ski-Lift taking skiers up to the top of a mountain. The lift is traveling at a certain speed and the skiers attach themselves to the lift (usually by sitting down) and *instantly* are going the Ski-Lifts velocity. A step motor can start from rest in a similar fashion, and this starting speed is called the Base Speed.

Line	Command	Parameter 1	Parameter 2	Comments
1	X Base Speed	500		

CURRENT HOLD COMMAND

This Command specifies that current in the motor can be turned OFF=0 or ON=1 during the program execution. The example holds current off in the X-axis motor after a motion is completed.

Line	Command	Parameter 1	Parameter 2	Comments
	X Current Hold	0		

DIR- COMMAND

This command will cause the Direction Register to be 0, and will cause motion to take place in the Counter Clock Wise Direction.

DIR+ COMMAND

This command will cause the Direction Register to be 0, and will cause motion to take place in the Clock Wise Direction.

MAX SPEED COMMAND (1 to 2,500,000)

This Command will determine the speed at which the Motor ramps up to from the Base Speed. To go speeds over 20,000 steps/second, the Speed Limit Command must be used. This is to protect Half-Step Drivers that can be damaged at speeds above 40,000 steps/second.

Line	Command	Parameter 1	Parameter 2	Comments
1	X Max Speed			

POSITION COMMAND (-8,388,607 - +8,388,608)

This Command is used to set the Position Register. When the unit is powered up, the Position Register will initially be 0. When a move takes place, like a Go Relative 1000 in the CW Direction, the Position Register will count up from 0 to 1000 as the move takes place. If a subsequent move takes place, the Position Register will then count up from 1000 to 2000. This Register will always keep track of where the motor was instructed to be. Remember if the motor stalls for some reason, this register will not reflect the true position of the motor. For a close loop system, an encoder must be used. The sample below will home your system, wait for the motor to stop running then sets the position register equal to zero.

Line	Command	Parameter 1	Parameter 2	Comments
1	X Home	1		
2	Wait Motor			
3	X Postion	0		

ACCEL/DECEL COMMAND(1 - 1,000,000)

This defines the acceleration/deceleration (or ramping) of the motor.

Line	Command	Parameter 1	Parameter 2	Comments
1	X Accel/Decel	500		

SPEED LIMIT COMMAND

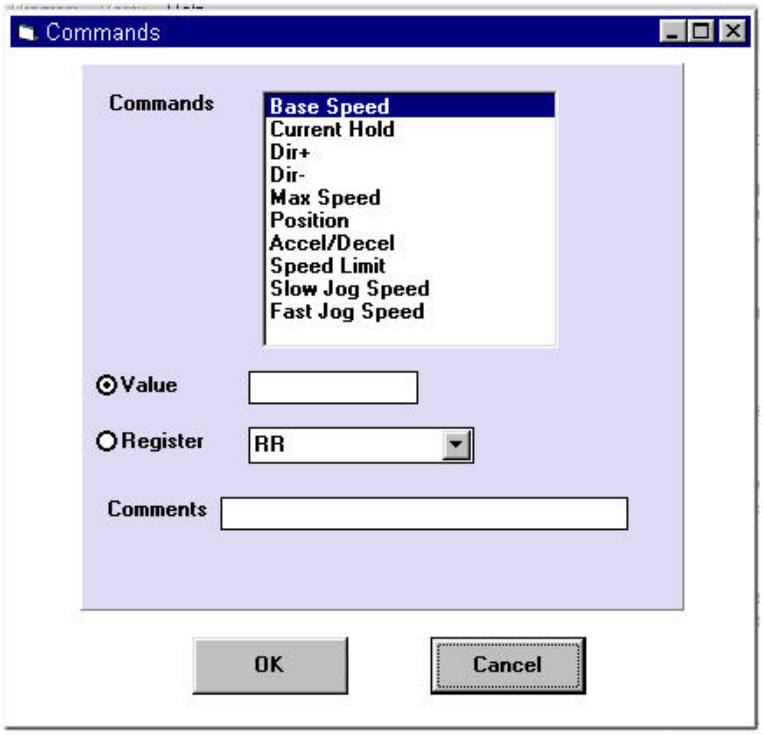
This Command sets the limit for the Base Speed and the Max Speed. The Default value is 20,000. This should always be kept at the highest speed that the motor would ever be run at.

SLOW JOG SPEED COMMAND

This Command presets the Run Speed when the users sets the Slow Jog+ or Fast Jog- to 0VDC. This function is usually used to manually position the motor.

FAST JOG SPEED COMMAND

This Command presets the Run Speed when the users sets the Fast Jog+ or Fast Jog- to 0VDC. This function is usually used to manually position the motor.



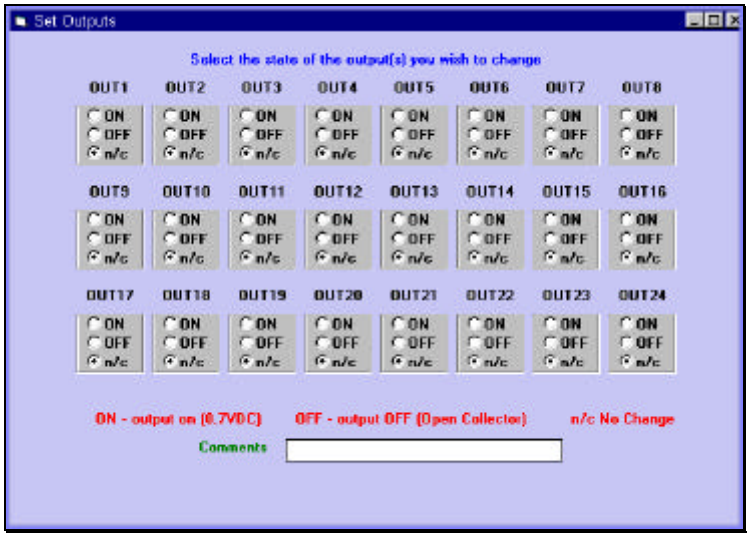
OUTPUT COMMANDS

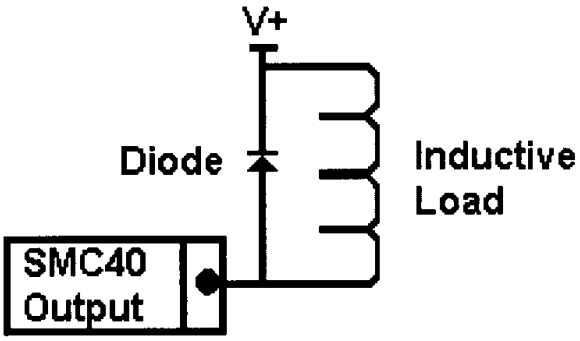
SET OUTPUT COMMAND

This will Activate or Deactivate any of the 24 Outputs. Select only the Outputs that are going to be changed, and leave the other Outputs as Don't Cares(n/c). The SMC40 has 24 programmable open-collector out puts. When an open-collector is set ON, the voltage reading is about 0.7Vdc. When the output is set OFF, the reading is an OPEN. Note: Only five outputs can be turned on per line. The following command line can be repeated to turn more than five outputs when required.

Line	Command	Parameter 1	Parameter 2	Comments
1	Set Output	ON:1,2,10,18	OFF:5,6,12,20	

In this example SET OUTPUT sets outputs 1,2,10, and 18 ON. On the same command line, outputs 5, 6,12,and 20 are set OFF. Note, that a SET OUTPUT command might be used with a conditional IF Statement.





WARNING: Always use a Diode (Motorola 1N4002 or Equivalent) to clamp any Inductive Loads (Relays, Solenoids, etc.) due to fly back voltages. Refer to Specifications for further information.

USER ENTRY COMMANDS

USER ENTRY COMMAND

This Command allows any Commands to be entered that the Software does not directly support. See the Direct Programming Guide discussed in this manual, and the SMC40.WRI Document on the Intelligent Indexer Disk for more information.

Register Manipulation

The Read Value (RV) Register will read a value and temporarily store it. Note that any register comparison that needs to be evaluated may need to be stored into a defined Register (R1- R100) for the Single Axis SMC40's and (R1-R200) for Dual & Triple SmC40's to avoid losing the stored number.

The "IF THEN" Command may compare Register to a known value without having to reassign it but you may lose the RV Register value if any MATH or other Read Values Commands are in your program.

Example: Load the register value RV into R1 and increment register R2 using Math Equations.

Line	Command	Parameter 1	Parameter 2	Comments
1	User Entry	R1=RV		
2	User Entry	R2=R1+1		

PROGRAM 1, 2, 3

Start Program 1, 2, 3

When the Menu Item *Program Start* is executed, it starts only the Main Program. To start Program 1, the Command is **XPA=1**, to start Program 2, the Command is **YPA=1**, and to start Program 3, the Command is **ZPA=1**.

Example: To Start Program 1

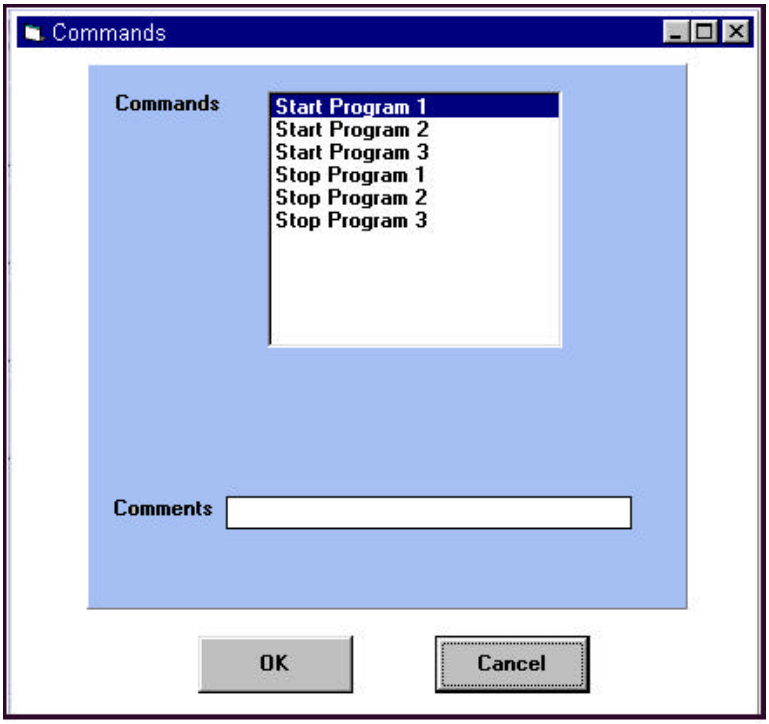
Line	Command	Parameter 1	Parameter 2	Comments
1	Start Program 1			

Stop Program 1, 2, 3 (Stopping a Program)

When the Menu Item *Program Start* is executed, it starts only the Main Program. To stop Program 1, the Command is **XPA=0**, to stop Program 2, the Command is **YPA=0**, and to stop Program 3, the Command is **ZPA=0**.

Example: To start Program 1

Line	Command	Parameter 1	Parameter 2	Comments
1	Stop Program 1			



ENCODER COMMANDS

ENCODER AUTOCORRECT (Enable/ Disable)

This Command will set the Encoder Autocorrection Bit. *A **1** will cause the Autocorrection feature to be enabled, and a **0** will cause it to be disabled*. If the Autocorrection feature is enabled, after the completion of a move, the Encoder Register will be verified to see if it is in the correct position. If it is not, a correction move will be implemented. See Encoder Retries.

Note: Anytime the ER register, EP register, EW register, or the EM register is written to, the Encoder Auto Correction feature is disabled to prevent spontaneous Motor Errors - hence it will need to be re-enabled. *Make sure to use the Encoder Auto Correction command as the last encoder line command to avoid disabling.*

Line	Command	Parameter 1	Parameter 2	Comments
	Enc Auto	1		

ENCODER DELAY

This Command will set the dwell time that the unit will wait before checking the Encoder Register, after the end of the move. This will take out the ringing that might be associated with the mechanics of the system. The range for this is between 1 and 1000 and is in 0.001 seconds [1000 is 1 second]. (*Encoder Delay is used to allow for the motor settling time after a move*.)

Note: Anytime the ER register is written to, the Encoder Stall Detection feature is disabled to prevent spontaneous Motor Errors - hence it will need to be re-enabled.

Line	Command	Parameter 1	Parameter 2	Comments
1	Enc Delay	500		½second

ENCODER POSITION (EP)

This Command will set the starting point of the Encoder Position.

Note: Anytime the EP register is written to, the Stall Detection and Encoder Auto - Correction features are disabled to prevent spontaneous Motor Errors - hence they will need to be re - enabled.

Line	Command	Parameter 1	Parameter 2	Comments
1	Enc Position	0		Position 0 can be the Home Position

ENCODER RETRIES (ER)

This Command will set the number of Encoder Retries or number of tries (0 - 250) when autocorrecting.

Note: Anytime the ER register is written to, the Encoder AutoCorrection feature is disabled to prevent spontaneous Motor Errors - hence it will need to be re - enabled.

Line	Command	Parameter 1	Parameter 2	Comments
1	Enc Retries	5		retry 5 times

ENCODER RATIO (EM)

This Command will set the Ratio between the Motor and the Encoder. The **ENCODER RATIO** register is the conversion factor from 1 motor step to 1 encoder quadrature pulse for the associated indexer - this accounts for any gearing mechanism between the motor and the encoder.

Note: Anytime the ENCODER RATIO (EM) register is written to, the Encoder AutoCorrection feature is disabled to prevent spontaneous Motor Errors - hence it will need to be re - enabled.

{ Negative values represent when the encoder direction is opposite to the motors direction.}

Line	Command	Parameter 1	Parameter 2	Comments
1	Enc Ratio	2.75		

ENCODER WINDOW (EW)

This Command will set the Encoder Window which allows the user to determine the number of steps the motor can be off before correcting. (1- 1,677,215)

Example: The motor missed its stop position by 2 steps. This will mean if the user set the Encoder Window to 3, the encoder will not correct its position.

Note: Anytime the ENCODER WINDOW (EW) register is written to, the Encoder AutoCorrection feature is disabled to prevent spontaneous Motor Errors - hence it will need to be re - enabled.

Line	Command	Parameter 1	Parameter 2	Comments
1	Enc Window	3		

NOTE: Any Advanced Programming questions should be discussed with the factory direct.

SECTION 9 – DIRECT PROGRAMMING

ALPHABETIC QUICK REFERENCE LIST

COMMAND TYPE	COMMAND	DESCRIPTION
MOTION PROFILE	MA	Motor Accel/Decel [X/Y/Z]
	MB	Motor Base Speed [X/Y/Z]
	MM	Motor Max Speed [X/Y/Z]
	MS	Motor Speed Limit [X/Y/Z]
	MF	Motor Fast Jog Speed [X/Y/Z]
	MJ	Motor Slow Jog Speed [X/Y/Z]
	HT	Home Type [X/Y/Z]
	D-	Motor Direction -
	D+	Motor Direction +
	MN	Motor Number [X/Y/Z]
	MP	Motor Position [X/Y/Z]
START/STOP MOTION	GA	Go Absolute [X/Y/Z]
	GH	Go Home [X/Y/Z]
	GR	Go Relative [X/Y/Z]
	GS	Go Slew [X/Y/Z]
	H-	Hard- limit input [X/Y/Z]
	H+	Hard+ limit input [X/Y/Z]
	SA	Stop All Indexers
	SH	Stop Hard [X/Y/Z]
	SI	Soft Limit Input[X/Y/Z]
	SL	Soft Limits Enable [X/Y/Z]
	SS	Stop Soft[X/Y/Z]
MOTION STATUS	MC	Motion Complete [X/Y/Z]
	MD	Motor Direction [X/Y/Z]
	ME	Motor Error Code [X/Y/Z]
	MV	Moving [X/Y/Z]
	WM	Wait Motor [X/Y/Z]
	AB	At Base Speed Bit [X/Y/Z]
	AD	At ramping Down Bit [X/Y/Z]
	AF	At Fast Jog Bit [X/Y/Z]
	AJ	At Slow Jog Bit [X/Y/Z]
	AM	At Maximum Speed Bit [X/Y/Z]
	AR	At auto-correct Retry Bit [X/Y/Z]
	AS	At Speed [X/Y/Z]
	AU	At ramping Up Bit [X/Y/Z]
INPUT/OUTPUT	O	Outputs (1-24)
	OE	Outputs External Module (O25 - O48)
	OR	Output Register
	I	Inputs (1-96)
	IO	Bi-directional Input/Output (IO1- IO24)
	IR	Input Register (I1 -I24)
EXPANSION	IA	Input A module (I25 -I48)
	IB	Input B module (I49 -I72)
	IC	Input C module (I73 -I96)
	TA	Thumbwheel A Module
	TB	Thumbwheel B Module
	TC	Thumbwheel C Module
PROGRAMMING	PA	Program Autostart [X/Y/Z]
	PI	Program Insert [X/Y/Z]
	PC	Program Clear [X/Y/Z]
	PD	Program Delete [X/Y/Z]
	PE	Program Edit [X/Y/Z]
	PM	Program Memory available
	PQ	Programs Quit All
	PR	Program Run [X/Y/Z]
	PS	Program Start [X/Y/Z]
	PT	Program Trace [X/Y/Z]
	NO	No Operation
	PP	Program Pause
	PW	Program Window Size
	PL	Program List [X/Y/Z]
	PN	Program Instruction Number [X/Y/Z]
	PB	Program Brief Listing
TERMINAL	RK	Read Key From Terminal
	RTS	RTS to Send
	RV	Read Value
	WK	Write Key
	WT	Write Text
	WV	Write Value
	CTS	Clear To Send
BRANCHING & LOOPING	BB	Branch Breakpoint
	BE	Branch End of Program
	BF	Branch if False
	BG	Branch Goto
	BL	Branch Loop
	BQ	Branch Quit
	BR	Branch Return
	BS	Branch Subroutine
	BT	Branch if True
	FL	For Loop
	IF	IF Condition
REGISTERS & BITS	B	Bits (1-100)
	R#	Indirect Register
	R	Registers (1-100)
	RS	Read Value Scale Factor
	B#	Indirect Bit
MATH	B?	Pattern Match
	B-	result Bitwise NOT
	B%	result Bitwise XOR
	B+	result Bitwise OR
	B*	result Bitwise AND
	GT	Greater than Flag
	LT	Less than Flag
	RP	Positive Flag
	R/	Result Divide
	R?	Result compare
	RF	Result fraction
	RI	Result Integer
	R+	Result add
	R*	Result multiply
	R%	Result modulus
	RK	Read Key
	RN	Negative Flag
	RO	Overflow Flag
	R-	Result Subtract
	RR	Result Register
	RT	Square Root
	RZ	Result Zero Flag
	EQ	Equals Flag

Miscellaneous	VN	Version Number
	ID	Identify
	HC	High speed counter
	HP	High Priority
	MH	Motor Hold Current [X/Y/Z]
	@	unit select
	@.	unit select all
	#	indirect register
	!	Error
	AC	Auto Cursor
	AE	Auto Error
	AL	Auto Linefeed
	CR	Cold Reset
	DL	Delay
	DP	Decimal Place
	EC	Error Code
	FD	Full Duplex
	JF	Jog Fast Input bit/configuration
	LB	Label
	WD	Wait Delay [X/Y/Z]
	WR	Wait Read Value
	WS	Wait value Scale Factor
	WW	Wait Write
	J-	Jog- Input Configuration [X/Y/Z]
	J+	Jog+ Input Configuration [X/Y/Z]
	JF	Jog Fast Input configuration [X/Y/Z]
	FD	Full Duplex
	UP	Unlock Program
	LP	Lock Program
ENCODER	EA	Encoder Auto-correction [X/Y/Z]
	ED	Encoder Delay Register [X/Y/Z]
	EM	Encoder to Motor ratio [X/Y/Z]
	EP	Encoder Position [X/Y/Z]
	ER	Encoder Retries [X/Y/Z]
	EW	Encoder Window (auto-correction)
LIMIT INPUTS	H-	Hard - limit input
	H+	Hard + limit input
	HI	Home limit Input
	SI	Soft Limit Input
	SL	Soft Limits Enable

Note: Refer to Chapter 13 for Command Set

TALKING WITH A TERMINAL (TT1R2-1)

WARNING: DO NOT CONNECT THE PC AND HANDHELD TO SMC40 AT THE SAME TIME. DOWNLOAD PROGRAM AND THEN TEST WITH HANDHELD AFTER DISCONNECTING FROM PC.

Start Communications

Press CTRL and SHIFT simultaneously and while holding down on those two buttons press F1. This will now display the terminal parameters. Press F2 to scroll through the parameters. Set the Address and the Baud Rate on the SMC40 Indexer switches first (Pg 24), then set the Baud Rate on the Terminal. Press F1 to scroll through the different settings available. The SMC40 Indexer will communicate by sending the characters back, so be sure to have the Terminal with the Echo = Disable set.. Data Bit = 7, Parity = Space, Display PE = Disable, Repeat = Fast, Handshake = Disabled, and Self Test = Disabled.

Note: Always set the Terminal to Full Duplex On (FD1) when programming commands. Standard Communications Settings are 8 Bit with No Parity. The TT1R2-1 will accomplish this buy setting the Data Bit = 7 and the Parity = Space. AL = 1 will allow the commands to be entered one per line as shown in the example below.

Connect the Terminal
Type ***@0*** (where *0* is the axis you dialed in SW3)
Notice that the Status LED should start to blink rapidly on the indexer.

Type ***ID***
This should return the version number of the chip - like “SMC40”

Example of Entering a Short Program

```
@0              ‘select axis 0
ID              ‘chip type
        “SMC40”
VN              ‘version number
        “0”
AL1             ‘Auto -Linefeed On
FD1             ‘Full Duplex

PI1LB TOP       ‘Label Top
PI2XMH1         ‘Current Motor Hold/ Power Off
PI3XMM4000      ‘Program Insert at line 1, Max Speed = 4000
PI4XMB500       ‘Base Speed = 500
PI5XMN400       ‘Distance = 400
PI6XGR          ‘Go Relative
PI7XWM          ‘Wait Motor until motion is complete
PI8BG TOP       ‘Branch Goto Label Top
PS              ‘Program Start
```

Motion should start at this point and your stored program will begin to execute.
Note: To stop this stored program inserted you need to type @0 then Enter.

Example of a Short Program on the X Program

```
XPI1XMH1        ‘Current Motor Hold/ Power Off
XPI3XMM4000     ‘Program Insert at line 1, Max Speed = 4000
XPI4XMB500      ‘Base Speed = 500
XPI5XMN800      ‘Distance = 800
XPI6XGR         ‘Go Relative
PS              ‘Program Start
```

Note: To stop this stored program inserted you need to type @0 then Enter.

DIRECT PROGRAMMING QUICK START

Direct Programming of the Driver Pack can be done with any programming language that can communicate via the serial port. The complete Command Dictionary is located in the directory that the software was installed in. Section 2

A sample of Commands

@0 The @ (at symbol) is used to select which unit is to be communicated with. This unit, in this case zero, must be the setting on the intended unit.

ID Entering this command will return the indexer type.
>SMC40

FD1 The full duplex command…

XMN100 The motor number command enters the distance that the motor will travel when indexed by a go relative command.

XMN Entering just MN alone will return the motor number. The default value is zero.
>100

XGR The Go Relative command tells the motor to index the MN number of steps.

PI1XGR The PI is the Program Insert command. Here a go relative command is inserted in line one. If the program is already store in the memory, the PI1 command here will push the whole program down a line.

PS Program Start is used to run the program stored in the memory.

PW2 Program Window set the number of instructions to be displayed when the Program List command is used.

PL1 The Program List command list the program starting at the line number proceeding the PL.

>1 GR
>2 BE The Branch End is displayed when there is no more instructions in the Indexer's memory.

To List a Program
PL0 'Program list starting at line 0
"XMH=1"
To Edit a Line
PE4 XMN400 'Edit Line 4 to read X Program Motor Number 400

To Delete a Line
PD2 'Delete line 2

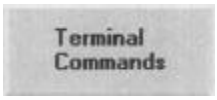
To Check Memory in a Program
PM

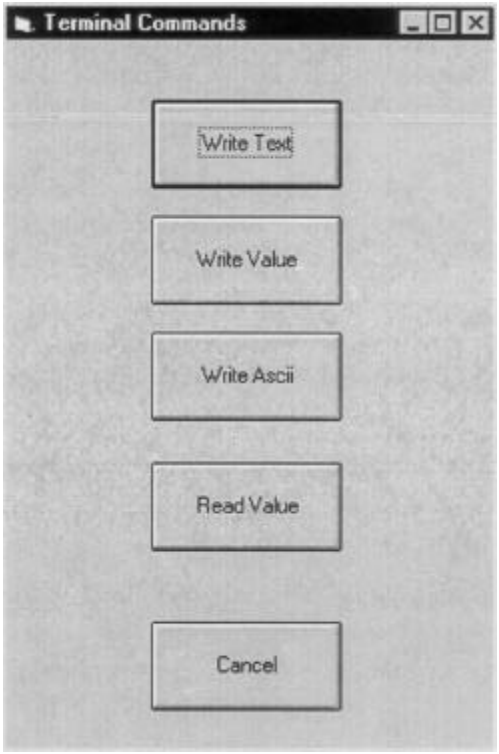
Example **To Stop** a Direct Command In Execution
@0
XGS 'X Program Go Slew - Motion Activated
SA '**Stop All Indexers**

TERMINAL COMMANDS (SMTNR2-1 AND TT1R2-1)

WARNING: DO NOT CONNECT THE PC AND HANDHELD TO SMC40 AT THE SAME TIME.
DOWNLOAD PROGRAM AND THEN TEST WITH HANDHELD AFTER DISCONNECTING FROM PC.

The terminal commands are used in conjunction with the Panel Mount Terminal (SMTNR2-1) or the Handheld Terminal (TT1R2-1). The commands allow an SMC40 program to prompt an operator per operational information needed to begin a cycle of operation. The instructions are Write Text , Write Value, Write ASCII, and Read Value. Write Text allows the user to transmit a character string to the handheld terminal, which produces a character string on the terminal display. Write Value allows the user to transmit a numerical value to the terminal display. Write ASCII allows the user to transmit an ASCII character/code to the terminal display . The Read Value command waits for a numerical value to be entered- a number terminated by a carriage return from the handheld terminal – and stores it in the RV register.





WARNING: Do not connect PC Serial Port to the SMC40 Indexer at the same time the Handle Terminal is connected. Damage may occur to the serial port or handheld.

It is best to write your program, download the program via the serial port, enable the autostart from the Program Menu, then power the indexer down.

The RS232 cable must now be disconnected from the SMC40 and the handheld terminal connector should be plugged into the telephone connector J1. The final step now is to power up the SMC40 Indexer.

The sample program below demonstrates the terminal commands:
Lines 1 and 2 are used to clear the terminal display and home the cursur to the top left corner of the display using the ESC E code. Line 3 is a text string that will be displayed on the LCD Screen. Line 4 is the command that drops the cursur to column one of the next line. Line 5 will read a numerical value the user enters when excecuting the program. Line 6 is a math application that will use the Register Value(RV), multiply it by 400 and set the x-axis motor number equal to the result. Line 7 will index the motor the value stored in the Result Register. Line 8 is a branch quit command required at the end of all programs.

Line	Command	Parameter 1	Parameter 2	Comments
1	Write Ascii	27		ESC
2	Write Ascii	69		ESC E means clear display & home
3	Write Text	Anaheim Automation		Text String
4	Write Ascii	13		Cursor Left to Column 1
5	Read Value			RV Register is loaded
6	Math	xmn=rv*400		AXIS X Motor Number = Register Value * 400
7	XGo Relative	RR		Result Register = RV*400
8	Branch Quit			

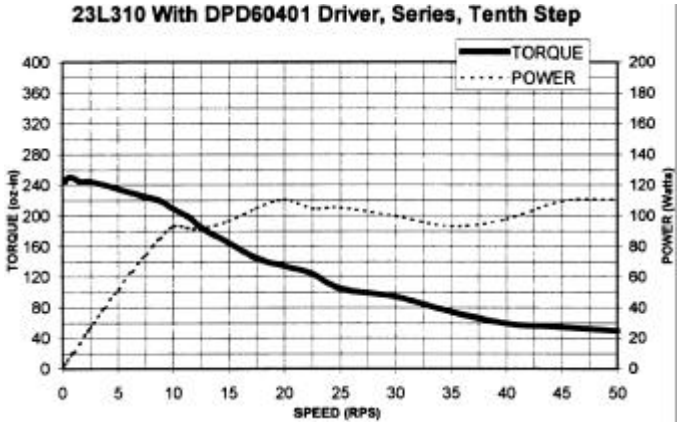
SECTION 10 - SPECIFICATIONS

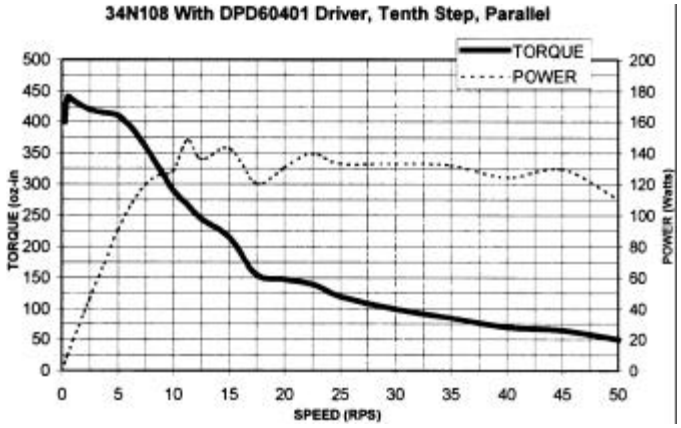
Power Requirements 7A Fuse	105/ 125 VAC, 50/60Hz *(Standard Package)* 7 Amp, 5mm, Fast blow type Littlefuse: 217005 210/ 250VAC, 50/60 Hz *(X250 Version)* 7 Amp, 3AG, Fast blow type
Power Requirements for PCL Versions	9 –12 VAC 50/60 Hz +5VDC @ 1.5 Amp +12 – 16 VDC UNREG @ 1.5 Amp *NOTE: Any External Electric Component current (Amp) consumption should be taken into account..*
Operating Temperature	0 to 60 degree C
Driver Output Current Rating	1 to 7.0 Amp per phase *(DPD72401,DPK Series)* 1 to 6.0 Amp per phase *(DPD60401)*
Control Inputs	TTL-CMOS Compatible Logic "0": 0 to 0.8 VDC Logic "1": 3.5 to 5.DC
Pulse Output Range	1 to 2,500,000
Outputs (CLK, DIR, PWR): TTL-CMOS compatible Logic "0": Logic "1":	 0 to 0.32 VDC, 4 Ma 4.3 to 5.1 VDC, 4 Ma
Baud Rate:	50 to 38400 BAUD
RS422 Input Logic "0" Logic "1" sensitivity	 -2 to -10 VDC, 1.5 Ma 2 to 10 VDC, -2.5 Ma 200 mV
RS422 Output: Voltage Output High Voltage Output Low	 2.5 VDC min, 20 Ma 0.5 VDC max, 20 Ma
RS232 Input Logic "0" Logic "1" sensitivity	 2 to 10 VDC, 1.5 mA -2 to -10 VDC, -2.5 mA 200 mV
RS232 Output Logic "0" Logic "1"	 0.5 VDC max, 20 mA 2.5 VDC min, 20 mA
Encoder	Quadrature only
Inputs TTL-CMOS Compatible Logic "0": Logic "1":	5 VDC @ 10ma Max 0 to 0.32 VDC, 4 Ma 4.3 to 5.1 VDC, 4 Ma
Data Format: Half-Duplex, 1 start bit, 8 data bits, no parity	
Outputs (24 programmable I/O): Maximum voltage: Current sink: **NOTE:** Always use a Diode (Motorola 1N4002 or Equivalent) to clamp any Inductive Loads (Relays, Solenoids, etc.) due to fly back voltages. +5VDC Output (PCL401 ,DPD72401, and DPD60401) +5VDC Output (PCL402, PCL403, DPK72402,and DPK7403)	Open Collector Type 40 VDC 500 ma (total, for outputs 1 - 8) 500 ma (total, for outputs 9 - 16) 500 ma (total, for outputs 17 - 24) Maximum Current Sourcing 0.75 AMPS Maximum Current Sourcing 0.75 AMPS

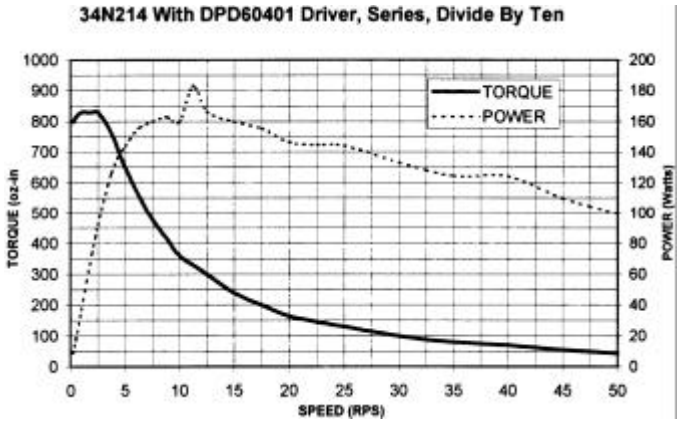
MODULE SPECIFICATIONS

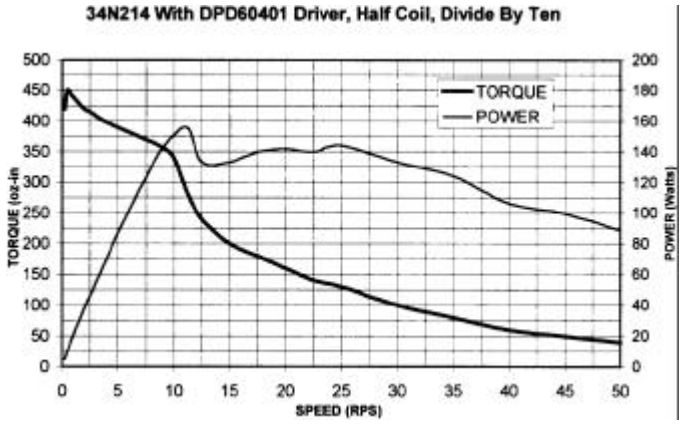
SMC40 to Input Module Cable Maximum Length	5 feet
Inputs Internal pull-up Resistor to 5V	1k ohms
Mounting:	DIN Rail
Additional Expansion Modules Connector (Refer to Section 4)	J2

PERFORMANCE CURVES –DPD60401

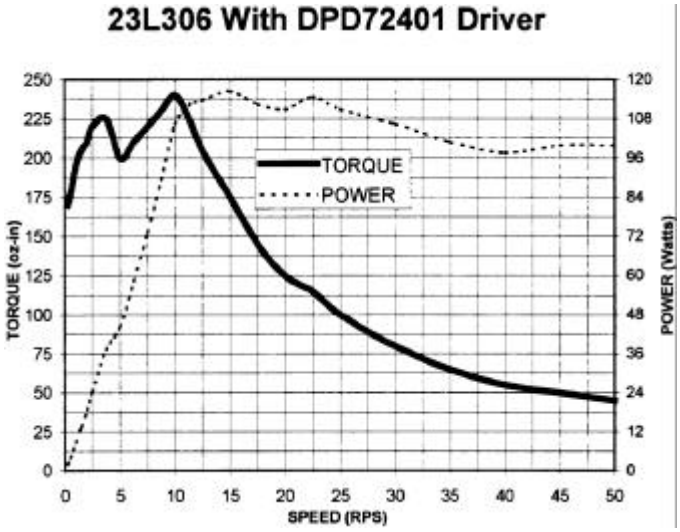


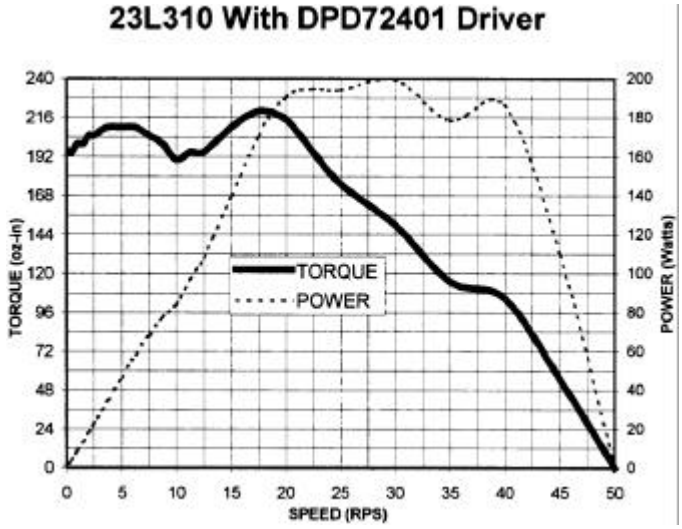


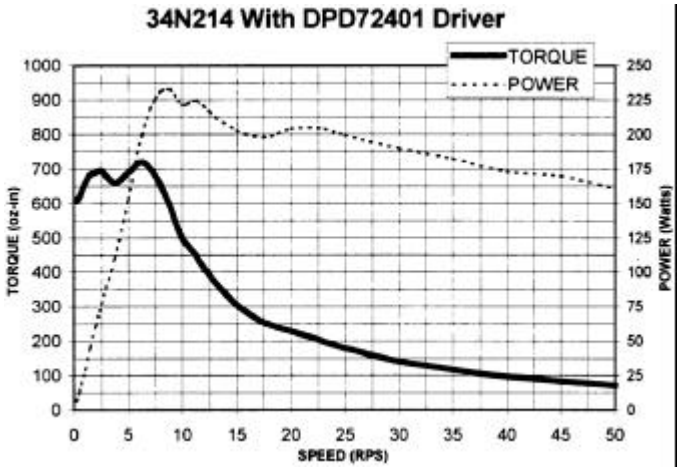




PERFORMANCE CURVES - DPD72401 SERIES







SECTION 11 - TROUBLESHOOTING

MY COMPUTER WON'T TALK TO THE SMC40

When using a computer to communicate to the SMC40, you must use Anaheim Automation communications program such as Hyper Terminal, or a programming language such as BASIC or C. The communication parameters must be set up correctly for 8 Data Bits, No Parity Bits, and 1 Stop Bit. The Baud Rate can be set up for rates between 50 Hz to 38400 (50, 300, 1200, 2400, 9600, 19200, 38400).

Check the Baud Rate setting on the yellow and black rotary switch on the side of the unit. This setting must match to the baud rate you specified (see Table 6). The Request To Send (RTS) signal is supported by some software and not others. Refer to the software manual to see what you need, or just use trial and error - it will not damage anything. Generally use RTS on, or the first 8 positions of the dip switch. Once the communications are set up you finally need to talk to the SMC40. This is done by selecting the axis you want to talk to. **Position the Axis Select for the desired axis number for your unit.** If you chose axis 0 for example, the rotary dip switch would be pointing to 0. To communicate to this axis, type ID then ↵Enter.

MOTOR IS STALLING

Check the kick current pot setting on the driver, be sure it is set to motor's current rating. Check the wiring of the motor to the driver. Different step motors have different performances, some may be able to start faster than other. If your motor is stalling at start up or during ramping, then lower the base and max speed and increase the acceleration time (BASE SPEED 500, MAX SPEED 1000, AND ACCEL/ DECEL 10000 are settings that should be good for testing most motors).

THERE IS NO POWER TO THE UNIT

If the fan of the DPD72401 is not running, check the fuse in the fuse tray in the power connector terminal (use 7 amp fast blow fuse only). If the fuse continues to blow call factory for assistance.

STATUS LED TURNS OFF WHEN I PLUG INTO THE UNIT

The communication switches on the side of the unit are most likely incorrect.

SECTION 12 – ERROR CODES

SMC40 (V0.957)

1. WarningCodes
2. Syntax_UnexpectedPrefix
3. Syntax_CarriageReturnExpected
4. Syntax_RegisterPrefixExpected
5. Syntax_RegisterOrNumberExpected
6. Syntax_BitOrBinaryExpected
7. Syntax_NumericalChannelExpected
8. Syntax_BitPrefixExpected
9. Syntax_TextStringExpected
10. Syntax_InstructionNumberExpected
11. Syntax_NegativeNumberLiteralExpected
12. Syntax_UnknownRegister
13. Syntax_UnknownBit
14. Syntax_BinaryLiteralExpected
15. Syntax_UnknownPatternCharacter
16. Syntax_NumericalIndexExpected
17. Syntax_BitPatternOverflow
18. Syntax_UnexpectedNumeric
19. Syntax_UnexpectedSymbol
20. Syntax_UnknownInstruction
21. Syntax_InvalidLabel
22. Syntax_LabelExpected
23. Syntax_InvalidInstructionNumber
24. Range_InputChannelValue
25. Range_OutputChannelValue
26. Range_BidirectionalChannelValue
27. Range_RegisterIndexValue
28. Range_BitIndexValue
29. Range_ProgramTextWindowValue
30. Range_InstructionNumber
31. Range_DelayValue
32. Range_CounterValue
33. Range_ThumbWheelDigitValue
34. Range_DecimalPlacesValue
35. Range_WriteKeyValue
36. Range_EncoderDelayValue
37. Range_EncoderPositionValue
38. Range_AbsolutePositionValue
39. Range_EncoderRetriesValue
40. Range_EncoderWindowValue
41. Range_MotorNumberValue
42. Range_BaseSpeedValueLimits
43. Range_MaximumSpeedValueLimits
44. Range_SpeedLimitValue
45. Range_AccelDecelValue
46. Range_JogFastSpeedValueLimits
47. Range_JogSlowSpeedValueLimits
48. Range_JogInputValue
49. Range_HomeTypeValue
50. Range_MotorPositionValue
51. Range_InstructionNotFound
52. Range_OutputInternalRegisterValue
53. Range_AutoErrorValue
54. Range_OutputExternalRegisterValue
55. Range_ResultBinaryValue
56. Range_ProgramLockKey
57. Warning_ErrorQueueOverflow
58. MotorErrors
59. XMotor_AutoCorrectionFailure
60. XMotor_EncoderTrackingOverflow
61. XMotor_HardLimits
62. XMotor_JoggingHardLimits
63. XMotor_ErrorOverflow
64. YMotor_AutoCorrectionFailure
65. YMotor_EncoderTrackingOverflow
66. YMotor_HardLimits
67. YMotor_JoggingHardLimits
68. YMotor_ErrorOverflow
69. ZMotor_AutoCorrectionFailure
70. ZMotor_EncoderTrackingOverflow
71. ZMotor_HardLimits
72. ZMotor_JoggingHardLimits
73. ZMotor_ErrorOverflow
74. TerminationErrors
75. Execution_IndexerNotPresent
76. Execution_XIndexerMotionInProgress
77. Execution_YIndexerMotionInProgress
78. Execution_ZIndexerMotionInProgress
79. Execution_ReadOnlyRegister
80. Execution_ReadOnlyBit
81. Execution_ProgrammableOnlyInstruction
82. Execution_ProgramAlreadyRunning
83. Execution_ProgramNotRunning
84. Execution_ProgramNotPaused
85. Execution_LabelNotFound
86. Execution_InstructionNotFound
87. Execution_BranchNumberValueRange
88. Execution_GosubLevelOverflow
89. Execution_BranchReturnWithoutGosub
90. Execution_NoForLoopInstruction
91. Execution_NoThumbWheelModule
92. Execution_BranchSubroutineWithinMESubroutine
93. Math_DivideByZero
94. Communications_ReceiveBufferOverflow
95. Communications_InstructionBufferOverflow
96. Communications_TransmitBufferOverflow
97. Communications_BreakCommandReset
98. Programming_MemoryFull
99. Programming_CannotDeleteBEInstruction
100. Programming_CannotInsertBEInstruction
101. Programming_AssignmentValueRequired
102. Programming_NeedCLEARConfirmation
103. Programming_CannotProgramProgrammingCommand
104. Programming_ExecuteOnlyInstruction
105. Programming_ExecutionInProgress
106. Programming_ControllerProgramInstructions
107. Programming_MemoryAlreadyLocked
108. Programming_MemoryAlreadyUnlocked
109. Programming_UnableToUnlockMemory
110. Programming_MemoryLocked
111. Programming_AutoStartMemoryLocked
112. FatalErrors

SECTION 12 - error codes (Continued)

113. Internal_MissingControllerBEInstructions
114. Internal_MissingXIndexerBEInstructions
115. Internal_MissingYIndexerBEInstructions
116. Internal_MissingZIndexerBEInstructions
117. Internal_CorruptedControllerProgram
118. Internal_CorruptedXIndexerProgram
119. Internal_CorruptedYIndexerProgram
120. Internal_CorruptedZIndexerProgram
121. Internal_InvalidProgramMemory
122. Internal_ErrorStackOverflow
123. Internal_NVMemoryNotInitialized
124. Internal_NVProgramChecksumFailure
125. Internal_ProgramTerminatorDeletion

BLINKING ERROR CODES (INDEXER)

The SMC40 Indexer will flash an error code (LED on Indexer Side) . Their will be a slow blink followed by a fast blink when an error occurs. The number of times that flash slow is first and the fast flash is second.

Example: Five Slow (5) flashes and Nine Fast (9) flashes, this would indicate that the SMC40 has Error 59 occurring which means you have an Auto Correction Failure on Axis X..

Example: Nine Slow (9) flashes and One Fast (1) flash, this would indicate that the SMC40 has Error 91 occurring. Error Code 91 means you have no thumbwheel module connected but you used a command in your line code.

USING HYPER TERMINAL

SETTING PARAMETERS

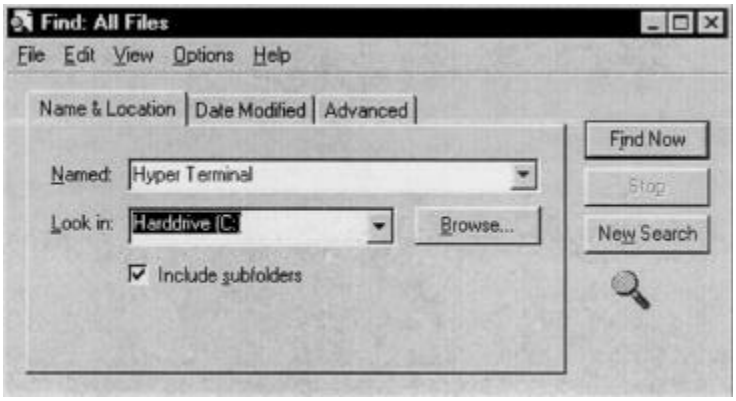
Windows 95 or 98 Is recommended when running this program.

Exit the SMC40 Software before attempting to run Hyper Terminal Program.

1. The first thing you need to do is to Find the Hyper Terminal EXE. At the bottom left corner of your Windows screen is the Start button. Click this icon to locate the Find All Files folder.

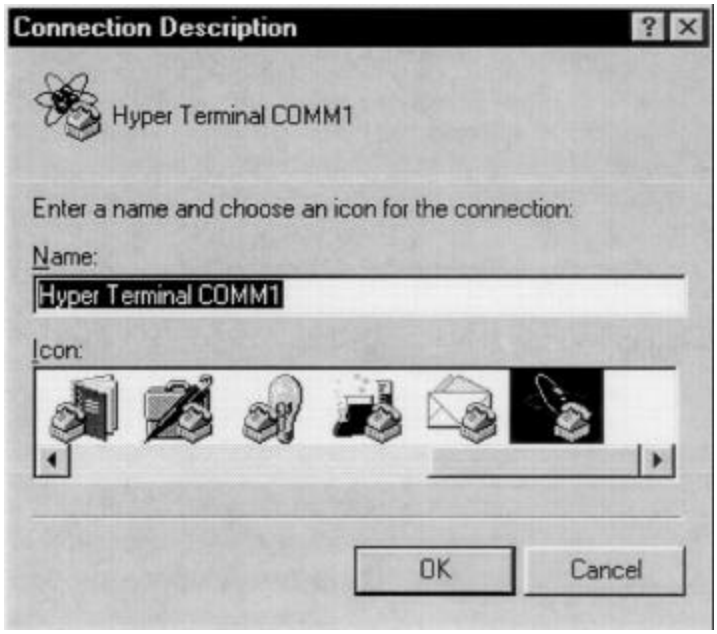
2. Click on this Find Icon . 

3. Type in under the Named: ***Hyper Terminal*** . Make sure Look in: ***Harddrive (C:)*** is displayed.



4 Once located, the Hyper Terminal will need to designate a Name and Icon. Name HyperTerminal COMM1 or COMM2 depending on the available Comport.

Warning*: Press the Cancel Button if your computer tries to select a modem communications. This setup is not meant for modem communication, it is meant to establish Port Communications with the SMC40 Controller.*

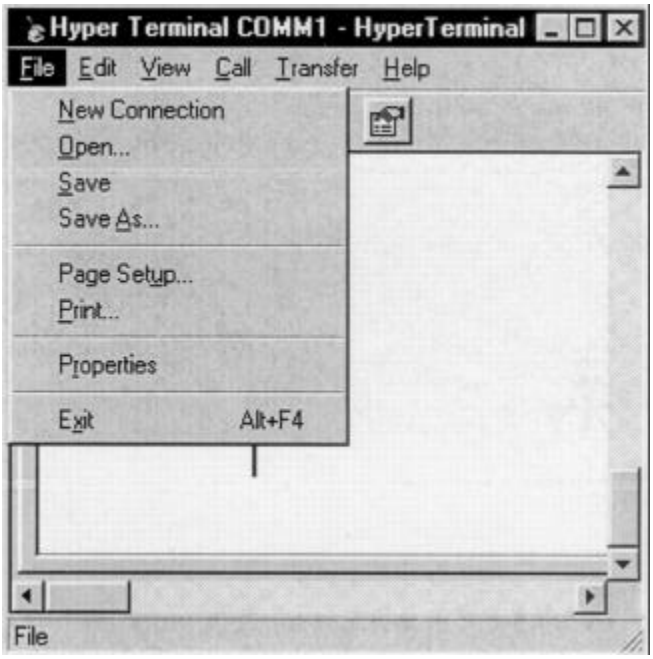


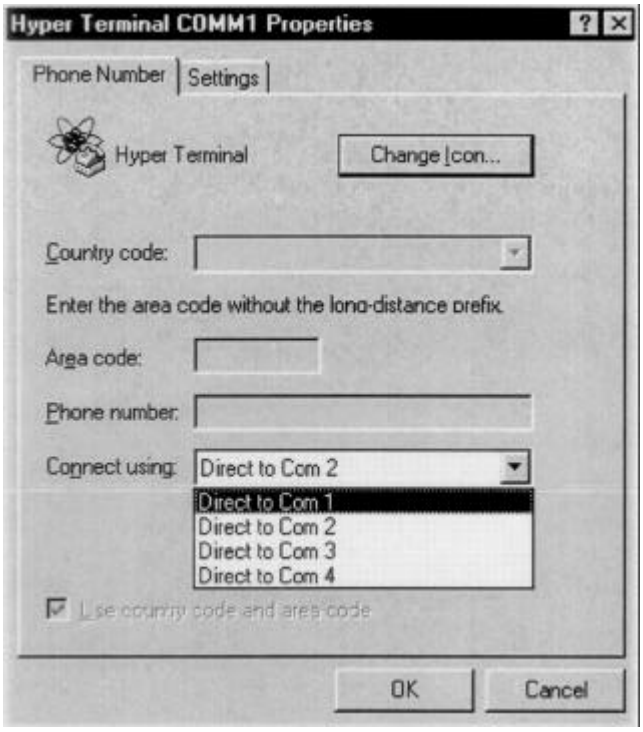
SETTING PROPERTIES

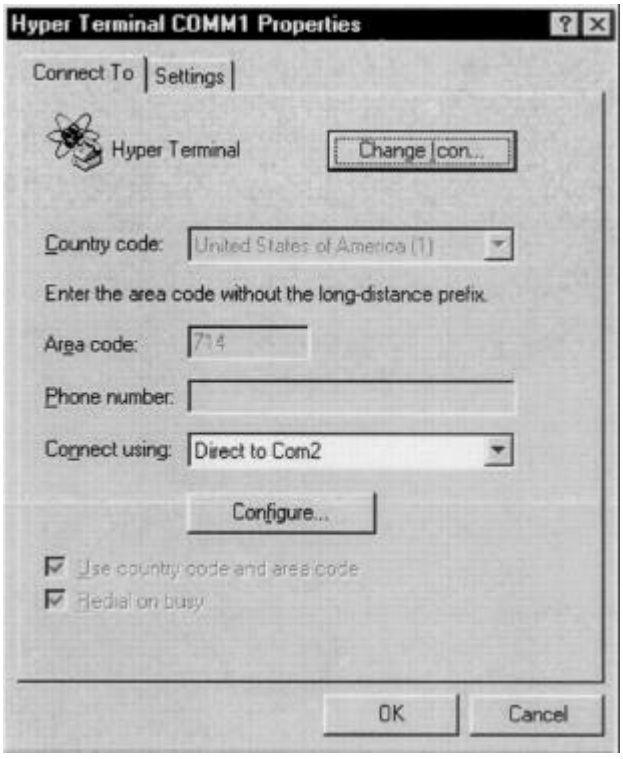
1. Under the File Menu select Properties.

2. Select any used Comport.
   Direct to Com 2

3. Click on the Configure Button, it will allow the Port Settings to be Configured.
   Baud = 9600
   Data Bits = 8
   Parity = None
   Stop Bits = 1
   Flow Control: Xon / Xoff



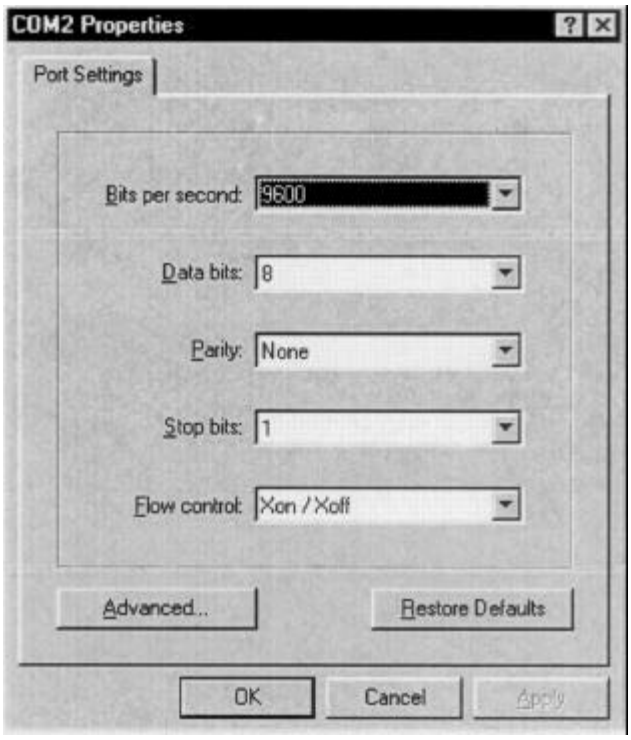


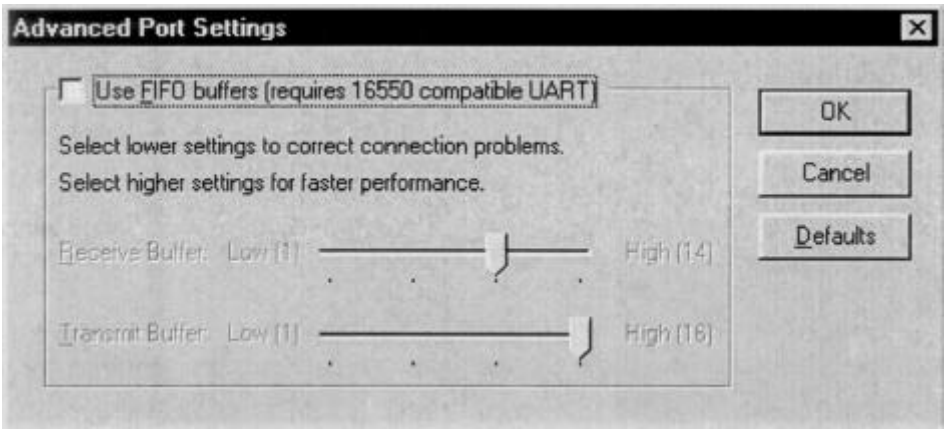


SETTING PROPERTIES

Baud = 9600
Data Bits = 8
Parity = None
Stop Bits = 1
Flow Control: Xon / Xoff

4. Click on the Advance Button.
The FIFO should be unselected





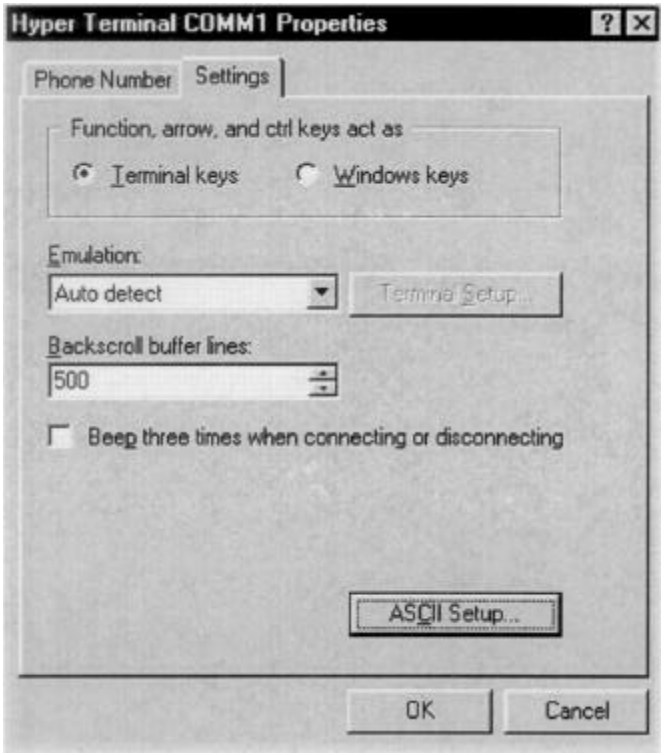
TERMINAL SETTINGS

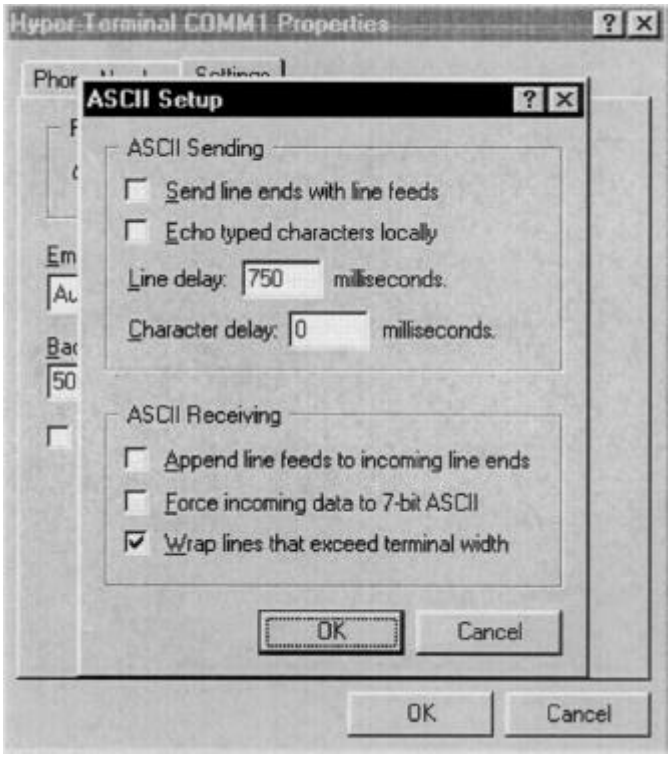
1. Under the Settings Section use the following settings:

   Select Terminal keys (On)
   Emulation: Auto Detect
   Backscroll buffer lines: 500

2. Click on ASCII Setup:

   Only requirement is have Line Delay = 750
   Select Wrap Lines that exceed terminal width (ô)



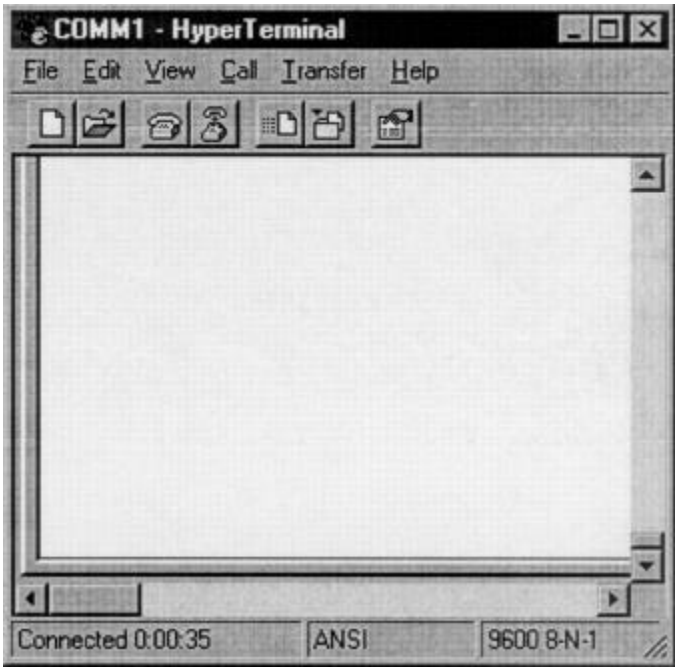


CLEARING ERROR CODES ON THE SMC40

Program Checksum Failure (EC 124 on single axis SMC40's and EC 125 on Dual or Triple SMC40 Controllers)

The very first screen will have a blank screen as shown to your right. The user will need to type in the instructions Procedures as written below:

*Note: You may or may **NOT** see the first few commands. Type in **FD=1** so that text can be displayed (full duplex). The program will then begin showing instructions on the screen.*



Procedure	**Comment**
1. @0	Begins Communication
2. @0 ID	Board should return with it's name and number of axes
3. @0 PA0	Turns off Auto Start Mode
4. @0 PQ	Stops all Programs
5. @0 PC RESET	Resets the SMC40
6. @0 PC CLEAR	Makes sure all programs are cleared out of memory
7. @0 EC0	Clears out the Error Code
8. Board Power	Press the Board Reset button. LED should be on Solid
9. @0	Resume Communication. Error is now cleared.

Most Errors can be cleared using the Command EC which clears the Errors.

@0 EC0 <Enter> **Enter EC0 and press Enter**

GETTING STARTED

The user will need to type in the following instructions:

*Note: You will **NOT** see the first few commands. Type them in as instructed below and the program will then begin showing instructions on the screen.*

1. Type in ***@0*** then ENTER it.

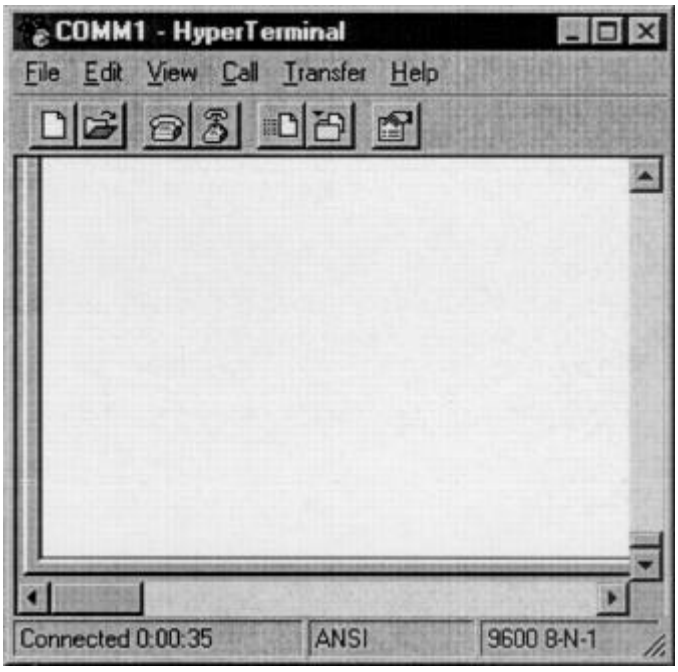
2. Type in ***id*** then ENTER it.

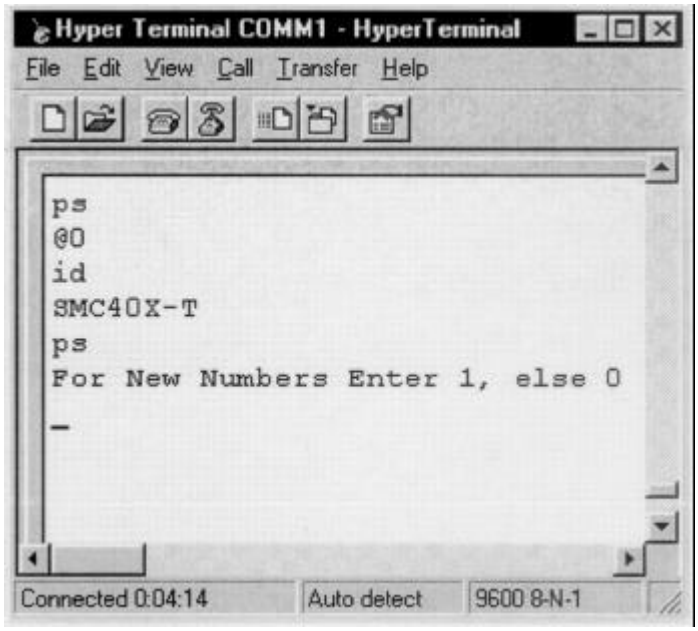
3. SMC40X-T should now be on the screen.

4. Type in ***ps,*** then Press Enter.

5. The Stored Program should have Started.
(The command PA will start a stored program if a downloaded and auto stored program is already loaded.)

6. ***If you have problems communicating, power the controller down. Exit Hyper Terminal and Start over from the beginning. Power up the Controller and Start Hyper Terminal.***





SECTION 13 - GLOSSARY

Absolute Mode
: A positioning coordinate reference wherein all positions are specified relative to some reference, or "home" position. This is different from relative, or incremental programming, where distances are specified relative to the current position.

Baud Rate
: A term used frequently in serial data communications but often is misunderstood. 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.

Break Signal
: A break is often used to signal a remote computer to stop transmission. Typically a Break Signal is produced by holding the data terminal equipment (DTE) transmit data (TXD) low for some time significantly longer than the time it takes to send a word.

Daisychain/ MultiDrop
: A term used to describe the linking of several RS422/RS232C devices in a sequence such that a single data stream flows through one device and on to the next. Daisy-chained devices usually are distinguished by device addresses, which serve to indicate the desired destination for data in the stream.

Debug
: A term used to define refinements to a system or program that remove undesirable effects.

EEPROM
: Electrically Erasable Programmable Read-Only Memory. A memory device frequently used with microprocessors, that can be erased and reprogrammed with- out removing it from the circuit. This creates non-volatile memory; i.e. memory that won't be lost if the power is turned off.

Hard Limit Switch
: A switch (i.e. photo, Hall-effect or mechanical) that defines the absolute limit of motion in a particular direction. It may be used to prevent collisions or out-of-bounds conditions.

Home
: A reference position in a motion control system, usually derived from a mechanical datum. Often designated as the "zero" position.

Home Limit Switch
: The switch used to establish the reference position designated "home".

Program Counter
: The Program Counter is used by the processor to point to the address of the next instruction to be executed by the processor in the stored program mode.

Mask
: A binary-weighted number that conceals some or all of the bits in an associated memory address or register.

Relative Mode
: A coordinate system where positions or distances are specified relative to the current position.

Stack

A register or buffer in memory that uses Last-In-First-Out (LIFO) entry and retrieval of data.

Soft Limit Switch

This switch is used exclusively in Homing Mode 0 (zero).If positioned properly for the appropriate parameters, it causes the motor to ramp down to the Base Speed before encountering the Home Limit Switch. This ensures that the motor speed is within the start-stop region.

Start-Stop Region

That range of speeds in which a step motor can start, stop, or reverse direction in synchronism with the external pulse signal.

APPENDIX A – ASCII CODE TABLE

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15
0	*0	**	**	**	**	**	**	*1	*2	**	*3	**	*4	*5	**	**
16	**	**	**	**	**	**	**	**	**	**	**	*6	**	**	**	**
32		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
48	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
64	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
92	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
112	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	**

** = special character; its representation is device dependent
*0 = ASCII NULL character code
*1 = Bell character code
*2 = Backspace character code
*3 = Linefeed character code
*4 = Formfeed character code
*5 = Carriage Return character code
*6 = Escape character code