D-404 USERS MANUAL

1 Progress Drive, Dover, NH 03820-5449 USA, (603) 742-3330, Fax (603) 742-9080, www.eadmotors.com

Table Of Contents

Table Of Figures

Introduction

Thank you for choosing the D-404 Stepper Motor Driver from EAD Motors. This unit is designed to provide years of reliable, accurate and cost-effective motion control. The D-404 is backed by over 60 years of manufacturing excellence and a commitment to quality and support that guarantees your satisfaction.

This manual will assist you in optimizing the performance of your system. Its' purpose is to provide access to information that will facilitate a reliable and trouble-free installation.

In addition to the Users Manual, EADdemo setup software can be found free on the EADmotors website. This programming tool is an easy to use, menu driven utility file with on-line help screens. Used together, the Users Manual and EADDemo will enable you to quickly take advantage of the advanced programming features and system capabilities inherent in the system design.

Although the D-404 and supporting documentation were designed to simplify the installation and on-going operation of your equipment, we recognize that the integration of motion control often requires answers to many complex issues. Please feel free to take advantage of our technical expertise in this area by calling one of our support personnel to discuss your application. Technical support options are available at the end of this manual.

Product Overview

The **D-404** combines efficient ***bi-polar chopper Driver*** circuitry with an integrated micro-controller on a single, heat sink mounted board, to operate small to mid-size stepping motors. It is designed for low cost O.E.M. applications; yet includes many enhanced operating features found in products costing much more:

Features

- Programmable hold and run currents
- Selectable "PLC" sourcing input mode
- All inputs withstand >28 volts
- Intelligent control up to 40,000 SPS (Steps Per Second)
- Single 24 to 48 volt power supply input
- Full, 1/2, 1/4, 1/8 micro-step resolution
- 2k bytes of non-volatile memory (NV Memory)
- Go, Stop, Limit, and Home input
- Step, Direction and Jog inputs
- RS232/RS485 Serial communication (1-32 axes)
- 4.0 amp/phase (Peak) chopper drive output
- Programmable acceleration and deceleration ramp
- Constant velocity commands
- Heat-sink mounted and mating connectors included

Required Hardware and Peripherals

The following table is a list of components needed to get your D-404 up and running. All components listed in this section are available through your motor sales representative.

Qty	Unit	Model #	Description
1	Axis	D-404	Driver-Controller Board
1	System	User defined	+24 to 48Vdc power supply
1	Axis	ASI-9	RS-232 serial adapter (single axis)
		Or	
1	System	ASI-10	Intelligent serial adapter (Windows 2000/NT)
1	Axis	BLC-38	7 pin home/limit mating connector (included)
1	Axis	BLC-44	6 pin mating motor connector (included)
1	Axis	BLC-50	12 pin mating I.O. connector (included)
1	Added axis	CAT503	Interconnect cable, Cat5 (3 ft.)

Hardware and Peripherals

Specifications

Electrical

Output Current (maximum)……. 4.0 Amps
Chopping Frequency.....……...20kHz
Supply Voltage.....……. +24 to 40 Vdc
Motor Step Resolution.....……. 1/8,1/4,1/2, Full, Wave

Non-Volatile Memory.....……...2k Bytes
Position Counter.....……. ±8,388,607
Baud Rate.....……... 9600, 470k
Serial Interface……... RS-422 4-Wire, Full Duplex

Signals	Min	Typ	Max	Units
RX, TX	-7	5	12	Vdc
High Input Voltage		2	28	Vdc
Line Input Current	-0.8		1	mA
Party Select	-.3	*2.5	36	Vdc
External Terminator		220		Ohms (Ω)

General Use Pins Electrical Characteristics

*Threshold

J1 Input Signals: Ports 1, 2, 3, P4_In, Jog 1, Jog 2, Jog Speed, Step and Direction
J4 Input Signals: Limits, Home, Go and Soft Stop

Signals	Min	Typ	Max	Units
Vlo Supply (J3-7)	4.6		28	Vdc
Threshold	*2.3	½ Vlo		Vdc
Input Voltage	-0.3		28	Vdc
Input Current	*0.5	Vlo/10	2.8	mA

J1 and J4 Electrical Characteristics

* Vlo = 4.6 volts using internal supply

J5 Output Signals: Ports 4_Out and 5 (open drain with 100k pull-up's to Vlo)
Clamp diodes are required for inductive loads.

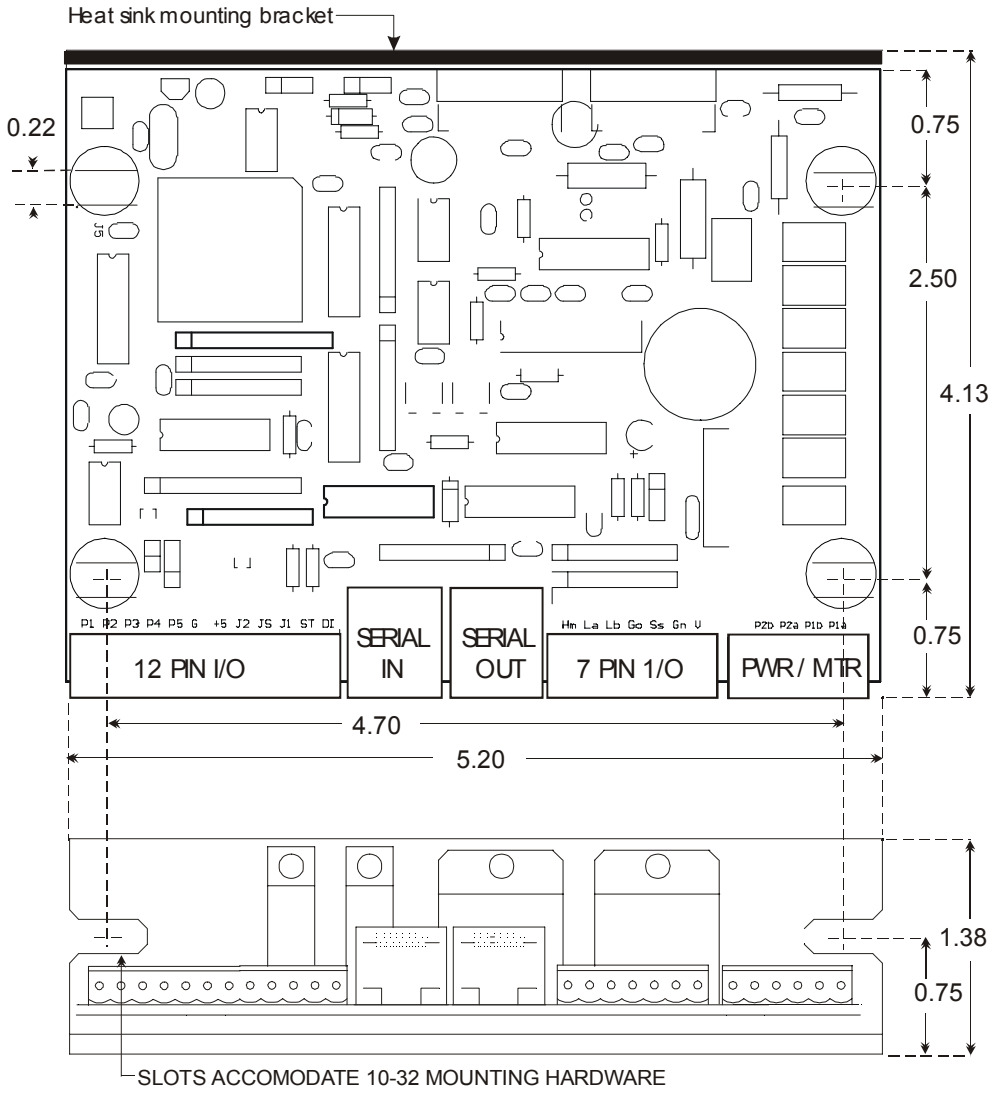
Output Signals	Min	Typ	Max	Units
Outputs Ports 4 and 5	0.7		28	Vdc
Output Current (sink)			2	Amp

J5 Electrical Characteristics

Environmental
Storage...……. -45 to 85 Degrees C
Operating.....…... 0 to 55 Degrees C
Humidity.....….. 0 to 95% (non-condensing)

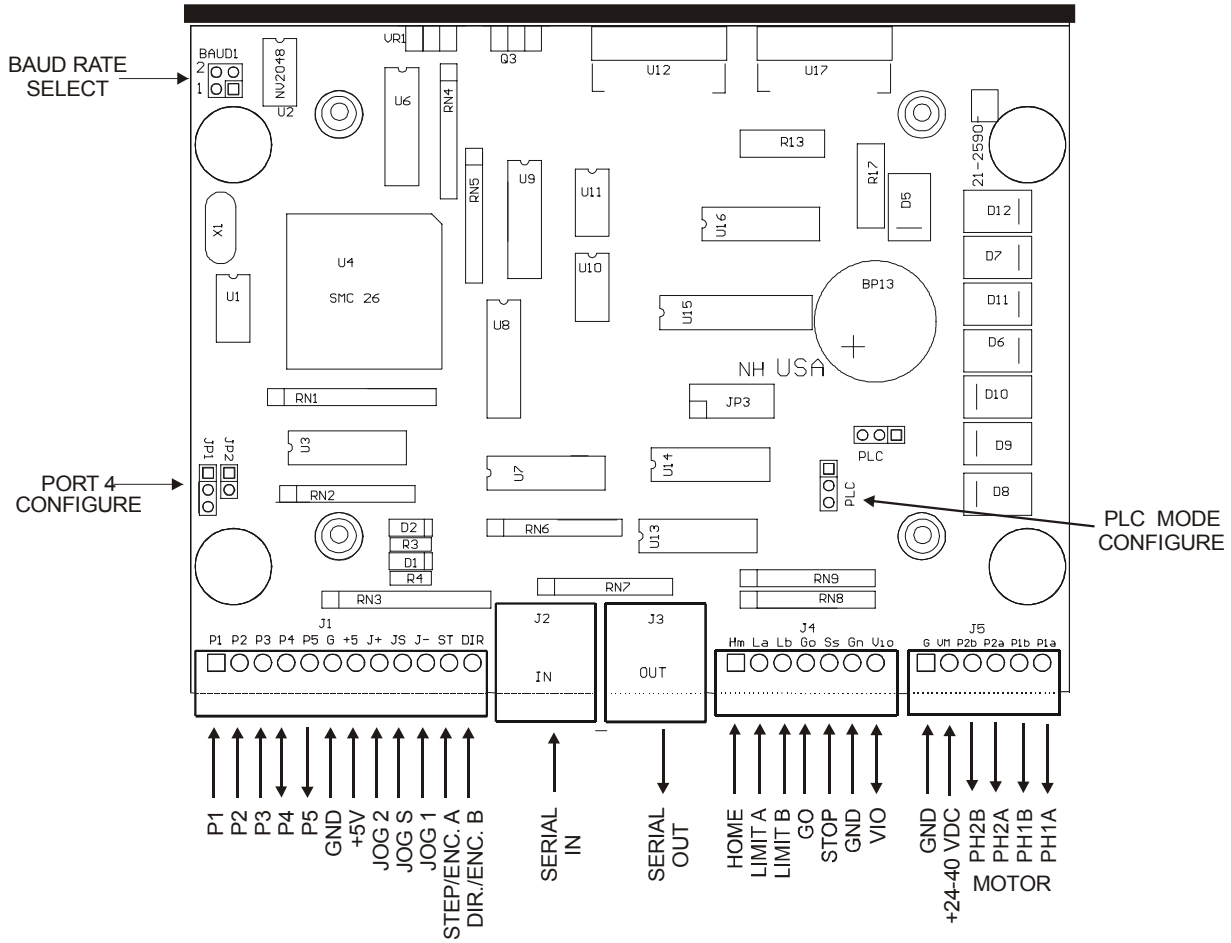
Physical
Size (in.)…….... 5.2 x 4.13 x 1.4
Weight.....…… 8 oz.

D-404 Board Dimensions (inches)



D-404 Circuit Board Dimensions

Assembly Drawing



Assembly Drawing

Unless otherwise specified clamp diodes D-1 and D-2 are not installed. Refer to "Output Ports" further on in this section for more information.

Out of the Box – Quick Start

Getting the D-404 up and running is a quick process involving a PC and hardware supplied with your D-404 purchase.

Equipment Check List

To begin development with the D-404 the following list of items are needed to properly configure and interface with the controller:

1. A computer with one available serial port.
2. Installed serial communications software such as Hyper term, EADDEMO, ProComm, etc.
3. D-404.
4. ASI-X serial adapter (ASI-10 recommended).

5. A +24 to 48Vdc power supply[1].
6. A compatible stepper motor.

Preliminary Settings

-ASI-9 (non configurable)
Or
-ASI-10 (all (8) switches in off position- towards DB-9 connector or away from LEDs).

-D-404: Remove all baud rate jumpers. This configures a 9600-baud default.

Basic Set-up

1. Connect the ASI- (n) adapter to serial input (J2) of D-404 with RJ-45 cable.
2. Connect the ASI- (n) DB-9 end of the serial adapter to the computer RS-232 serial port.

Make Sure Power Supply is OFF

3. Connect your power supply to the power connector (J5). J5-pin 1 is negative (GND). J5-pin 2 is positive (+) VMM. Improper power connection will destroy the electronics. GND is also connected to the base plate.

Motor Connection

1. Set the motor current off (especially if your motor is small and low current model) using the "Y" command.
    1. "Y 0"<CR> (windings off)[2]
    2. "S"<CR> (store settings in NV[3] memory)
    3. Connect stepper motor to drive according to table below[4].

Drive Pin	Motor Connection
P1A	A
P1B	A'
P2A	B
P2B	B'

Motor Connection Pin Assignments 1

Note: Before plugging into J5, insure low resistance between pins 5 and 6, and pins 3 and 4. There should not be low resistance between pins 4 and 5. An Ohmmeter can be used to determine this.

Configure Serial Communications

Full duplex, 9600 baud, no parity, 8 data bits, 1 stop bit.
Note: <CR> is the enter key.

Apply Power

4. Turn on power supply at this time.

[1] The D-404 will operate at 12Vdc minimum.
[2] See the usage of the Y command later in this manual. This command is used to adjust hold and run currents.
[3] Non-Volatile Memory
[4] See the EAD website for 6 or 8 wire motor configurations

Sign-on

5. Depress the SPACE BAR key.
    a. D-404 should sign-on. "13190 EAD-SMC27 V1.01" or similar.
6. Type X, then press Enter <CR>

The drive parameters are displayed. The last characters displayed will be "n= " followed by the axis "name" character, usually "A."

Setting/Resetting Axis Name

Setting the axis name is not important when operating in single axis mode. It is necessary when operating in a multiple axis system.

1. Turn off power supply.
2. Allow D-404 circuitry to discharge (10 seconds).
3. Turn on power.
4. Depress the desired "name" key, for instance "B."
5. Depress the SPACE BAR. The D-404 will sign-on.
6. Enter X<CR>. The new name "B" is displayed (n=B).
7. Issue the S<CR> (save command). The name is stored in memory.
8. Depress Ctrl-C key (soft reset) followed by SPACE BAR. Sign on occurs.
9. Enter X<CR> to check that the name was saved correctly.

Troubleshooting

1. No Response from terminal program:
    a. Check connections.
        a. Power Supply
        b. Adapter
        c. RJ45 Port
    b. Check for correct COM port.
    c. Check switches if using ASI-10.
        a. Party
        b. Others
        d. Inspect RJ-45 connectors for bent contacts.
2. Unrecognizable characters appear on the screen:
        a. Verify baud rate and serial parameters.
        b. Check the RJ-45 connectors/cable ground continuity.
3. "GO-SS conflict" messages appear on screen.
        a. Make sure the three-pin PLC jumper is in the standard mode (pins 1 and 2 installed).

Do's, Don'ts and Important Notes

- ***The physical direction of the motor with respect to the direction input will depend on the connection of the motor windings. To reverse the direction of the motor with respect to the direction input, switch the wires on phase 1 or phase 2 of the outputs.***
- ***NEVER connect or disconnect motor wires while power is supplied.***
- ***When using a 6 lead motor be sure to insulate/isolate unused wires.***

********** **WARNING** **********

CONNECTING OR DISCONNECTING MOTORS WHILE POWER IS SUPPLIED WILL CAUSE DAMAGE THAT IS NOT COVERED BY WARRANTY.

****

Serial Interface (J2, J3)

Two (RJ-45) connectors provide a loop-through connection, facilitating multiple axis systems. This "mini-drop" network allows for a single ASCII character "name" to be assigned and stored in the integral non-volatile memory of each axis during the setup procedure.

J2			J3		
Pin	Signal	Comment	Pin	Signal	Comment
1	J2-1	Not used	1	J3-1	Not used
2	GND	Power Gnd	2	GND	Power Gnd
3	RX-	+Data in	3	RX-	+Data in
4	TX-	+Data out	4	TX-	+Data out
5	TX+	-Data out	5	TX+	-Data out
6	RX+	-Data in	6	RX+	-Data in
7	5V	Power for serial adapter	7	N/C	Not used
8	Party	Enable party line or single	8	Party	Enable party line or single

Serial Interface Connections

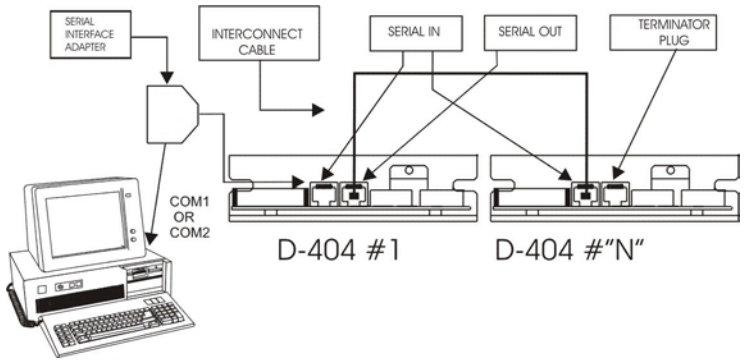
All signals on J2 and J3 are interconnected excluding Pin 7.

There are two types of serial adapters available.

1. ASI-9 (DB-9). This is a passive adapter that allows basic RS-232 connections, suitable for single axis usage. Party line is not possible with the ASI-9.
Note: Daisy chain protocol is possible using the ASI-9 but not recommended.

3. ASI-10 (DB-9) is a micro controller based adaptor that features a dual UART, RS-232 to RS-422 converter with selectable baud rates, and built in handshaking. Communication rates range from 9600 to 470K baud.

Note: If you have not chosen the ASI-10, an "echoed characters" handshake protocol must be implemented in the host computer to avoid loss of characters. If your operating system or application software is not capable of character-by-character transmission, an ASI-10 is probably necessary. A complete description of the serial interface specifications and operation is contained in Section 2, "Serial Interface."



Multi-axis Serial Interface Connection

I/O Connections (J1, J4)

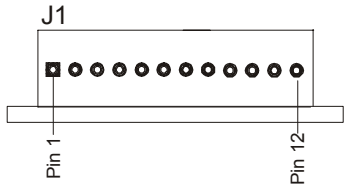
Two connectors (J1 and J4) provide dedicated inputs and outputs. Five user-defined signals are also provided. Two high power outputs are available to drive solenoids, relays, dc motors, etc.

All inputs can withstand +28 volts; care should be taken when operating at or above 28V. Unexpected results or damage can occur when this input level is exceeded.

Fourteen input signals can be configured as either sinking or sourcing "PLC" inputs.

(J1)

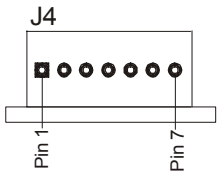
Pin	Signal	Description
1	Port 1	Input
2	Port 2	Input
3	Port 3	Input
4	Port 4	Configurable output or input
5	Port 5	Output
6	Gnd	Power common
7	+5v	Vcc – logic power
8	Jog-1	Input
9	Jog-Speed	Input
10	Jog-2	Input
11	Step	Input
12	Direction	Input



J1 Pin Assignments

(J4)

J4 Pin	Signal	Description
1	Home input	Used with the F[ind] home command
2	Limit A input	Inhibits motion in + direction only
3	Limit B input	Inhibits motion in - direction only
4	Go input	Start stored program sequence at location 0
5	Soft Stop input	Stop stored program sequence
6	Gnd	System power common
7	Vlo	+4.6 volts out or Vlo input



J4 Pin Assignments

Pin 7 - Vlo Reference Input

This signal defines the input and output voltage range. The default voltage is 4.6 volts derived from the internal 5-volt supply, minus one diode drop.

A higher, external "Vlo" voltage can be supplied to the Vlo pin to define a new reference voltage. The input signal thresholds will be approximately ½ of Vlo.

External loads such as sinking resistors or LEDs must be limited to 10 milliamps.

Input Ports

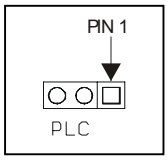
All inputs incorporate 10k resistors that may be set as pull-up to the Vlo voltage (sinking) or pull-down to ground (sourcing) as defined by the three-pin "PLC" jumper.

When the jumper is in the "PLC" position, (pins 2 and 3 installed) the resistors pull-down and maintain a zero volt level on the inputs. All inputs will withstand 28 volts DC (even if a lower Vlo voltage is used). The possible inputs include; (J1)- ports P1, P2, P3 and P4, Jogs (3), Step and Direction, Home, Limits, Go, Soft Stop, Vlo.

Sinking and Sourcing Configuration

Port 4 defined as:	JP1	JP2
Input	1-2	None

Port 4 Jumper Settings Input 1



Standard Mode- Sinking Inputs

When the PLC jumper is in the standard mode position (pins 1and 2 installed) the resistors act as pull-up to the Vlo supply. This pull-up effect reverses the logic levels from low to high. You can use commands to invert logic levels relative to the processor. Using the ”l” (lower case L) command will cause the processor invert the levels.

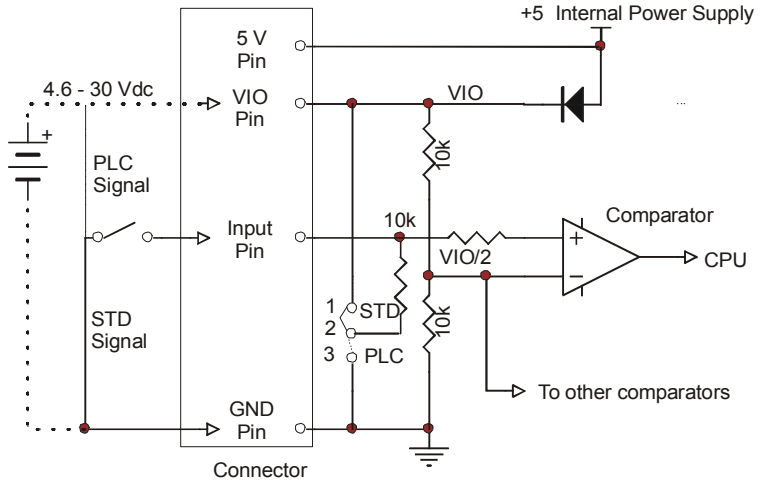
In this scenario 0 volts or Grounding the input is on and 4.6 (5) volts is Off.

PLC Mode- Sourcing Inputs

When PLC jumpers (pins 2 and 3) are installed, the pull-up resistors are wired as pull-down resistors. Activating an input is accomplished by asserting a voltage exceeding ½ Vlo on the given input. In this scenario 0 volts is off and 4.6 (5) Volts is on.

For proper operation a mode command must be entered and stored in NV memory. Use the “l” (lower case L) command to configure the inputs. The “l 9” command will invert all of the inputs. The “l 8” command will invert all but the limit switch inputs.

The following diagram describes the typical interface option for sinking and sourcing modes.



Typical User Input Circuit

(Step, Direction, Ports 1, 2, 3, 4 (used as an input), Jog (3), Limits, Home, Go and Soft Stop)

Note: A higher Vlo (i.e., 24 volts from a PLC with 24 volt drivers) increases the logic threshold to 12 volts, providing better noise immunity.

Output Ports (P4, P5)

Two user output ports are provided on the D-404 with a power FET output circuit capable of sinking up to 2 amps DC. They cannot be converted to a sourcing mode. If a source signal is necessary, a relay should be installed.

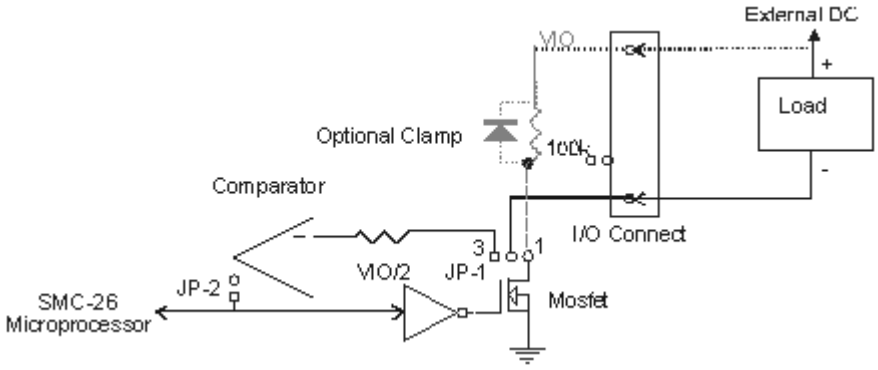
The outputs do not include clamping diodes attached to Vlo. There is provision for diodes (D1, D2) on the board .If these diodes are installed; **Vlo MUST BE EQUAL TO OR HIGHER THAN THE DEVICE POWER SOURCE VOLTAGE.** If a 24-volt solenoid valve is to be driven, the Vlo input must be at 24 volts. Because Vlo also defines the logic input levels, this can be a problem. If your design is to drive inductive loads such as a relay or motor coil, you must implement external clamp diodes as required. R3 and R4 should also be removed. Call EADmotors for further description.

Modifications to the board should be performed by EADmotors at time of production.

Port 4 (Configurable user port with JP1 and JP2)
This port can be configured as either an input port (as described above) or output port.

Port 4 defined as:	JP1	JP2
Output	1-2	None

Jumper Setting for Port 4 (Output)



Port 4 Input / Output Circuit

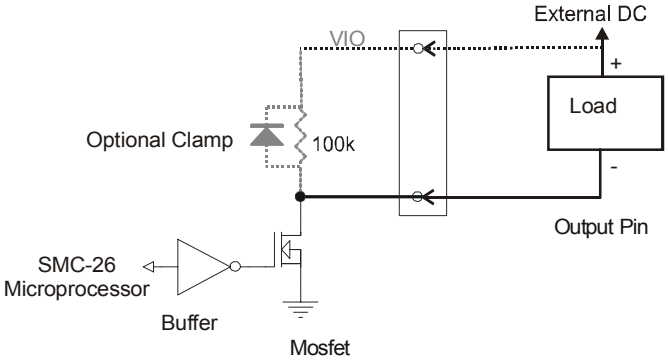
As an output (as shipped) the signal can be redefined as a stepping pulse output with the "l 2" (lower case L) command. This also defines P5 as a direction output signal

Port 5
Port 5 is an output port/driver, rated at 2-amps. The output signal can be redefined as a direction output with the "l 2" (lower case L) command. Port 4 becomes a step pulse output.

This signal can be redefined as a moving output with the "l 4" (lower case L) command. The moving output is useful in designs where a PLC or computer cannot interrogate motion status using serial communications (the preferred method).

The FET transistor ratings are RDSon=. 065 ohm, VDS = 55-volts and ID=17 amps (because the FET transistors are not heat-sunk, never draw more than 3 amps) Damage caused from misapplication of these devices is not covered by warranty.

The default "off" condition for port 5 is non-conducting (5-volts) when a port is turned on (such as using the "A 16" command). The output will conduct up to the rated current of 16 mA at a zero volt level (Ground), DC.



Port 5 Output Driver

Power Supply and Motor Connection (J5)

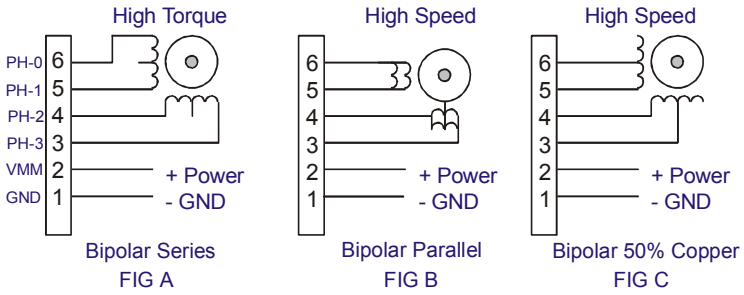
Connector J5 provides the power supply input and motor phase drive outputs. The recommended power supply is an unregulated DC design with the following voltage and current ratings.

Parameter	Min	Max	Units
Output Voltage	24	48	Volts (V) DC
Output Current	1	4	Amps (A)
Type	Regulated or Unregulated		Load configuration

Power Supply Characteristics

Pin	Signal	Type
1	Gnd	Ground
2	VMM	+24 to 40Vdc
3	P2B	Motor B'
4	P2A	Motor B
5	P1B	Motor A'
6	P1A	Motor A

Motor Power Supply Connector



Typical Wiring Diagrams for Step Motors

Fig. A: Series winding for higher torque and lower current. The inductance is 4 times that of the parallel mode, reducing the maximum obtainable speed.

Fig. B: Parallel winding for better high-speed performance but requires higher drive current. A 4-wire motor is the same as an 8-wire motor, but it is connected (in either parallel or series) internally. Some motors can be rewired at the factory.

Fig. C: A 6 wire motor is a variation of the 8 wire series configuration, where the "center taps" are available. The 6-wire motor can be used in series mode but cannot be connected in parallel. A compromised 50% copper connection can be used, producing higher speed with reduction of torque.

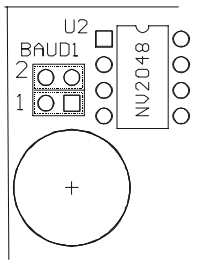
Note: NEVER connect or disconnect the motor when the power is "ON". Wait at least twenty seconds after turning off power, before connecting or disconnecting the motor. This will allow proper dissipation of voltage from the unit. Failure to do so may cause damage and void the warranty.

Baud Rate Jumpers (B1, B2)

Jumpers JP1and JP2 are used to configure baud rate on the D-404. In addition to baud rate the UART properties are no parity, 8 data bits, 1 stop bit.

JP1	JP2	Baud Rate	Comment
N	N	9600	Factory setting N, 8, 1
Y	N	460k	Requires ASI-10 serial adapter
N	Y	38.6k	""
Y	Y	19.2k	""

Baud Rate Settings



Baud1 Illustration (JP1, JP2)

Interface Overview

This Application covers the serial communication design for the D-404. The basic design is based on two-way serial communication using a standard ASCII character set used in virtually all computers.

The hardware interface for the can be RS-232 or RS-422. In either protocol two signals are used, TXD (transmitted data) and RXD (received data). These standards describe the physical wire layout and signal voltage swings.

RS-232
This single ended transmission is useful only for short distances[5]. The connector (ASI-9) is a 9 pin "D" connector. RS-232 includes a 3-wire interface using RXD, TXD and ground.

Typical voltage swings are –12 to +12 volts. The relatively high impedance and low speed limit both baud rates and wire length. In non-hostile environments and limited distance communications it will operate with 0 to +5 volt swing.

RS-232 can operate with one serial driver and one serial receiver at a maximum of 20k baud.

RS-422
RS-422 incorporates a 2 wire per signal differential standard that is faster and useful at up to 400 feet at 1M baud or 4,000 feet at 100K baud. This 5-wire signal, full duplex interface includes RX+, RX-, TX+, TX- and Ground. This system is capable of driving up to 32 receivers (motor control axis) on a single "drop" bus. Voltage swings are 0 to +5 volts.

While not recommended, the RS-422 line receivers will withstand RS-232 levels. RS-232 receivers work with the lower 5-volt swing, permitting both single ended and differential operation.

Communication Hardware

ASI-9 RS-232 Serial Adapter
The ASI-9 is a DB-9 to RJ-45 adapter for single axis, RS-232 communication applications. This converter facilitates connection between a standard computer "D" connector output and the D-404 RJ-45 connector inputs.

Features include:

- Connects computer or terminal to axis
- Standard 9 pin connector for COM port
- Passive - no power required

The other method, called "daisy chain," is for multi-axis applications and is not recommended while using the ASI-9 (see section ASI-10 for daisy chain). Daisy chain is accomplished by interconnecting one or more axis in "series." Here, the TXD output of the first axis is connected to the RXD input of the next axis, etc., with the last axis TXD connected to the RXD of the computer.

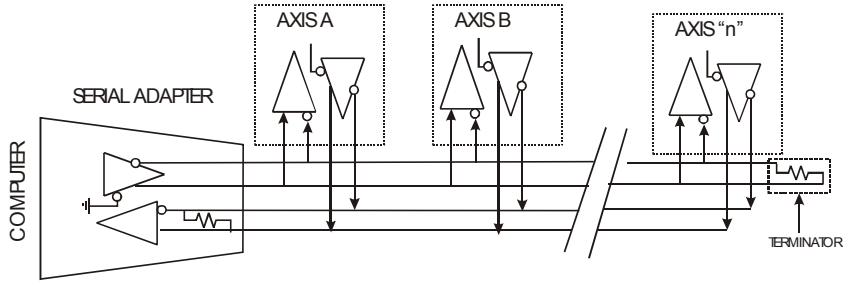
Multiple axis daisy chained systems are not recommended using the ASI-9 due to electrical and software issues.

[5] Information can be found about cabling lengths at Texas Instruments. Maximum cable length at 9600 baud is 200 feet; at 19200 baud maximum cable length is 50 feet.

ASI-10 RS-422 (Party Line) Hardware
EAD "Party Line" communication is an RS-422 design that uses RS-485 rated circuits. This interconnect is comparable to a LAN configuration. The hybrid design merges the best of both EIA specifications and maintains compatibility with EIA RS-422 and features:

- Multi drop serial bus
- Full duplex connection.
- Zero to five volt differential
- Data speeds from 100K to10M Baud
- Up to 32 controllers from one COM port
- Cable network length to 1200 Meters (4000 ft)

Party Line Connect



RS-422 "Party Line"

Party line hardware requires three components:

1. A serial adapter (RS-232 to RS-422),
2. A cable(s) and
3. A terminator (supplied with adapter).

Cable Specifications
The recommended interconnect cable for party line use is UTP 24 AWG Twisted-Pair (category 5) network cables. The older style telephone cable is suitable for RS-232 or lab work. These cables can be purchased optionally with the D-404 (P/Ns C-CAT503, C-CAT506).

Device (Axis) Connection
Each motor controller has two RJ-45 connectors for party line. The +5V line is isolated to prevent "back" feeding power supply voltage to any individual axis with the wall power to the power supply switched off or power removed. Thus, it is important that interconnecting cables travel from the output connector (on right) of one axis to the input connector of the next axis (on left)[6].

Two terminating resistors are used on the last differential line receiver. The serial adapter contains one resistor and a terminator plug is to be used on the output connector on the last axis in the Party Line bus.

ASI-10 Intelligent Serial Adapter
The ASI-10 is an intelligent serial line converter that simplifies application software development and improves overall performance. Dual, independent UARTS permit spooling commands at rates between 1200 and 115k baud. ***Specific operating instructions are contained in the ASI-10 Users Guide.***

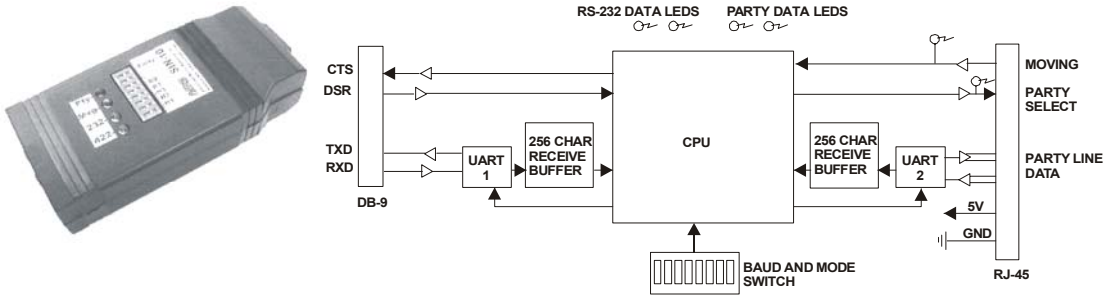
The ASI-10 has a built-in microcontroller that offers a number of features:

- Diagnostic LED's
- Numerous Input baud rates

[6] See "Multi-axis Serial Interface Connection" diagram shown previous**.**

- Output (party line) baud rates – including the special 470k baud rate[7]
- DB-9 serial input connector
- RJ-45 party line connector
- 5 volt powered from controller
- 250 character buffers for multiple commands per line

Because the ASI-10 eliminates the need for echoed character software, it can be used in Windows applications where either the machine or software is slow and/or the operating system prevents direct programming of input or output instructions.



ASI-10, intelligent serial line converter

Other Party Line Signals
In addition to the serial data bus wires, several other signals exist on the D-404 party line.

1. **GND** (pin 2) Ground is common for all devices (axes). All power supply commons are connected to prevent high common mode voltages. Please note that the power common is generally connected to the case return.

2. **+5 Volts** (pin 7) is available to power the serial adapter (ASI-10) from the first axis.

Note, The ASI-10 determines motion via special ASI-10 instructions.

4. **Party Select** (pin 8). Each axis tests this input after reset (power up or in response to a ^C command) to determine either single or party line mode. Single mode (only one axis connected) is used for debug, programming for stand-alone use or assignment of party line "name". The ASI-10 is selectable via host commands or a built in mode switch.

Serial Communications Overview

There are three methods (protocols) used to send and receive command and data from an EAD controller (axis):

1. Simple "Dumb" Communications
This is accomplished by connecting one single axis to the computer. Commands can be typed in and the controller will execute them. The designer can also enter program sequences into the NV memory and execute them. Virtually every capability can be explored. It is a "human friendly" interface and **NEVER** a computer controlled operation.

Serial adapters used: ASI-9 or ASI-10.
At start-up:

[7] As referenced previously, the 470K-baud transfer rate takes place between the ASI-10 and the D-404 controller. Baud rate from the pc to the ASI-10 is not affected by this setting.

1. If using an ASI-10 serial adapter, ensure the adapter is in "single-line" mode.
2. Hit the SPACE BAR key to sign on after reset.

2. Party Line Mode
Party line mode is intended for computer-controlled designs. A computer (usually a PC) can address one or more axes using a "mini drop" network implemented with CAT-5 network cable with RS-422/485.

Between 1 and 32 axis are configured as "slaves." Unlike the "Dumb" mode, a proper character by character echoed protocol is necessary for proper operation.

Serial adapter used: ASI-10.
At start-up:

1. Make sure the serial adapter is in "party line" mode.
2. The host computer interrogates and records axis name(s).

3. Daisy Chain Mode (not recommended for more than 1 axis)
This older protocol is similar to the party line mode but RS-232 protocol is used. Because it involves special wiring of RXD to TXD signals, it should only be used with a single axis design. When multiple axes are implemented they are less reliable, communication speeds are slower and troubleshooting is difficult.

The advantage is the host computer on power up sequence can dynamically assign the name and the computer protocol can be implemented with the lowest cost RS-232 adapters.

Serial adapters used: ASI-9.
At start-up:

1. The host computer emits axis #1 name, receives ending axis name +1.

Serial Communications Software

The communication protocols described herein make use of COM ports available on most PCs. EAD has chosen to support PC interfaces with the D-404.

The provided interface software (EADDEMO) is a DOS based application. This can be downloaded from the EAD website.

Baud Rate
The standard default baud rate for the D-404 is factory set to 9600 baud. . Serial data format is 8 bits; 1 start bit, 1 stop bit, and no parity.

EADDEMO Software

To assist with the implementation of your project, EAD offers a software program called EAD-DEMO. This software operates under DOS, Windows-95, Windows-98, and Windows NT (as a DOS application).

In addition to EADDEMO, almost any communication program, such as ProComm, or Hyper Terminal can perform dumb terminal operations, provided the necessary protocol is implemented (Echoed Character).

Serial Interface Using EADDEMO

1. Attach the DB9 connector end of the serial adapter to either COM1 or COM2 of your PC/AT compatible computer.
2. Connect the other end of the cable assembly, to the mating connector 'J2'.
3. Install a terminator plug (TERM-1) into J3 "Party Line Serial Output" of the last axis. The controller will work without a terminator plug, however reliability may be compromised, based on the environment (noise) and length of the interconnect cables.

4. Apply power to the device. (Reference Hardware Section).
5. Install the diskette and type: EADDEMO<CR> at the prompt. At the opening screen enter "n" if you are operating from a monochrome terminal or hit the ENTER key for color. Use the arrow keys to select COM1 or COM2. COM1 is the default setting. Follow the same procedure to select the correct baud rate (9600 BPS is the default setting).

Sign-On

Before starting the sign on procedure verify the following settings:

- Using an ASI-10, make sure the party line switch (S8) is in the off position, Switched away from the led displays.

Single axis mode must be used for name assignment. Any terminal software, including EADDEMO "dumb terminal" can be used.

1. Start EADDemo.exe.
2. Select the "DUMB TERMINAL" mode. A blank window will appear.
3. Strike the SPACE BAR key. The controller should sign on with the software version number Vx.xx. If not, enter a (^C) (Reset) and press the SPACE BAR key again. The "reset" message is generated by EADDemo.exe, not the axis output.
   a. If sign-on does not occur:
      a. Verify all connections.
      b. Insure that the ASI-10 is in the "single" mode.
      c. Check your COM port set-up.
4. Pressing the ENTER <CR> key should result in an echo of the "#" character, indicating communication is established.

Examine Command

The Examine command (X)<CR> will display a set of parameter values that were last stored into non-volatile memory. These parameters may be modified using the appropriate commands, then stored in non-volatile memory as the new "defaults."

X Y 10 / 75 K 5 / 5 I 802 V 10370 D 1 b 30 s ¼ n=A

Where:	Y= Hold and Run Current	K= Ramp up/ramp down
	I= Initial velocity	V= Slew velocity
	D= Divide factor	b = Decay Threshold
	s = Step Resolution	n= Axis name

Some Rules

1. The command line may be edited using backspace as characters are typed.
2. The line may be canceled using <ESC>.
3. The command line is limited to 15 characters.
4. Only one command may be entered per line.
5. A space is optional between the command and first number.
6. A space or comma must be used to separate two parameter commands.

Axis Name Assignment

Whenever the application is controlled via a "host" computer the proper protocol (handshake) must be used. Either the programmer must write the necessary serial, echoed character-by-character software (driver), or implement an ASI-10 for the proper handshake.

EAD-Demo software searches for axis names starting at the beginning of the alphabet. While X, Y, or Z might be more descriptive, it can take a longer time to scan than an axis named with a letter earlier in the alphabet. Recommended names are as follows:

Valid Names: - (Upper case A through Z) - (Lower case a through z)		**Invalid Names:**	
ASCII	HEX	ASCII	HEX
[	5B	^C	03
\	5C	CR	0D
]	5D	LF	0A
^	5E	@	40
-	5F		
‘	60		

Axis Naming Convention Table

To assign an axis name:

1. Reset the controller to cycle power (5 volts) or enter CTRL+C (^C).
2. Type the single name character[8] (once only).
3. Follow the name with a SPACE by pressing the space bar. The sign-on message will appear.
4. Enter the X<CR> command. The name will show at the end of the echoed line.
5. Issue the S<CR> store command (saves name in NV memory).
6. Reset the controller by pressing CTRL+C.
7. Sign on with the space character.
8. Use the X<CR> command to verify the proper name.

The name is now stored in the memory on board the D-404, and can be used to reference the controller.

Note: The controller will accept any character as a name, including control characters. Two common error are space ‘ ‘ or ^C. In either case, reset the controller and start the axis naming procedure again.

Daisy Chain Start-Up

Multiple axes may be wired (daisy chained) using the controller logic levels on the RJ45 connections or buffered with line driver-receivers. Characters are received by the first controller input (RXD), and then echoed to the next controller (RXD->TXD) in the serial link. The host terminal/computer receives characters from the last axis (TXD) in the link. The closed loop communications assures the integrity of data.

Baud rate settings must be the same for all axes.

The initial input sequence must be a Line Feed (LF) character followed by a valid “name” character in the standard ASCII collation sequence. Once the sequence is received, the controller assumes the name is unchanged until a hardware reset is made. The first axis in the chain will transmit the Line Feed character followed by the next higher ASCII character in the character set. This sequence continues until all axes have assigned names (e.g. If the first valid usable “name” character is an upper case “A” the controller will be assigned the prefix name “A” then output a “B” to the next axis controller in line. Four Daisy-Chained axes would then assume the names of “A”, “B”, “C” and “D.”). The last controller will respond with a Line Feed and character representing the last name plus one (“E” in this example).

NV Memory Programming

Entering Instructions

The following is a sample sequence used to store a sequence in the non-volatile memory. Note that when programming, the sequence is immediately written to non-volatile memory and saved. This example starts at location “0” of NV memory.

[8] Names are case sensitive an axis can be named with lowercase ‘a’ and the next axis is uppercase ‘A’ and both axes will be unique.

	Enter	Remark
	P0<CR>	Place in Program mode. Insert instructions at location 00.
Address		
0	O0<CR>	Set Origin to zero.
1	R10000<CR>	Move 10,000 steps in the "+" direction, relative to Origin.
6	W 0<CR>	Wait until complete.
9	P0<CR>	End Program.

Now list the stored program

	Q<CR>	Query command.

Verify the Program

The controller will respond with:

0	O	0
1	R	10000
6	W	0

Execute the Program

Enter	Remark
G0	Programs start executing at location zero.

Note: interfacing a switch on the "Go" input can trigger this program.

Note: Caution should be exercised when making Program Edits in dumb terminal mode due to variations in command byte length that may effect subsequent command address locations and possibly corrupt stored programs. It is recommended that application programs be developed using the menu driven program (Party Line selection) in EADDEMO, which includes an editor and interpreter.

Editing features are not supported in daisy chain.

Party Line and Daisy Chain Line Commands

Note: The ASI-10 will perform the necessary handshaking automatically.

Some Rules

1. The first character of a command must be the "name" character assigned to the axis.
2. The command line terminator must be a Line Feed character.
3. The axis name must be preceded by an LF (generally the terminator from the previous command), i.e., <LF>"n" xxxxxxx <LF>.

Note: An LF can be generated using Ctrl+Enter key combination on a PC.

The first Line Feed "resets" the command buffer for all axes. The controller then tests the character immediately following a Line Feed. If this character matches the assigned "name," the axis will interpret the following characters (up to 12) as an input command. The designated controller then re-issues the Line Feed after processing the command. If the axis does not detect a proper name and command, the data is echoed back to the terminal.

If the command results in data output (such as "Z"), the data (result) will be inserted before the Line Feed. The Line Feed does not indicate that a move or other time consuming command is finished but only initiated.

Note: the commands "CTRL+ C" and "ESC" are global to all axes. All devices will respond.

The party line sequence can be sent using the dumb terminal. Caution must be used because any typing errors cannot be corrected with a backspace. You must cancel with the ESC command and start over. *See note above.*

Party Line Startup (performed automatically by the ASI-10)
The programmer can verify the presence of the axis on power up by:

1. Sending a linefeed<lf> character.
2. Sending a good "name" character.
3. Waiting for echo of same name.
4. Sending a <lf>.
5. Repeating 2 thru 5 for each axis in system.

Command Example
The following example assumes two controllers are connected with name assignments of "X" and "Y." The characters are echoed back to the host as a handshake function. The host awaits each individual character. Timeout routines should be used to prevent processor hang-up.

Index 1000 steps for axis X
Output from Host: `X + 1 0 0 0 (LF)`
Response from named controller: `X + 1 0 0 0 (LF)`

Index 500 steps for axis Y
Output from Host: `Y - 5 0 0 (LF)`
Response from named controller: `Y - 5 0 0 (LF)`

Read Motion Status
The returned decimal value (xx, yy) represents the motion status. When both least significant bits are zero ("and" with 3), the motion is stopped.

Output from Host: `X ^ (LF)`
Response from named controller: `X ^ xx (LF)`
Output from Host: `Y ^ (LF)`
Response from named controller: `Y ^ yy (LF)`

Read Position
Input from Host: `X Z (LF)`
Response from named controller: `X Z 1000 (LF)`

Note: Response is the position data requested from axis X. The handshake must be character-by-character confirmation.

Example: the +1000 command
Host sends "X", host waits for "X" echo.
Host sends "+", host waits for "+" echo.
Host sends "1", host waits for "1" echo.
Host sends "0", host waits for "0" echo.
Host sends "0", host waits for "0" echo.
Host sends "0", host waits for "0" echo.
Host sends "LF"; host waits for "L" echo.

Example: Read Position
Host sends "X", host waits for "X" echo.
Host sends "Z"; host waits for "Z" echo.
Host sends "LF"; host waits for "LF" echo.
While waiting for the LF the host receives the "1000" data and stores it into the position value.

Anatomy of Instruction Execution
This information is intended to familiarize the programmer with the internal operations involved in executing a command.

For each motion command (M, R, etc…) there are four cycles; Entry, Execution, Result, and Completion. Other commands have three cycles; Entry, Execution and Result. In the idle state the controller continually tests for jog, go, or command input. The following describes each operation that takes place on receipt of a command.

Cycle 1: Entry
A. Serial command and data information is placed in a command line buffer as received. Editing is permitted in SINGLE axis mode. Pressing Escape aborts the operation and returns to idle state. A carriage return (Line Feed for Daisy Chain) terminates the entry cycle and initiates execution.

Cycle 2: Execution
The command is processed. In the case of two consecutive action commands, execution will be delayed until any previous completion cycle has been completed.

Cycle 3: Result
The result cycle returns any numerical result required by the command, i.e., the position. The result type is signed numerical data, preceded by space padding and followed by a Carriage Return and Line Feed. If the result does not produce numeric data then the Carriage Return, Line Feed output indicates execution is complete.

Cycle 4. Completion
The completion phase is required for any Action command cycle.

The following are Action commands:

Action Command	Completion Cycle
GO	Until last instruction is complete
Step Resolution	Until previous action complete
Constant Speed	Until previous ramp is complete
Find Home	Until home is found
Relative Move	Until full index is complete
+Step Index	Until full index is complete
- Step Index	Until full index is complete

During the completion cycle (except for “GO”), any non-action command such as “Read Position” may be executed.

The controller has the capability to “queue up” another action command during the completion cycle resulting from a preceding action command. The execution and result cycle of this “Pending” command is delayed until the completion phase is complete. This interval is called the “Pending Period”. During this Pending Period, the only input accepted is the one character interrupt (abort) command, limit switches, soft stop, and hard stop (ESC).

Interrupt Commands
Interrupt commands are single character commands that will interrupt the operation in process as follows:

Abort
Any action command may be terminated using Escape.

Process	Resulting Action
Command line input	Clear input buffer.
Program mode	Exit without inserting “END”.
Action command	Terminate all motion (Hard Stop).
Program execution	Terminate execution, Hard Stop.

Abort is Global – execution on all axes will be halted.

Soft Stop “@”
The Soft Stop “@” can be either a command (Immediate mode), or a single character interrupt (Program mode). The Soft Stop operates only when motion resulting from action commands or instructions is taking place.

Soft Stop Interrupt
After velocity deceleration, the process is terminated.

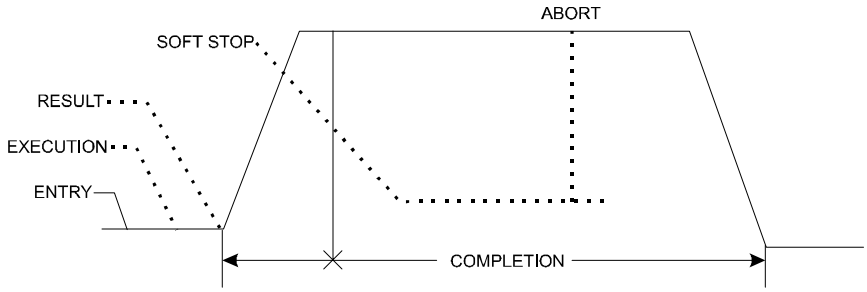
Process	Resulting Action
Pending period	Decelerate and cancel pending instruction.
Program execute	Decelerate then terminate execution.

During pending periods that are a result of multiple Constant Velocity commands (inter-speed ramping), deceleration will be delayed until the previous ramp-to-speed has been completed.

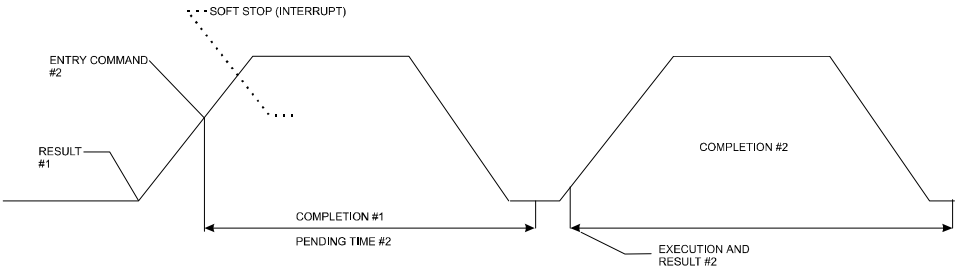
Jog Speeds, Homing
Jog input and home speed is a special case of the constant velocity command. Inter-speed ramping is used if the programmed jog speeds are above the initial velocity. Homing does not employ a deceleration ramp on reaching the home sensor.

Note: In all modes, jogging and command- reception are mutually exclusive. That is, a command cannot be loaded while jogging and jogging cannot be performed until the last command is complete. A command starts with the reception of the first command character.

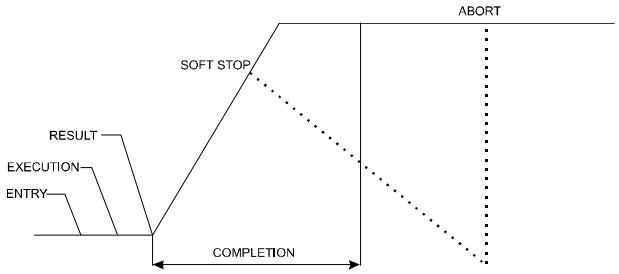
Command Cycle Examples
Index Cycle Resulting From +, -, R Commands



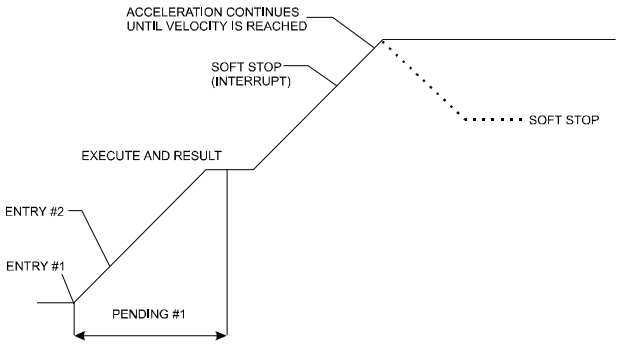
Command Instruction Timing



Queued Index Cycle Resulting From +, -, R Commands



Constant Velocity Cycle Resulting From M Command



Constant Velocity Cycle From 2nd M Command

Execution Times

The time for a complete cycle between command entry and result is variable, depending on number of data bytes, command type, and motion in process. On receipt of a line feed, most commands execute in less than one millisecond. The exceptions are:

Instruction	Execute Time
I, V (SPS)	3-4 ms
C0 (Reset defaults)	60 ms
C (Clear memory block)	1500 ms
S (Store)	60 ms
/,] (Read, Write)	1.1 ms
Index +, and R	5-10 ms

Execution Times

Times will vary depending on product. The EADDEMO program has a benchmark feature that can be used to determine times.

Non-Volatile Memory Details

The SMC-27X2 uses the X24C16, a 2048 byte EEPROM. A worst case of 4 bytes per instruction yields a capacity of 500 commands. These devices are rated to retain data for 100 years. As with all EEPROMS, the number of times it may be re-programmed is limited. At this time a life in excess of 1 million cycles is available.

To extend the life of the EEPROM in your device it is necessary to be aware of which commands of the SMC-27X2 perform writes to the EEPROM, and eliminate those which are not needed. For example, the RESTORE command ("C 0") will retrieve the parameters from the EEPROM without doing a write. If the INITIALIZE command ("C 1") was chosen, the first 256 BYTES of EEPROM are written. If you require a sequence of motions to be done without host attention, break-up the motions into sub-groups rather than repeatedly programming the EEPROM. Then use the GO from address command to execute the sub-groups in the required sequence.

Note: Use the SAVE command sparingly.

Do not change parameters by writing directly to EEPROM. The SMC-27X2 won't recognize that it was changed and may over-write them. Use the commands available to set parameters.

Trying to read and write Initial and Slew velocities from the EEPROM may be confusing as they are stored as timer reload values. Use the EXAMINE command ("X") in SERIAL mode to get initial values from the D-404 controller.

Memory Map

The following locations are accessible through the NV memory read/write commands:

Decimal	Description
0-127	User program or data storage
128-191	Shadow program area
160-*	Trip routine
192-226	Unused
227	Configuration byte
228	Options Byte
229	Divide factor (D)
230-1	Initial velocity low and high bytes (I)
232-3	Pointer value (I)
234-5	Slew speed (V) low and high bytes
236-7	Pointer value (V)
238	Low speed jog value (B)
239	High-speed jog value (B)
240	Acceleration ramp factor (K)
241	Deceleration ramp factor (K)
242-244	Trip Point low, mid and high bytes
245	Port value for trip (“k” data)
246	Resolution
247	Name
248	
256-2047	User program or data storage
256-511*	Branch area power up commands

NV Memory Map

Locations 247 thru 255 are protected from the “Clear” command. Most of the data contained in these locations is in binary format and should be left unaltered.

*Committed only when specific command is being used, otherwise used as general-purpose storage.

Default Table

The following default values are written to NV memory after the ‘Clear’ command (C1):

Parameter	Value
Initial Velocity (I)	800 SPS
Slew Velocity (V)	10,000 SPS
Divide Factor (D)	1
Ramp Slope (K)	5
Decay threshold (b)	30
Jog Speeds (B)	90/600
Trip Point (T)	Off
Mode (H)	1/4 micro step
PLC(l)	Off (invert all inputs)
Limit Polarity (l)	Low assert
Auto Position Readout (Z)	Off
Name (after reset)	Unchanged
User Programs (0-191)	Cleared

Default NV Values using Clear Command

Turbo Ram

The SMC-27X2 has a small, dedicated memory area called Turbo Ram. There are 64 bytes, which reside between address location 128 and 192. Instructions written here during program mode use "real" internal RAM rather than EEPROM in order to achieve these advantages:

1. Very fast execution. EEPROM access time is 1 microsecond or more per byte.
2. No wear and tear on the EEPROM.
3. The trip service routine executes at address 160.

Macros may be downloaded directly into this area and executed as frequently as desired. Programs in this area are stored in corresponding NV memory and "down-loaded" at power up, making an effective shadow RAM.

Command behavior between address locations128 and 192

Q: List from RAM
P: Program to RAM
S: Copy to EEPROM
]: Write to EEPROM
\: Read from EEPROM
C1: Clear EEPROM, reload register

Commands

Command Format Description

The following section describes the various commands available for the D-404 programming interface.

Each section will contain a table explaining the various parameters of the command, and a description of the action each command performs when used. The various elements in each command description table are described in the list of definitions below.

Command:	Keystroke
Function:	Functional description of command
Type:	Immediate = Direct execution Program = Executable in stored program Global = All axis present Default = Initial parameter setting Hardware = Auxiliary I/O
NV Bytes:	Storage requirements in program
Mnemonic:	Single character prefix used in multi-axis protocol; (Prefixed by axis "name" assignment in party line mode)
Data 1:	Affected parameters
(Range):	Valid numerical range of parameter(s)
Data 2:	Same as Data 1 (as required)
Result:	Information returned as a result of command execution or examination

Command	Function	Type		NV Bytes
ESC	Terminate Operation	Immediate		N/A
	Mnemonic	Data 1	Data 2	Result
	(Name) Esc Char	None	None	Echo #

ESC (Global Abort)

Terminate any active operation and cause the controller to revert to the idle state waiting for a new command. Output drivers or ports are not affected. Stepping and position counter update will cease immediately without deceleration. The lack of deceleration can cause mechanical overshoot. The controller will echo a "#" character. This command may not be used within the non-volatile program memory.

Command	Function	Type		NV Bytes
@	Soft Stop	Immediate, Program		1
	Mnemonic	Data 1	Data 2	Result
	(Name) @	None	None	None

@ (Soft Stop)

If moving, decelerate immediately to a stop using ramp parameters. If running a program, when this command is entered, the program will terminate after deceleration. The soft stop may be embedded in a program without causing termination.

An example of this command within a program in conjunction with the Loop on Port command as explained later is:

P 0	Enter program mode.
M 2000	Move at a constant step rate of 2000 SPS.
L0 0	Loop to memory address location 0 until port 1 is low.
@	Decelerate and stop program execution.
P	Exit program mode.

Command	Function	Type		NV Bytes
^C	Reset Controller	None		N/A
	Mnemonic	Data 1	Data 2	Result
	(Name) ^C	None	None	None

^C (Reset)

Resets controller to power-up condition, waiting for start sequence. All outputs are set off, defaults are reloaded from NV memory, and position is set to zero. This command may not be used within the non-volatile program memory. This does not modify the NV memory values.

Command	Function	Type		NV Bytes
A	Read/Write to Ports	Immediate, Program		2, 2
	Mnemonic	Data 1	Data 2	Result
	(Name) A (n)	0-129	None	Port Data

A (Port Read/Write)

This command controls the user input/output ports.

Inputs

The A129 command will read a value between 0 and 7 (or 0 and 15 if port 4 is configured as an input). Output ports are not read.

Ports 1, 2, and 3 (flags 1, 2 and 4) are input ports only[9]. Refer to "l" (lower case L) command for option flag information.

Port 4 is configurable as either an input or output, with jumpers.

Port 4 defined as	JP1	JP2
Input	2-3	1-2
Output	1-2	Open

Port 4 Jumper Settings

Port 4 as an input:

1. Install JP1and JP2.
2. Insure that P4 input option flag 32 ("l" command) is on.

Outputs

[9] See description of the pins in the Hardware section of this document. PLC mode inverts the logic level on these pins.

One or two user outputs are available. They will withstand in excess of 30 Vdc and will sink over 1 amp to a low voltage when turned on. In the off state (after reset and power-up) they have a weak pull up resistor to Vlo.

The PLC option flag 8 has no affect on the output behavior.

Port 4 as an output (as shipped):

1. Install JP1and JP2 in off (storage) position.
2. Insure that P4 input option flag 32 ("I" command) is off.

Port 4 is controlled by the "A 8" (flag 8) command and read back with the "A 129" command. The step pulse is output to P4 when the step/direction out (option flag 2) is set.

Setting option flag 4 will convert P4 to a moving status output.

Port 5 is an output only. It is turned on with the "A 16" command (flag 16).

Port	Flag	Port	Flag
1	1	4	8
2	2	5	16
3	4	129	Read Port

Port Control Using Flags

Port 129 Read Command
Reading the port data (A 129) provides the following result information:

Data	Cause (Standard Mode)	Cause (PLC Mode)
1	Low input present on port 1	High input present on port 1
2	Low input present on port 2	High input present on port 2
4	Low input present on port 3	High input present on port 3
8	Low input present on port 4	High input present on port 4
3	Low on ports 1 and 2	High on ports 1 and 2
15	Low on ports 1, 2, 3 and 4	High on ports 1, 2, 3 and 4

Data Returned From Port 129 Read Command

Output Ports Programming Example
The following example program shows how to turn on an output port. Some uses for this could be illuminating an LED to signal a sequence is complete, or to operate a valve.

P 0	Enter program mode.
A 4	Turn on port 3.
W 60	Wait 600milliseconds.
A 6	Turn on port 2 and 3.
W 10	Wait 100ms.
A 0	Turn off all ports.
P0	Exit program mode.

Note: ports usable for output are determined by the hardware design. Ports 4 and 5 are generally defined as outputs.

Command	Function	Type	NV Bytes
B	Set Jog Speeds	Default, Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) B (n1, n2)	Slow Speed (0-255)	High Speed (0-255)	None

B (Set Jog Speeds)

These two numbers represent the speeds to use for jog inputs. The first is usually a lower speed. The second number is used when the high-speed jog (pin J1-9) is held low. Jog values are multiplied by thirty to determine the step-rate in full-steps-per-second. The D-404 automatically scales step rates to maintain a constant shaft RPM when the micro-step resolution is changed.

Setting values of 0 will disable the jog. Speeds are divided by the parameter defined using the "D" command. The power-up settings are stored in NV memory.

This command is generally implemented during the initial customer default parameter assignment. However, it may be implemented and changed within a program. Following, is an example:

P 0	Enters program mode.
B 0 0	Disable jog switches.
+ 100000	Move 100000 in the plus direction.
W 0	Wait until move is complete.
B 30 100	Re-enable jog switches.
P	Exit program mode.

Command	Function	Type	NV Bytes
b	Set Slow or Fast Decay	Default, Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) b (0, 255)	Speed Threshold		None

b (lower case B; Fast and Slow Decay)

The D-404 has been designed to add both Slow and Fast decay. When there is no motion (stopped), the decay will always be slow. The threshold defines a motor speed where slow decay changes to fast decay during acceleration and switches back to slow decay during deceleration. The threshold will occur at an RPM where step resolution is taken into account.

Threshold Value	½ SPS	1/8 SPS
0	Fast	Fast
25	650	2400
50	1300	4800
100	2400	9600
150	3700	14000
200	5000	19200
255	Slow	

Fast and Slow Decay Threshold

Because the step rate is measured, the decay detection functions for external step pulse input. Slow decay provides smooth operation with increased resonance immunity at slow to medium speeds. Fast decay will generally enhance high-speed operation at speeds above 200-300 RPM. The actual settings should normally be determined empirically (see Addendum: "About Step Motor Current") and tailored to the specific design.

Multiple variables that interact include:

- Operating step speed range
- Step resolution
- Motor size and characteristics
- Load inertia and load damping affects
- Supply voltage
- Motor current setting
- Acceleration and deceleration rates

Once the optimal settings are determined, they will apply to future production, provided the factors remain constant.

Command	Function	Type	NV Bytes
C	Clear and Restore NV Memory	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
(Name) C (n)	0-8	None	Version

C (Clear and Restore NV Memory)

Previously stored programs are erased. Using a 1 forces complete NV memory initialization with factory default values with erasure of all previously stored programs. This must be done when new NV memory is installed or existing memory is corrupted. Frequent use of this command should be avoided, as memory longevity may be impacted over time[10]. The "C 0" command simply reads the last stored values into the working registers.

Command	Function	Type	NV Bytes
D	Divide Speeds	Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) D (n)	Resolution (1-255)	None	None

D (Divide Speeds)

All speeds during ramping and slewing are divided by the parameter specified in the D command (n). The pre-scale number may range between 1 and 255. Speeds as low as 4 1/2 steps per minute may be obtained. As "n" is increased, other parameters (internal speeds) must be increased to obtain a given output step speed.

Using a value of 2 is usually necessary to produce smoother acceleration characteristics at Full and Half step modes. The specified SPS must be doubled to recover the motor shaft speed. D should not be changed while moving at speeds that require ramping. The power-up settings are stored in NV memory.

This command is generally implemented during the initial customer default parameter assignment. However, it may be implemented and changed within a program. Following, is an example:

```
P 0     Enter program mode.
D 10    Change the divider to 10.
P       Exit program mode.
```

[10] This command should not be used within the non-volatile program memory, due to the cyclical nature of a program this command could be executed many times reducing the lifespan of the NV memory.

Command	Function	Type		NV Bytes
F	Find Home	Immediate, Program		3
	Mnemonic	Data 1	Data 2	Result
	(Name) F (n, d)	SPS (40-36,000)	Direction (0,1)	None

F (Find Home)

The special Home algorithm is intended to eliminate mechanical hysteresis typically found in many switches, encoders and is generally present in the form of system mechanical backlash.

The SMC-27X2 microprocessor implements an intelligent homing algorithm whereby home is always approached from the same direction based on the initial logic state of the Home switch and the value (0 or 1) assigned to the "d" direction byte.

Normally Open Home Switch

The Find Home step velocity, using a normally open Home switch (actuation from logic high to low) is programmable over the entire slew velocity available, from 40-36,000 SPS. Once the Home switch is encountered the system inertia typically overshoots the exact switch transition point so that the controller changes the direction signal and shifts the step speed down to the (I) initial parameter velocity. This direction reversal and speed reduction continues until the exact Home switch actuation point is reached and the Homing function is complete.

Normally Closed Home Switch

The Find Home step velocity, using a normally closed Home switch (actuation from logic low to high) will always be the (I) initial velocity parameter setting. Once the Home switch is actuated all motion ceases and the Homing function is complete. The following table illustrates the possible combinations of switch motion:

Home Switch	"d" Parameter	Direction of Motion
Normally Open (High to Low)	0	Negative
Normally Closed (Low to High)	0	Positive
Normally Open (High to Low)	1	Positive
Normally Closed (Low to High)	1	Negative

Homing Direction Behavior

This command may be implemented within a program. Following, is an example:

```
P 0       Enter program mode.
F 1000 1  Find the home switch in the "1" direction at a step rate of 1000 SPS.
P         Exit program mode.
```

Command	Function	Type		NV Bytes
G	Execute Program	Immediate, Program		3
	Mnemonic	Data 1	Data 2	Result
	(Name) G (a, t)	0-192, 256-2048	Trace (0-1)	None

G (Go)

The Go command is used to execute a user programmed sequence starting at the location provided in parameter "a" (Data 1) Most programs will start at "0", however, you may wish to start at another address. When starting at another address the address must begin at a stored instruction address, i.e., "go to" data produces unpredictable results.

If "t" is set (a one), the TRACE mode is turned on. A display of the current step being executed is produced while the program is running. The list format is the same as that of the "Q" command The TRACE mode will be in effect until the program execution terminates or until an embedded 'Go' without the trace attribute is encountered.

The address range is 0 to 2047, depending on NV memory capacity. Address locations between 225 and 255 are reserved for parameter storage and may not be used in programs. The SMC-27X2 also features a special case for the "Go" instruction.

The controller is factory set with the following program example:

P 0	Enter program mode.
+ 1001	Move 1001 steps in the plus direction.
W 100	Wait 100 milliseconds.
- 1000	Move 1000 steps in the minus direction.
W 100	Wait 100 milliseconds.
Z 0	Display step position.
G 0 0	Go to location 0 and run stored program.
P	Exit program mode.

Special case "Go"

If the address is specified as 2048 (above the last NV memory address), the SMC-27X2 will read the input ports, then, branch to an address based on the state of input ports 1 through 4. In such case, each state of the input ports acts as a "go switch." To prevent any confusion to the controller, each address should have a program associated with it even if it is simply "G 2048 0" to go back into the "branch to mode." The target address starts at the second page of program memory, starting at address 256 with 16 character (byte) intervals. This instruction is analogous to "on PORT go to."

Input Port State:				Address of "Go-to":	
P1	P2	P3	P4	HEX	
1	1	1	1	0	256
0	1	1	1	1	272
1	0	1	1	2	288
0	0	1	1	3	304
1	1	0	1	4	320
0	1	0	1	5	336
1	0	0	1	6	352
0	0	0	1	7	368
1	1	1	0	8	384
0	1	1	0	9	400
1	0	1	0	A	416
0	0	1	0	B	432
1	1	0	0	C	448
0	1	0	0	D	464
1	0	0	0	E	480
0	0	0	0	F	496

Input To Branch Address Map

The input ports are internally inverted as part of the address computation. When the PLC jumper is in the STD mode (pins 1 and 2) a low is defined as a voltage above ½ Vlo (no connect). A high is less than ½ Vlo (GND).

When the PLC jumper is in the PLC mode (pins 2 and 3) the inputs are inverted. A low is defined as zero volts (no connect), a high is a voltage above ½ Vlo mode jumper.

Note: Commands in memory locations 129-191 will execute faster than commands in other areas of memory. For timing critical sections of a program it may be helpful to place the program in this are.

Command	Function		Type	NV Bytes
	Set Step Size Resolution		Default, Immediate, Program	2
H	Mnemonic	Data 1	Data 2	Result
	(Name) H (n)	0-5	None	None

H (Step Resolution)

This command selects step size resolution. The SMC-27X2 has an internal lookup table of up to 32 bytes corresponding to 1/8 step counts. This specifies which table is to be used. Each time this command is executed the values are reset to "step 1" and the armature is repositioned to the start phase. Initial and final velocities may require appropriate changes.

The H command sets the phase switching sequence:

Sequence	H Command	Steps per rev. (1.8 deg) Motor)	Remark
1/8 Micro	H 0	1600	Highest resolution smoothest
1/4 Micro	H 1	800	Resonance reduced or higher resolution
Half	H 2	400	High torque - 1 phase on/2 phase on
Full	H 3	200	Highest torque - 2 phase on
1/2 Micro	H 4	400	Low torque - half step
Wave	H 5	200	One phase on - full step

Step Resolution Definitions

This command is generally implemented during the initial customer default parameter assignment. However, it may be implemented and changed within a program. Following, is an example:

P 0 Enter program mode.
H 1 Change the resolution to ¼ step.
P Exit program mode.

Command	Function		Type	NV Bytes
	Set Initial Velocity		Default, Immediate, Program	3
I	Mnemonic	Data 1	Data 2	Result
	(Name) I (n)	SPS (40-36,000)	None	None

I (Initial Velocity)

This parameter sets the initial velocity in steps per second. This is the first speed used at the beginning of acceleration. It must be slow enough that the motor can start without loosing steps or stalling.

The initial velocity applies to:

1. All index commands (+, -, R).
2. First execute in constant velocity.
3. Decelerate to 0 in constant velocity or soft stop.
4. Final phase in home command if home speed is above initial velocity.

See "Default Table" in the beginning of this section.

This command is generally implemented during the initial customer default parameter assignment. However, it may be implemented and changed within a program. Following, is an example:

P 0 Enter program mode.
I 100 Change the initial velocity to 100 SPS.
P Exit program mode.

Command	Function	Type		NV Bytes
i	Restart Special Trip	Default, Program		5
	Mnemonic	Data 1	Data 2	Result
	(Name) i (n)	Next Trip Position ±8,388,607	Port (0-63)*	None

i (lower case I; Restart Special Trip)

See lower case "k" command. *Actual values are determined by the hardware configuration.

Command	Function	Type		NV Bytes
J	Jump to Address	Program		4
	Mnemonic	Data 1	Data 2	Result
	(Name) J (a, n)	Address (0-2047)	N + 1 Times 0-255	None

J (Jump to Address a, n+1 times)

This loop command allows repetition of a sequence up to 256 times. The address specified MUST be a valid instruction address, and is usable only within a program. This instruction may not be nested, because only one jump counter is available for use at any given time.

Example:

P 0 Enter program mode.
+ 1000 Move in the plus direction 1000 steps.
J 0 3 Go to and run command at location 0, 4 times.
P Exit program mode.

Command	Function	Type		NV Bytes
K	Set Ramp Slope Time	Default, Immediate, Program		3
	Mnemonic	Data 1	Data 2	Result
	(Name) H (n1, n2)	Accel (0-255)	Decel (0-255)	None

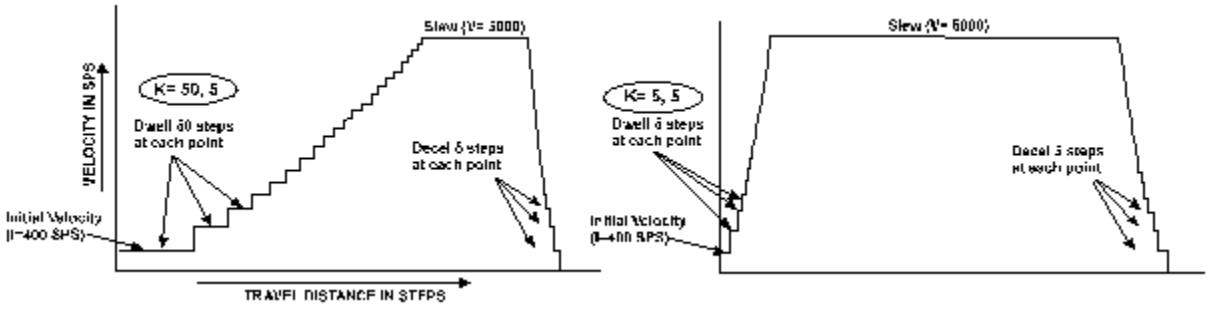
K (Ramp Slope)

Specify the ramp acceleration and deceleration time. The "K" command is used to adjust the ramp slope during the motor acceleration or deceleration. An internal lookup table defines the profile or shape of the acceleration/deceleration curve. Depending on the values of initial and slew velocities, a number of discrete velocities are used to define the acceleration or deceleration of the motor armature rotation.

The "K" value determines how many steps are made at each step rate point on the acceleration curve during ramping. Higher "K" values will increase the dwell time at each discrete point on the acceleration ramp. Lower values of "K" will increase the acceleration rate. A value of 0 will eliminate any ramping.

In practical applications, it is typically easier to decelerate a system, rather than accelerate a system. The separate decelerate parameter feature is a valuable time saver when compared to systems with fixed acceleration/deceleration times.

The following two examples are of ramped indexes, each 2000 steps with I=400, V=5000, but different "K" values; K50 5 and K5 5:



Acceleration and Deceleration Example

Note: The default value of "K" is 5 (Accel), 5 (Decel). To modify the ramp slope it is always necessary to enter two (2) data values (from 0 to 255), corresponding to the desired slope for motor acceleration vs. deceleration. The value of "K" can be proportionally changed if the micro step resolution (H command) or Divide Speed (D command) is increased.

The K command can be issued:

1. As part of a setup.
2. In an application program.
3. As User defined defaults at reset.

Following is an example of changing the K parameter inside a program:

P 0	Enter the program mode.
K 100 50	Change the acceleration ramp to 100 and the deceleration ramp to 50.
P	Exit program mode.

Command	Function	Type	NV Bytes
k	Next Trip Point, Port Output	Default, Program	5

Mnemonic	Data 1	Data 2	Result
(Name) k (n)	Next Trip Position ±8,388,607	Port (0-63)*	None

k (lower case K; Trip Output Value)

*Actual values are determined by the hardware configuration.

The latency described in use of the "T" command can be avoided via use of the "i" and "k" (both lower case) commands. Both of these commands implement a trip mode similar to the T command, but there actions are performed in real time.

The following is an example of using the lowercase k and I command in a program:

P 0			Start programming mode.
	0	O 0	Set position to zero.
	9	+ 6000	Index 6000 steps.
	13	W 0	Force wait till index complete.
	16	P 0	End program.
P 128			Program RAM commands.
	128	k 400 0	Set new trip at 400 and turn ports off.
	133	k 600 16	Set new trip at 600 and turn port 5 on.

	138	i 200 8	Reset origin, RAM=128, port 4 on.
	143	P 0	End program.
S			Save the shadow RAM program.
G 0			Execute program.

Failure to store the program in shadow memory will result in loss of all commands between 128 and 192. Once they are stored, they will automatically reload with every reset. In general programs will automatically be stored in NV memory except for portions of programs that lie between memory segments 128 and 191. The S command insures these areas written correctly.

The following example further describes the program sequence:

Address		Description
0	"O 0"	The position counter is reset to zero.
4	"k 200 8"	The initialize command "k" is first used to initialize the real time sequence. Assume that the command "k 200 8" is executed at the beginning of the program. The following actions take place: 1. Port 4 is set on per data2 – see the "A" command. 2. The first trip position is set per Data1 (200). 3. A special trip program counter (PC) is set to 128.
9	"+6000"	Now the +6000 index command is started. When the position matches 200, the command located at 128 is checked for either a "i" or "k."
128	"k 200 0"	The "k" changes the trip position to 400 and turns all output ports off (high logic voltage) This is executed while at the exact 400 position. The program counter is advanced to 133.
133	"k 600 16"	This is executed like the previous 128, new trip=600, ports 4 and are turned on and the program counter set to 138.
138	"i 200 8"	The restart command performs the same actions as the initial k 200 8 command. Port 4 is turned on, the trip is set to 200 and program counter is reset to 128.

Command	Function	Type	NV Bytes
L	Loop on Port	Program	3

Mnemonic	Data 1	Data 2	Result
(Name) L (a, c)	0-2048	Condition (0-9)	None

L (Loop on Port)

Loop on Port will test the specified input port for the required condition (c). If the port is not at the required level then the program will jump to the specified address. If the address is from a previous instruction then the program will loop until it becomes the specified level. The program will then continue to the next step. The PLC mode inverts the input high/low definitions.

Input ports are available as follows:

Port	Low	High
1	0	1
2	2	3
3	4	5
4	6	7
5	8	9

Loop on Input Port Values

The SMC-27X2 can view all ports as inputs and outputs, restricted by contention with external hardware. Any "output" port can be modified, then subsequently used in conjunction with the L, G 2048, or A129 (read) command. The SMC-27X2 has an additional feature of implementing a "wait till" function. The standard loop tests the condition every 2-3ms. If the unique address is 2048, the controller executes a tight loop at this instruction, monitoring the specified condition. When the condition is met, program execution continues.

This feature is helpful in situations where the condition may be of short duration. This command is usable only in NV memory program execution. Following is an example of this command:

P 0	Enter program mode.
L0 4	Stay at location 0 until port 3 is low then go to next command in program.
+ 1000	Move 1000 steps in the plus direction.
P	Exit program mode.

Command	Function	Type	NV Bytes
	Hardware Options	Default, Immediate, Program	2
l	Mnemonic	Data 1	Data 2	Result
	(Name) l (a, d)	Option Flags		None

l (lower case L; Option Flags)

IMPORTANT – Do not change jumpers with power applied to the D-404!

This command configures several options, primarily relating to input/output operating modes and defining external hardware. Several options invert the sense of input signals. If appropriate jumpers or input signal inversion is not matched to the selected option(s) then the system may lock up or motion can be inhibited.

Flags and Numbers

Several commands use "on-off" flags to enable or disable some feature. The data supplied is in decimal ranging between 0 and 255. The corresponding binary bits are called flags. There are 8 flags, each equaling binary values of 1, 2, 4, 8, 16, 32, 64, 128.

Two or more flags can be combined arithmetically for multiple options, i.e., flag 1 + flag 8= 9. So, command "l 9" will invert limits and set the PLC mode.

Note: When reading flags, host software must convert decimal to binary to evaluate the results.

Flag	Bit	Mode	Function	Note
1	0	Invert limits	Both inputs must be held low to allow a move.	
2	1	Step/dir out	Change port 4 and 5 into step and direction out.	P4 is output (JP1, JP2)
4	2	Moving	Change port 5 into moving signal.	P4 is output (JP1, JP2)
8	3	PLC mode	Inverts all input signals.	Set PLC jumper
16	4	P4 input	Inform the controller that port 4 is an input.	Based on JP1, JP2
32	5	Gentle limit	Soften limit with decelerate (K).	Over shoots limit
64	6	Reserved		
128	7	Read back	Read out the values.	

Mode Flags

An "S" (save) command must be used to preserve the settings.
The CLEAR command sets all flags to 0.

Limit Polarity (Flag 1)
The input levels on the travel limit sensors are inverted, allowing source type sensors such as hall-effect devices to be used. This command cannot swap the limit directions. When this bit is set, motor travel in either direction is inhibited unless the appropriate limit inputs are forced low. The PLC mode will invert the logic levels (sourcing input necessary to trigger a limit).

Example:
- I 1 Invert the limit software.
- I 9 PLC - invert limit software and input logic.

Make Step and Direction Outputs (Flag 2)
This option converts port 4 to a step output and port 5 to a direction output. These signals can be directed to "slave" driver(s) to control additional motors if required. The step outputs are short negative going pulses.

Port 4 logic must be configured (JP1, JP2) as an output. The step and direction outputs can be a very powerful feature as it allows the user to send step pulses and a direction bit to a stand-alone driver, moving it at the same rate and direction.

Conflicts - Flags 4 and 16 must be off.

Moving Output (Flag 4)
This flag converts output P5 into a moving status signal. While motion is active, port 5 will be at a zero volt level. Several axes may be wire-or'ed together to form an "any axis moving" signal.

Note: Conflicts - Flag 2 is not allowed.

PLC Mode (Flag 8)
This option tells the controller that all inputs are inverted, and to compensate accordingly. The PLC jumper must be installed for this flag to respond. The jumper reconfigures the input hardware with pull-down resistors rather than pull-ups. A positive voltage must be applied to any of the affected inputs (such as GO) to trigger the "true" condition. If both GO and Soft Stop are true at power up, then a conflict message (single mode) is displayed and flag 8 will be temporally set to prevent a latch up condition.

Port 4 Input (Flag 16)
This flag informs the controller that Port 4 is to be used as an input. Jumpers JP1 and JP2 must be set accordingly[11]. The commands; read inputs (A129), branch (G 2048) and loop (L n) will use this input port.

[11] See Hardware section for JP1 and JP2 for further information.

Conflicts - Flag 2 and Flag 4 (outputs) are not permitted.

Example- turn on 3 options:
"l 25" = 1+8+16 = (invert limits) + (PLC mode) + (port 4 = input).

In general, an improperly installed jumper will not cause damage other than a malfunction.

Gentle Limit (Flag 32)
When a limit input is activated, the motion (presumably high speed) will decelerate to zero based on the deceleration value defined by the "K" command. This means that there will be some travel overshoot. The mechanical design should take this into account.

Command	Function	Type	NV Bytes
M	Move at Constant Velocity	Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) M (s)	SPS (±40-36,000)	None	None

M (Move at a Constant Velocity)

The "+" or "-" sign determines direction during the move at constant velocity function. The motor will ramp up or down to a constant velocity. Motion will continue at the given speed until a new velocity is entered. The specified slew speed is in steps per second. Ramp parameters may be modified prior to each velocity command, allowing different ramp slopes. The sign preceding the velocity specifies the direction (+-). The SMC-27X2 has the capability of decelerating from full speed in one direction, then accelerating to full speed in the opposite direction with this single command.

Motion may be terminated by:
1. The "M 0" command
2. Soft stop command or interrupt
3. Abort (ESC) interrupts (without deceleration)

The default initial velocity is used at the first invocation of the command. The following commands modify effective speeds and resolutions:
4. Divide
5. Ramp factor
6. Step Resolution

An example of this command within a program, in conjunction with the Loop on Port and Soft Stop commands, is as follows:

P 0	Enter program mode.
M 2000	Move at a constant step rate of 2000 SPS.
L0 0	Loop to memory address location 0 until port 1 is low.
@	Decelerate and stop program execution.
P	Exit program mode.

Command	Function	Type	NV Bytes
O	Set Origin	Immediate, Program	4

Mnemonic	Data 1	Data 2	Result
(Name) O	Position (±8,388,607)	None	None

O (Set Origin)

This command sets the internal 24-bit position counter to the specified value. Zero position for the relative mode is "0000." Signed numbers are used. Hardware reset clears to "0000." The position counter is incremented or decremented for all motion commands. The position counter is used only for trip value comparison. This counter may be changed without affecting the distance of travel in process.

This command may be implemented within a program. It is very useful when used in conjunction with the Find Home and Relative Positioning commands. Following, is an example:

```
P 0       Enter program mode.
F 1000 1  Find the Home switch in the "1" direction at a step rate of 1000 SPS.
O 0       Set origin and counter to 0.
R 1000    Move to position 1000 relative to 0.
P         Exit program mode.
```

Command	Function	Type	NV Bytes
P	Program Mode On/Off	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
(Name) P (a)	Address (0-2047)	None	None, #

P (Program Mode)

The P command is always used in pairs. The first "P" initiates the program mode at the specified address. Once in this mode all commands and data are directed into the NV memory for future execution. Entering the second "P" command will terminate the PROGRAM mode, and then insert an end of program marker (0FFh) in the stored program. The controller will then return to the COMMAND mode.

The program mode may also be terminated with the Escape key, causing immediate return to command mode without inserting the end of program marker. This is useful for editing sections of the program, without requiring that all commands be re-entered.

More than one program may exist at different addresses. These commands can than be executed via the "G (address)" command. There are special address ranges that are assigned to various functions:

Address	Function
128-191	Fast "shadow" RAM
256-511	"G 2048" command
1600	Power-up routines

Special Address Locations

Command	Function	Type	NV Bytes
Q	List Program	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
(Name) Q (a)	Address (0-2047)	None	Listing

Q (List Program) (Note*: Use in dumb terminal, single line mode).*

List program stored in non-volatile memory using the format:

Address Instruction Value 1 Value 2

The values will be displayed only if applicable to the particular instruction type. Twenty instructions are displayed at a time. Use the <CR> key to list up to 20 more commands without pause. ESC quits and any other key single steps the listing.

Command	Function	Type		NV Bytes
R	Index Relative to Origin	Immediate, Program		4
	Mnemonic	Data 1	Data 2	Result
	(Name) R (n)	Position (±8,388,607)	None	None

R (Index Relative to Origin)

Move, with ramping, relative to the "0" origin. The target position has a range of ±8,388,607 steps from the '0' origin.

The motion sequence is:

1. Wait until any previous motion is finished,
2. Read the current position then calculate the distance to the new target position,
3. Energize the motor winding,
4. Start stepping at the rate of the initial velocity (I),
5. Accelerate using a profile defined by the fixed table that approximates straight-line acceleration and a slope set by the "K" command,
6. The acceleration continues until the slew speed as specified by the "V" command is attained,
7. Motion continues at the slew speed, until the deceleration point is reached,
8. Decelerate (determined by the second "K" value) to a stop completing the index,
9. If another index is not commanded for the settling period, power down the motor (if auto power down is enabled).

This command may be implemented within a program. It is very useful when used in conjunction with the Origin command. Following, is an example:

P 0	Enter program mode.
O	Set origin and counter to 0.
R 1000	Move to position 1000 relative to 0.
P	Exit program mode.

Command	Function	Type		NV Bytes
S	Save Parameters to NV Memory	Immediate		1
	Mnemonic	Data 1	Data 2	Result
	(Name) S	None	None	None

S (Save)

The following parameters are saved in the NV memory and will be recalled as defaults during power-on reset:

1. NV memory addresses 128 through 191 (shadow RAM)
2. Initial velocity (I)
3. Slew velocity (V)
4. Divide factor (D)
5. Ramp slope (K)
6. Jog speeds (B)
7. Resolution mode (H)
8. Auto power down (E)
9. Limit polarity (H)

10. Name (for party line use)
11. Trip point settings

All of these parameters are saved as a block from the working registers in the SMC-27X2. Frequent use of this command should be avoided, as memory longevity may be affected.

Command	Function	Type		NV Bytes
T	Set and Enable Trip Point	Default, Program		4
	Mnemonic	Data 1	Data 2	Result
	(Name) T (n)	Position (±8,388,607)	Address (0-255)	None

T *(Trip Point)*

During motion operations, the position counter is continuously updated. If the trip point function is enabled, the position is continuously compared to the programmed trip position. When equality is detected, a trip event will be triggered. If a program is running, a call or "Go Sub" will be made to the specified address between 1 and 255.

Programs located at the specified address can perform almost any function, including turning on/off ports and setting new trip points. A trip point cannot be "reentered" when executing a trip subroutine and a new trip is set as part of the routine. A new trip cannot be triggered until the end of the first trip routine. Routines located between 128 and 192 will execute faster because of the "Shadow RAM" feature. In general practice is not recommended to use routines that contain index, wait or time consuming instructions[12].

Disable

To turn off the trip function, use 0 (zero) as the address parameter. The trip is not currently usable in the encoder mode. The following is an example (all commands are followed by a <CR>):

1. Write program to location 0 (zero).

P0		Enter program mode at address 0.
0	A8	Turn port 4 on.
2	+2000	Rotate motor 2000 steps in the plus direction.
6	P0	Exit program mode.

2. Write program to location 100.

P100		Enter program mode at address 100.
100	A129	Read port states.
102	A0	Turn port 4 off.
104	P0	Exit program mode.

3. Set Trip Point.

In “dumb terminal” mode enter T1000 100. This tells the controller to run the program located at address 100 when the step position is 1000.

4. Run program.

Enter the “G” command. Port 4 will turn on and the motor will start moving. When the motor position is at 1000, the program will vector to address 100 and run that sequence. The number 8, signifying port 4, will appear on the screen.

[12] It is perfectly acceptable to use long period instructions, but the speed advantage in this region is lost with these commands.

Command	Function	Type	NV Bytes
V	Set Final (Slew) Velocity in SPS	Default, Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) V (n)	SPS (40- >36,000)	None	None

V (Set Slew Speed)

This is the maximum speed to be used after acceleration from the initial velocity.

Note: If Slew speed is not reached after setting this parameter the speed may be limited by the motor capability and/or power driver circuitry.

The final output velocity is divided by the value of the parameter set using the D command. This value is independent of constant velocity (M), jog (B) or home (F) speeds and is used when indexing absolute or relative (+, -, R commands).

If full or half step mode is chosen, the acceleration time may become to fast for larger motors. This is due to the high speed SMC27X2 microprocessor. A "D 2" pre-scale divider may be required to provide smoother acceleration characteristics.

Example

Assume that the desired running speed is 12,000 full steps per second (3600 RPM). The speed ("V") can be set to 24,000 SPS with D = 2.

Thus:
SPS (motor) = V/D or 24000/2 =12,000 SPS

With the "D 2" divider, the full output speed range is approximately 20 to 20,000 SPS.

See "Default Table."

This command is generally implemented during the initial customer default parameter assignment. However, it may be implemented and changed within a program. Following is an example:

P 0	Enter program mode.
V 10000	Change the slew velocity to 10000 SPS.
P	Exit program mode.

Command	Function	Type	NV Bytes
W	Wait (n) Milliseconds	Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) W (n)	10 ms. (0-65,535)	None	None

W (Wait)

The controller will remain in an idle state for the specified time (n). The Wait command, if issued while indexing (as a result of an R, +, -, or F command), will not start until the motion has completed.

Wait until motion complete

Using this command with zero time can provide an alternate method of determining motion. If issued while running at constant velocity, the time-out will occur without waiting for motion to cease. High-speed step operation during Wait commands will increase the delay time by as much as 14 times the normal value. The result will not be available until the delay is complete.

The following example program makes a move, waits for motion to complete, then turns on an output port. Some uses for this could be illuminating a LED, signaling a sequence is complete or operating a valve.

```
P 0       Enter program mode.
+ 1000    Move 1000 steps in the plus direction.
W 0       Wait for move to finish.
A 8       Turn on port 4.
W 500     Wait 50milliseconds.
A 0       Turn off port.
P 0       Exit program mode.
```

Command	Function	Type	NV Bytes
W	Wait (n) Milliseconds	Immediate, Program	3
	Mnemonic	Data 1	Data 2	Result
	(Name) w (n)	10 ms. (0-255)	None	None

w (lower case W; energize)

The "w" command is a "energize" command that can insure that motor current has built up when a step command is executed while in the "holding current" state. This parameter is useful under certain conditions, and should be zero (off) if possible. The following conditions might require a charge time:

1. Automatic current setback ("E" command) is in effect.
2. If delays between consecutive motion commands exceed 1 second.
3. If the initial speed ("I" command) specified is too high.
4. If the acceleration requirements are excessive.

When the auto current setback is used, motor current will be reduced after approximately one second of idle time. When a new motion command (+, -, R, M, F, etc) is executed the windings are energized almost instantly with the first motor step. Thus the motor must rapidly change from a "relaxed" position to the next step.

This energize command insures a delay after turn on. The delay is not used if the setback timeout (from a prior motion) is not timed-out. That is, the motor current is still at 100%.

Note: This function can introduce a substantial start delay.

The controller will remain in an idle state for the specified time. The Wait command, if issued while indexing (as a result of an R, +, -, or F command), timing will not start until the motion has completed.

NV default =0

The following example program pre-energizes the controller to the run current then makes a move.

```
P 0       Enter program mode.
w 25      Pre-energize the controller for 250 milliseconds.
+ 1000    Move 1000 steps in the plus direction.
P 0       Exit program mode.
```

[13] These delays will depend on the size and various other parameters of the motor connected to the D-404

Command	Function	Type	NV Bytes
X	Examine Settings	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
X	None	None	Display Setting

X (Examine)

The Examine command produces two responses, depending on the mode of operation. When not in the multi-axis mode (non-daisy chain or party line) the display is as follows:

X K= 5/5, I= 400, V= 5016, D= 1, ½, n=C

Where:

K= Ramp up/ramp down
I= Initial velocity
V= Slew velocity
D= Divide factor
½= Resolution mode
n= Axis name

In the multi-axis (daisy chain or party line) mode the data is returned in the following format:

mm[LF]
mm= model (26)

Command	Function	Type	NV Bytes
Y	Program Hold and Run Current	Default, Immediate, Program	3

Mnemonic	Data 1	Data 2	Result
(Name) Y	Hold 0-100	Run 0-100	

Y (Hold and Run Current)

This command allows specifying the Hold and Run values of motor current (per phase) between 0 and 100% with a resolution of 1%. The value 100% represents a maximum of 4 amps per phase. The switching between Hold and Run values is automatic whenever a motion function is executed. Current reduction to the "Hold" value is automatic and occurs when stationary. A settling time (nominally 1 second) is inserted after each move is completed.

The maximum current is limited to a value that will not consume more than the 100VA and is available from your power supply.

The following procedure is used to access the independently programmable "Hold" and "Run" current feature:

1. Issue the "Y" command to program the desired current values. Entering Y10 80 yields a 10% Hold current and 80% Run current.
2. Issue an "S" (Save) command. The values are now stored in non-volatile memory.

On receipt of an index or other motion command, the control circuits are incremented to the 80% boost, while moving is in process. On completion of motion (and settling time delay) the current is automatically reduced to the 10% Hold current level.

Quiet Mode

To access the "Current Disable" feature use the following current setup procedure:

Enter	Result
Y00 80	Initiates current program mode, sets Hold current at 0% and Run current at 80% of maximum. The 4-amp drive would be set for 3.2 amps peak.

Note: Refer to "About Step Motor Current" in the Addendum for more detail on setting the proper motor current.

Command	Function	Type	NV Bytes
Z	Read and Display Current Position	Immediate	1

Mnemonic	Data 1	Data 2	Result
(Name) Z	Readout Mode (0-1)	None	Position

Z (Read Position)

During motor move commands the value will change depending on the direction of travel. The counter is programmable by the "O" command.

The SMC-27X2 has the option of continuous readout via the serial interface. The "Z 1" command enables this operation. Any change in position causes the position data to be sent to the serial output. The readout will be terminated by a carriage return.

The readout mode will be defaulted as "On" if a SAVE command is issued. This mode is only practical using single axis protocol.

The controller is factory set with the following program example:

P 0	Enter program mode.
+ 1001	Move 1001 steps in the plus direction.
W 100	Wait 100 milliseconds.
- 1000	Move 1000 steps in the minus direction.
W 100	Wait 100 milliseconds.
Z 0	Display step position.
G 0 0	Go to location 0 and run stored program.
P	Exit program mode.

Command	Function	Type	NV Bytes
[	Read NV Memory	Immediate	N/A

Mnemonic	Data 1	Data 2	Result
(Name) [(a, b)	Address (0-2047)	Sequential Bytes (0-255)	Displayed Values

[(Read NV Memory)

The user may display any byte of the 2047 byte external NV memory. The address specifies the desired location to access. The user may also specify how many bytes of data to read up to 255 bytes from the starting address. All data returned by this command is in decimal format

Note: At addresses 128-191 the NV memory is read instead of values from RAM.

This command is normally used external from the controller's non-volatile program. Example:

[0 20	The result from this command would be 20 sequential bytes starting at location 0 and finishing at location 19.

Command	Function	Type	NV Bytes
]	Read Limits, Hardware	Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name)]	0-1	None	Status

] (Read Limits, Hardware)

This command allows the user to examine the status of the various switch inputs. The result will contain the state of the limit switch inputs and current phase outputs in binary values as follows:

Decimal value:	128	64	32	16	8	4	2	1
Bit position:	7	6	5	4	3	2	1	0
SMC-27X2:	Lb	La	Hm	P5	P4	P3	P2	P1

Switch Input Values using ']'

Where:
La = Limit "a" switch
Lb = Limit "b" switch
Hm= Home switch (32 = low input)
P 0-5= Ports 1-5 (see "A" command)

"] 1" Read other inputs:
This command reads other inputs; some of which can be used by external applications under the condition that the SMC-27X2 does not use them. For instance, if the jog speeds are set to zero, the three jog inputs could be used as general-purpose inputs.

Decimal value:	128	64	32	16	8	4	2	1
Bit position:	7	6	5	4	3	2	1	0
SMC-27X2:	JOG-S	JOG2	JOG1	*	*	*	*	*

Jog Values

The returned number is converted to a byte, each bit represents the state of the specified data.
This command is normally used external from the controller's non-volatile program. Example:

] 0 The result from this command would be in decimal format indicating what I/O port is active.

Command	Function	Type	NV Bytes
+	Index in Plus Direction	Immediate, Program	4

Mnemonic	Data 1	Data 2	Result
(Name) + (n)	Steps (0-16,777,215)	None	None

+ (Index in Plus Direction)

Step in the positive direction for the specified step count.

The motor will ramp up, slew, and then ramp down per the previously set parameters. The range is 0 to 16,777,215. The position counter will overflow at 8,388,607.

The motion sequence is:
1. Wait until any previous motion is finished,

2. Energize the motor winding as required,
3. Start stepping at the rate of the initial velocity (I),
4. Accelerate using a profile defined by the fixed table that approximates straight-line acceleration and a slope set by the "K" command,
5. Accelerate until the slew speed, as specified by the "V" command, is attained,
6. Motion continues at the slew speed, until the deceleration point is reached,
7. Decelerate (determined by the second "K" value) to a stop completing the index,
8. If another index is not commanded for the settling period, power down the motor (if auto power down is enabled).

The D-404 is factory set with the following program example:

P 0	Enter program mode.
+ 1001	Move 1001 steps in the plus direction.
W 100	Wait 100 milliseconds.
- 1000	Move 1000 steps in the minus direction.
W 100	Wait 100 milliseconds.
Z 0	Display step position.
G 0 0	Go to location 0 and run stored program.
P	Exit program mode.

Command	Function	Type	NV Bytes
	Index in Minus Direction	Immediate, Program	4
-	Mnemonic	Data 1	Data 2	Result
	(Name) – (n)	Steps (0-16,777,215)	None	None

- (Index in Minus Direction)

Same as "+" command with reverse rotation.

Command	Function	Type	NV Bytes
	Read Moving Status	Immediate, Program	1
^	Mnemonic	Data 1	Data 2	Result
	(Name) ^	None	None	Status

^ (Read Moving Status)

The host may use this command to determine the current moving status that exists within the SMC-27X2. A non-zero value indicates moving.

This command is normally used external from the controller's non-volatile program. The command would look like:

^ The result from this command would be a decimal number. Any number other than 0 indicates the controller is moving.

Command	Function	Type	NV Bytes
	Write to NV Memory	Immediate	N/A
\	Mnemonic	Data 1	Data 2	Result
	(Name) \ (a, d)	Address (0-2047)	Data (0-255)	None

\ (Write to NV Memory)

This command allows the programmer to modify any location in the memory. The command being changed must be done so in decimal format. Special step sequences may be entered, and all initialization constants may be changed. (Reference "Memory Map" in the beginning of this section for specific locations).

The life expectancy of the NV memory may be affected by this command. This command complements the Read NV Memory ([ ) command. Addresses 128-191 in the NV memory are always written to (not the RAM).

This is a very powerful command and care must be taken not to overwrite other needed sections of the nonvolatile program. The example of this command is as follows:

EAD controllers are factory set with the following program example:

P 0	Enter program mode.
+ 1001	Move 1001 steps in the plus direction.
W 100	Wait 100 milliseconds.
- 1000	Move 1000 steps in the minus direction.
W 100	Wait 100 milliseconds.
Z 0	Display step position.
G 0 0	Go to location 0 and run stored program.
P	Exit program mode.

To change the G 0 0 section of the program to the soft stop command @. When you query the program it shows the G command at location 16. The decimal value for the @ command is 0. So, to change the G 0 0 command to a soft stop you would enter the following command line:

\ 16 0

Command	Function	Type	NV Bytes
\	Terminate Program	Immediate	2

Mnemonic	Data 1	Data 2	Result Status Byte
(Name) \			

| (Selective Termination)

This command (pipend: Shift+\) can be placed at a point to terminate (equivalent to ABORT) the program that was started via the "G" command or hardware GO input.

Note: The Selective Termination command may be used to individually "ABORT" a single axis in multiple axis systems, when the global "ESC" command is not appropriate.

Program Example

```
P 0
        0 O             Set Origin To Zero
        1 I     400     Initial SPS
        4 T     1000 128 Set a Trip
        9 M     5000    Start Motion
        12 I    1000    Change Initial SPS
        15 V    1000    Slew
        18 G    18      Wait
        21 P    0       End Program Flag

P 128                   Trip Routine Start
```

```
                128        +2000        Decelerate and Index
                133 W      0            Wait
                136 Z                   Show Position
                138 (axis name)         Abort Program (where "axis name" is the selected axis to
                                        terminate)
                140 P      0            End Program Flag
          S                             Save Code at 128
```

Issue "GO": The sequence will complete and show the position that the number of extra steps is due to (decelerate) ramp plus a few steps of overhead. The overstep difference is repeatable.

Command Summary

MNEMONIC / COMMAND		DATA 1	RANGE 1	DATA 2	RANGE 2	NV	D	I	P
+	INDEX IN "+" DIRECTION	STEPS	1- 16,777,215			4		⊙	⊙
-	INDEX IN "-" DIRECTION	STEPS	1- 16,777,215			4		⊙	⊙
ESC	ABORT/TERMINATE							⊙	
@	SOFT STOP					2		⊙	⊙
^C	SOFTWARE RESET								
[	READ NV MEMORY	ADDRESS	0-2047*	NUMBER	0-255			⊙	
\	WRITE TO NV MEMORY	ADDRESS	0-2047*	DATA	0-255			⊙	
]	READ LIMITS/HARDWARE	LIM/HW	0-1					⊙	⊙
^	READ MOVING STATUS							⊙	⊙
|	SELECTIVE TERMINATE					2		⊙	⊙
A	PORT R/W	BINARY	0-128			2		⊙	⊙
B	SET JOG SPEEDS	SLOW	0-255	HIGH	0-255	3	⊙	⊙	⊙
b	FAST AND SLOW DECAY	DECAY MODE	0-255			2	⊙	⊙	⊙
C	CLEAR AND RESTORE	PAGE	0-9					⊙	
D	DIVIDE STEP RATE	DIVIDER	0-255			2	⊙	⊙	⊙
E									
F	FIND HOME	SPS	40-36000	DIRECTION	0-1	3		⊙	⊙
G	GO	ADDRESS	0-2048*	TRACE	0-1	3		⊙	⊙
H	RESOLUTION MODE	TABLE#	0-5			2	⊙	⊙	⊙
I	INITIAL VELOCITY	SPS	40-36000			3	⊙	⊙	⊙
i	RESTART SPECIAL TRIP	NEXT TRIP	±8388607	PORT	0-63	5	⊙		⊙
J	JUMP	ADDRESS	0-2048*	N+1 TIMES	0-255	4			⊙
K	RAMP SLOPE	ACCEL	0-255	DECEL	0-255	3	⊙	⊙	⊙
k	TRIP OUTPUT VALUE	NEXT TRIP	±8388607	PORT	0-63	5	⊙		⊙
L	LOOP ON PORT	ADDRESS	0-2048*	CONDITION	0-8	4			⊙
l	INVERT LIMIT/STEP-DIR	OPTIONS	0-255			2	⊙	⊙	⊙
M	MOVE AT CONST. VEL.	SPS	±36000			3		⊙	⊙
O	SET ORIGIN	STEPS	±8388607			4		⊙	⊙
P	PROGRAM MODE	ADDRESS	0-2047*					⊙	
Q	QUERY PROGRAM	ADDRESS	0-2047*					⊙	
R	INDEX TO POSITION	POSITION	±8388607			4		⊙	⊙
S	STORE PARAMETERS							⊙	
T	TRIP POINT	POSITION	±8388607	VECTOR	0-255	4	□		□
U	RESERVED								
V	SLEW VELOCITY	SPS	40-36000			3	⊙	⊙	⊙
W	WAIT, (DELAY)	0.01 SEC	0-65535			3		⊙	⊙
w	PRE ENERGIZE	0.01 sec	0-255			2	⊙	⊙	⊙
X	EXAMINE PARAMETERS							⊙	
Y	RUN AND HOLD CURRENT	HOLD	0-100	RUN	0-100	3	⊙		⊙
Z	DISPLAY POSITION	CONTINUE	0-1					⊙	

Command Listing

N – How much space in bytes of memory the command uses.
I – Immediate command.
P – Can be executed in program mode.

* Program address ranges are 0-192, 256-2047, 2048 is used as a special case, 128 – 192 is high speed "shadow" RAM.

Application Notes

Note that the minimum step rate is now 40 SPS. Slower step rates are obtained using the "D" (divide) command.

The I, V, and M commands now have a range of 40 to 36,000 SPS. Values entered below 40 SPS will be limited to approximately 40 SPS, Values above 36,000 SPS are accepted, and the SPS will increase to a 42,000 SPS ceiling and the acceleration ramp will become non-contiguous. As with all models, the top speed is reduced when trip points are introduced.

Full and Half Step Recommendations

The acceleration ramp profile is slightly modified (over previous designs) to accommodate the increased dynamic range. Consequently, the number of acceleration points that would be used with full and half stepping are reduced from previous low speed versions. This may manifest itself in a tendency to stall during ramp-up. Using the command "D2" (divide all rates by 2) may provide a smoother acceleration – deceleration. To obtain an equal shaft RPM, your specified speeds must be doubled, i.e., "V 5000" becomes "V 10000." Other D values can be used, as long as the V (SPS) value is in range. This technique produces a longer ramp slope thus allowing the "K" command values to be reduced.

FAQ
Q. When I set the SPS values, the X command reads back a different value?
A. The controller uses integer values for determining the "time per step".

Decay Control

Fast decay mode regulates motor current by varying the duty cycle and applying full negative DC bus voltage to the motor windings when OFF. Applying full DC bus voltage in this manner causes the motor winding current to change at a rapid rate; hence 'fast decay." The fast decay mode of operation permits better current regulation, but increases motor heating due to the higher current transients.

Slow decay mode also regulates motor current by varying the duty cycle, but instead, shorts the motor windings when OFF (i.e. zero volts vs. maximum DC bus voltage). The slow decay mode of operation permits higher currents (higher torque) at lower speeds with less motor heating; but also exhibits poor low current regulation.

Moving signal
Some products have a "moving" signal available on the serial connector. The signals from all axes are combined to provide an "any-moving" signal. Determination of moving status, on the D-404, is better done using the Read Moving Status (^) command.

For applications requiring this type of signal, a mode using the small "l" command is available to allow port 4 to become a hardware "moving" output.

Technical Support

For Technical Support contact EAD Motors directly.

EAD Motors
1 Progress Drive
Dover, NH 03820

Telephone – (603) 742-3330
Fax – (603) 742-9080

Email: info@eadmotors.com
www.eadmotors.com

Warranties

Your D-404 is backed by a 1-year manufacturers warranty. This warranty covers against defects and hardware failures.

Exclusions:

EAD may refuse to service hardware that has been reworked or damaged beyond what is considered normal usage.

A fee will be assessed for repairs to the D-404 that fall outside of the 1-year manufacturers warranty window.

Contact EADmotors for all repair information regarding the D-404 at the numbers listed in the Technical support section of this document.