DCB-261
USERS GUIDE

Revision date: 03/02/2010

Table of Contents

1) Introduction

Congratulations!

.....on your purchase of a DCB-261 Stepper Motor Driver-Controller Board. The DCB-261 will provide years of reliable, accurate and cost-effective motion control. As with all AMS products, the DCB-261 is backed by over three decades of manufacturing excellence and a commitment to quality and support that guarantees your satisfaction.

This Technical Reference Guide will assist you in optimizing the performance of your DCB-261. Its purpose is to provide access to information that will facilitate a reliable and trouble-free installation. This User Guide is organized into the following sections: Hardware, Communication, Memory / Parameters, Commands, and Addendum. We recommend that each section be reviewed prior to installation.

Although the DCB-261 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 at (603) 882 1447 to discuss your application.

Thank You!
Your AMS Team

Product Overview

The **DCB-261** combines efficient bi-polar chopper ***Driver*** circuitry with AMS' Award Winning SMC-26 ***Micro-controller*** on a single, heat-sink mounted board, to operate small stepping motors. They are designed for low cost O.E.M. applications; yet include many enhanced operating features found in products costing much more:

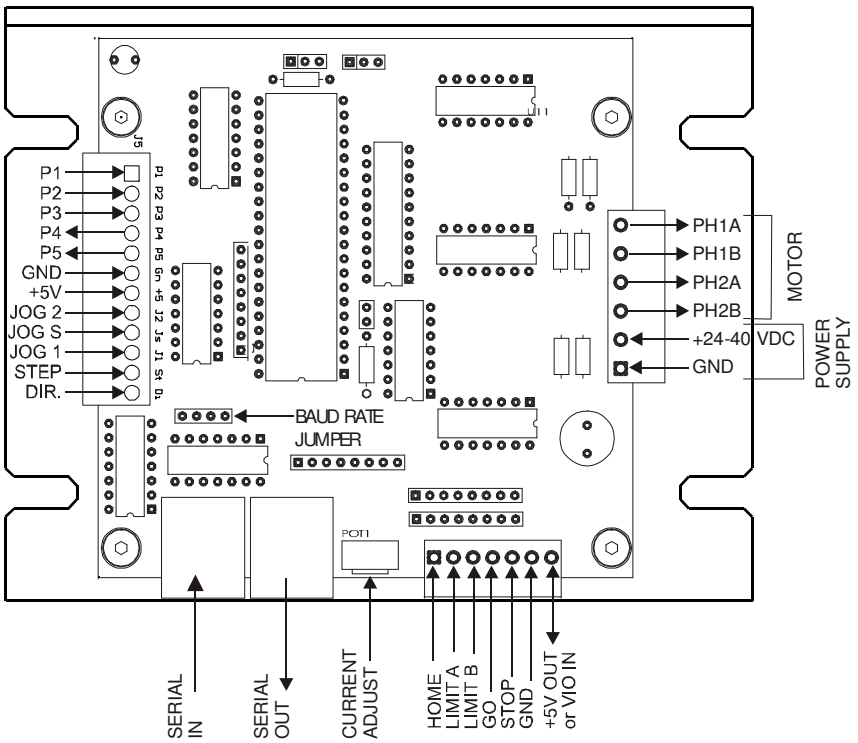
- 1.2 amp/phase chopper drive output
- Powerful SMC-26 intelligent controller
- Single 24 to 40 volt power supply input
- Full, 1/2, 1/4, 1/8 microstep to 20k SPS
- 2k bytes of non-volatile memory
- Limit, Home, Go and Stop inputs
- Step, Direction and Jog inputs
- Serial communication (1-32 axes)
- Adjustable run current pot
- Programmable hold current setback
- Programmable acceleration and deceleration ramp
- Constant velocity commands
- Heat-sink mounted
- Mating connectors included
- Free demo software

2) Hardware

Required Hardware for Operation

Qty	Unit	Model #	Description
1	Axis	DCB-261	Driver-Controller Board
1	System	User defined	+24 to 40Vdc power supply
1	Axis	SIN-9	RS-232 serial adapter (single axis- 9 pin)
		or	
1	System	SIN-11	Intelligent serial adapters
		and	
1	Added axis	BLC-51-3	Interconnect cable, Cat5 (3 ft.)
1	System	TERM-2	Terminator plug (included with SIN-11)

Assembly Drawing



Serial Interface (J1, J2)

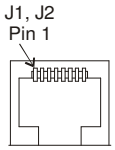
Two (RJ-45) connectors provide a "T" connection, facilitating multiple axis systems. This unique "mini-drop" network allows for a single ASCII character "name" to be assigned and stored in the integral non-volatile memory during the setup procedure.

J1			J2		
Pin	Signal	Comment	Pin	Signal	Comment
1	J2-1	Not used	1	J1-1	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

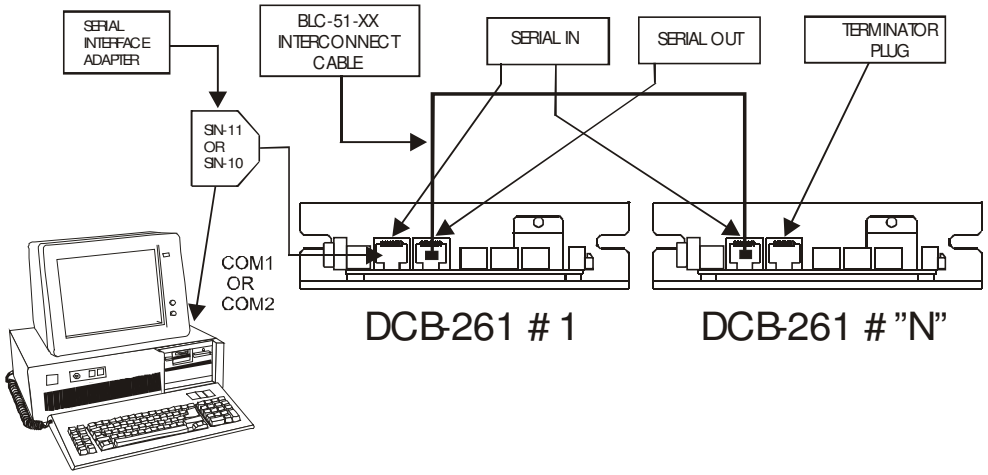
All signals on J1 and J2 are interconnected, except for the 5-volt power, which is not supplied to J-2 – 7.

The serial communications is full duplex at 9600 baud. The proper handshake MUST be implemented in the host computer to avoid loss of characters. This protocol is "echoed" characters. If your operating system software is not capable of character-by-character transmission, AMS offers a serial adapter called the SIN-11. The SIN-11 features a dual Uart microprocessor, RS-232 to RS-422 conversions and a character buffer. The necessary handshake is built in, thus the sometimes expensive and time-consuming software interface is avoided.

A complete description of the serial interface specifications and operation is contained in Section 2, "Communication Interface."



Multi-axis Serial Interface Connection



I.O. Connections (J3, J5)

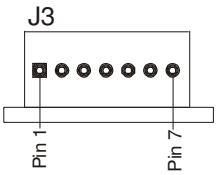
DANGER: Revision levels "0F" and earlier utilize a VIO jumper (W1) that MUST be removed before any external VIO supply is connected. Failure to do so may cause extensive damage to the unit.

Two connectors (J3 and J5) provide user inputs and outputs. All inputs are robust and can withstand voltages in excess of 28 volts. VIO is used to operate the input comparator circuitry**.** VIO is connected to the internal 5 volt supply. The user may supply an external voltage source of up to 28 volts DC. This is useful for interface to PLC's or unusual sensors. The input signal threshold will be approximately ½ of VIO. The inputs incorporate pull-up resistors (10k nominal) to the VIO signal.

Outputs are 5 volt sinking "logic" signals with limited (several mA) current ratings. For real world applications they may drive industrial relay connections with isolation. Opto-22 power series relays or equivalent TTL compatible (3 to 32 volt DC control) are recommended to switch high power loads. Typical choices include DC60S03/05 DC and 120D10-120VAC @10 amps (stocked by Allied Electronics). The same company supplies a line of digital I/O models (G4) for more elaborate optical input and output applications.

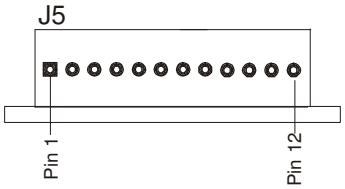
(J3)
A seven contact connector provides inputs for the most commonly used signals. This connector is mapped as follows:

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	Power common
7	VIO	+5V out or VIO input



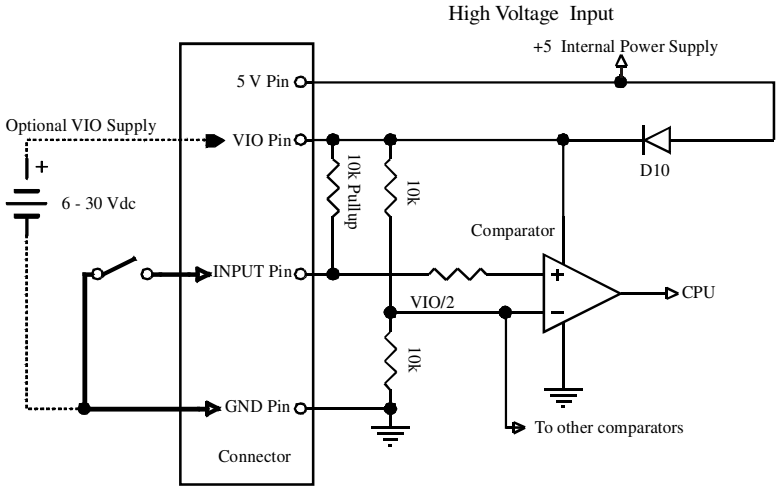
(J5)
In addition to Jog, Step and Direction, three input ports are available that can test and branch to multiple motion sub-routines. Two programmable outputs are also available to drive solid-state relays and other devices.

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



Note: Outputs are 5-volt logic with 10k pull up to 5 volts.

Input Ports (Step, Direction, Ports 1, 2, 3, Jog (3), Limits, Home, Go and Soft Stop)

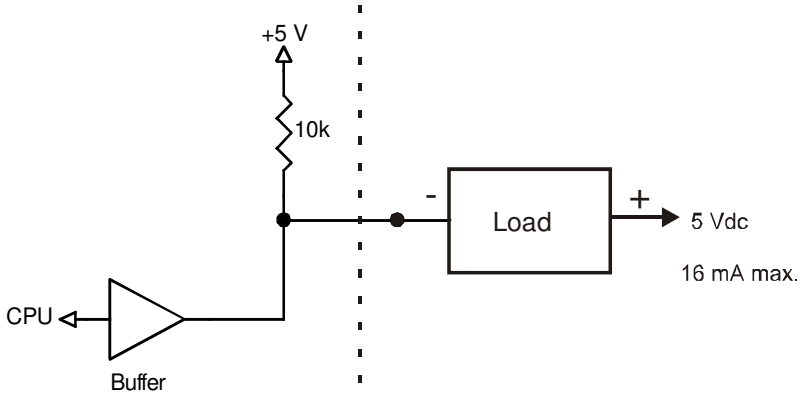


Typical user input circuit

A higher VIO (i.e., 24 volts from a PLC with 24 volt drivers) would increase the logic threshold to 12 volts, providing better noise immunity.

Output Ports

Two user output ports are provided on the DCB-261 utilizing a 5 volt output circuit capable of sinking a maximum 16 mA, DC.



Typical output circuit

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

Power Supply and Motor Connection (J4)

Connector J4 provides the power supply input and motor phase drive outputs. The recommended power supply is an unregulated DC design with voltage and current ratings, appropriate for the driver. The on-board 5-volt is for logic power.

For maximum motor speed performance the motor should have a low voltage (higher current, low inductance) and the power supply voltage as high as possible NEVER exceeding the DCB input ratings.

Pin	Signal	Type
1	Gnd	Ground
2	VMM	+24 to 40Vdc
3	PH-3	Phase 2-B
4	PH-2	Phase 2-A
5	PH-1	Phase 1-B
6	PH-0	Phase 1-A

Typical Wiring Diagrams for Step Motors (See "About Step Motor Current" in the Addendum for more information)

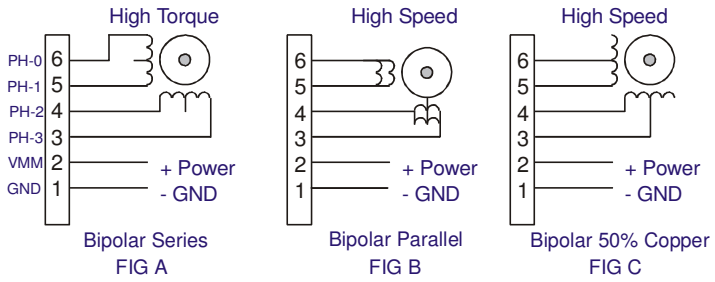
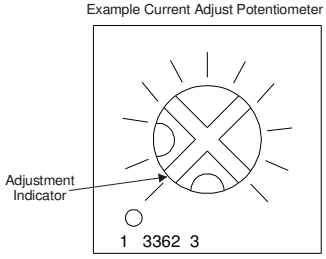


Fig. A: Series winding for higher torque and lower current. The inductance is 4 times that of the parallel mode, reducing the maximum obtainable speed.
Fig. B: Parallel winding for better high-speed performance but requires higher drive current. A 4-wire motor is the same as an 8-wire motor, but it is 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 two minutes after turning off the 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.

Adjusting the Motor Current
The current adjust potentiometer has ten equidistant marks that can be used to set the output current to the motor. The first mark, in the full CCW position, represents zero output current. Each incremental adjustment in the CW direction, adjusts the current approximately 1/10 of the maximum output current rating of the product.



The following table illustrates (approximate) current settings:

Adjustment Indicator	Current Setting (Amps)
Full CCW	0
1	0.1
2	0.2
3	0.3
4	0.4
5	0.5
6	0.6
7	0.7
8	0.8
9	0.9
Full CW	1.0

The run current may be further refined by use of the empirical method for fine-tuning (see Addendum; "About Step Motor Current").

"Do's, Don'ts and Important Notes"

NEVER connect or disconnect motor wires while power is supplied.

When using a 6 lead motor be sure to insulate/isolate unused wires.

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.

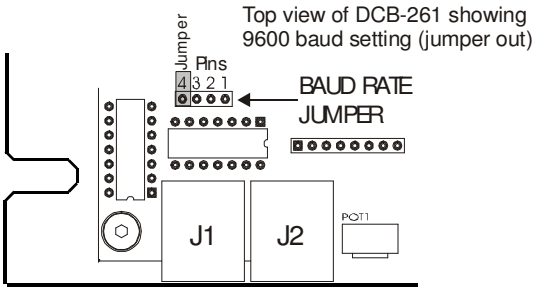
********** WARNING **********

LIVE CONNECTING/DISCONNECTING MOTORS WILL CAUSE DAMAGE THAT IS NOT COVERED BY WARRANTY.

Baud Rate Jumper (W2)

Baud rates for the DCB-261 can be configured as follows:

W2 Pin Connections	Baud Rate	Comment
Out	9600	Factory Setting
1-2	460k	Not used with SIN-11
2-3	38.6k	Not used with SIN-11



Specifications

Electrical

Output Current DCB-261 (max)..…… 1.0 Amps
Chopping Frequency.....……...20kHz
Input Voltage.....…… +24 to 40Vdc
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, 38.6k
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

*Threshold

J3 Input Signals: Limits, Home, Go and Soft Stop
J5 Input Signals: Ports 1, 2 and 3, Jog 1, Jog 2, Jog Speed, Step, Direction

Signals	Min	Typ	Max	Units
VIO Supply (J3-7)	5		28	Vdc
Threshold	2.5*	½ VIO	36	Vdc
Input Voltage	-0.3		36	Vdc
Input Current	0.5*	VIO/10	2.8	mA

*VIO= 4.7 volts using internal supply

J5 Output Signals: Ports 4 and 5.(open collector with 10k pull-up's to 5 volts)

I.O. Signals	Min	Typ	Max	Units
Outputs Ports 4 and 5	0.7		5	Vdc
Output Current (sink)	6		16	mA

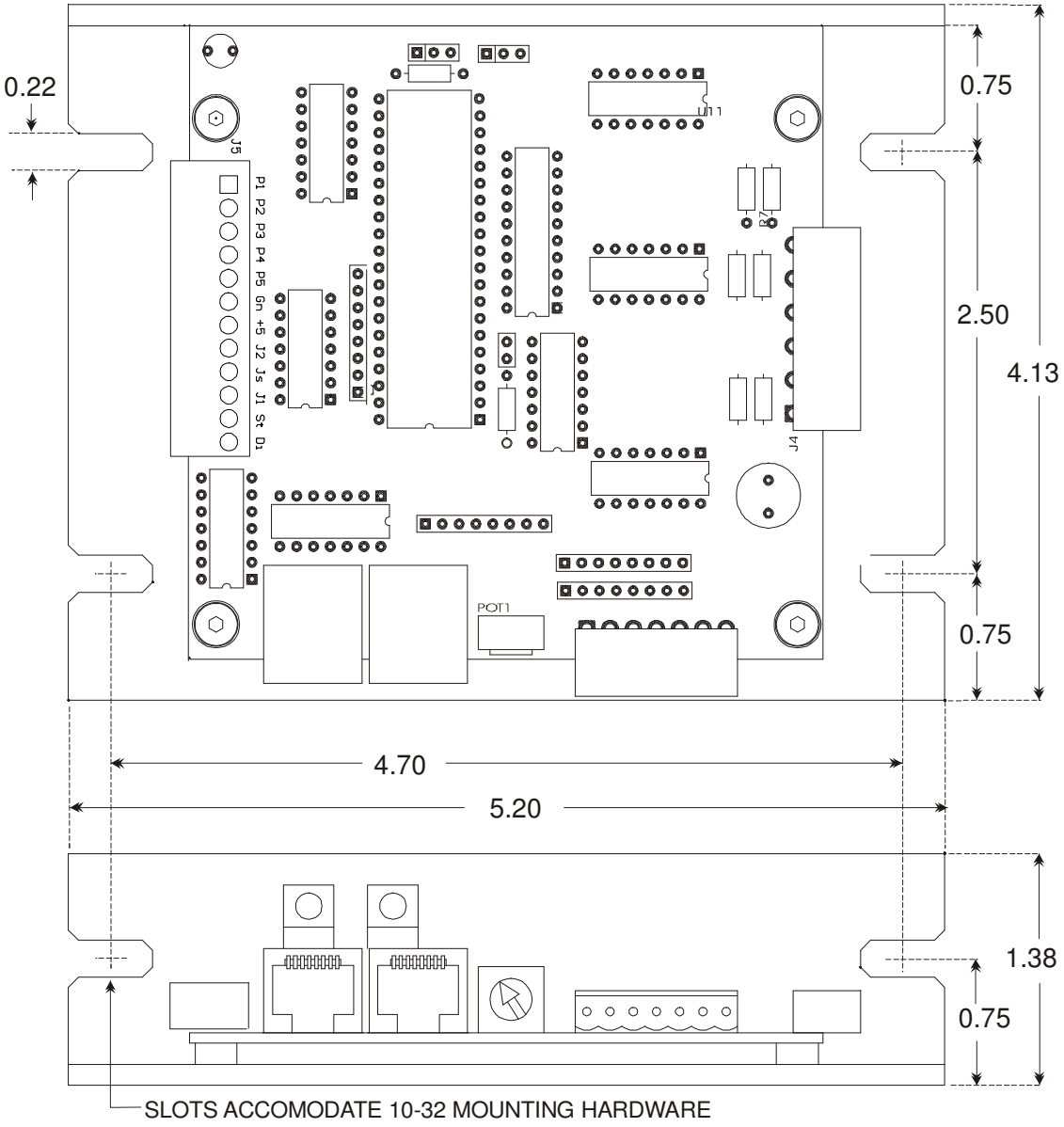
Environmental

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.



3) Getting Started

Required Hardware for Operation

Qty	Unit	Model #	Description
1	Axis	DCB-261	Driver-Controller Board
1	System	User defined	+24 to 40Vdc power supply
1	Axis	SIN-9	RS-232 serial adapter (single axis)
Or			
1	System	SIN-11	Intelligent serial adapter
And			
1	Added axis	BLC-51-3	Interconnect cable, Cat5 (3 ft.)
1	System	TERM-2	Terminator plug (included with SIN-11)

Basic Hardware Set-Up

Prepare for operation by referring to the illustration in the preceding section and following these simple instructions:

1. Attach the 9 Pin connector end of the SIN-11 (multi-axis application) or SIN-9 (single axis application) cable assembly to a COM port of your computer.

2. Connect the other end of the cable assembly (looks like a LAN connector), to the mating connector "J1-Serial Input" of the DCB-261.

3. (Multi-axis only) Install a terminator plug (TERM-2) into J2 "Party Line Serial Output" of the last axis.

4. Connect a motor and appropriate power supply to connector J4. The motor current is adjustable by the potentiometer labeled "current" and is factory set to 0.8 A/phase. A clockwise turn will adjust the current up. Likewise a counter clockwise turn will reduce the current. The motor current should be set based on the motor rating, speeds, load, etc. Refer to the Addendum "About Step Motor Current" for more information.

Note: NEVER connect or disconnect the motor when the power is "ON".

Software Set-Up

If you are using a

a. SIN-9 adapter, please download the "AMS Cockpit" software from the AMS web site (www.stepcontrol.com)
b. SIN-11 adapter, you may use the "AMS Cockpit" software or any other program that enables communication through a serial port, such as "HyperTerminal" which is provided as part of the Windows Operating system up to Windows XP.

Please see the Section Communication Interface in this manual for more information.

5. Once you have selected and - if needed - installed the communication software, make sure that the active COM port matches the COM port to which the SIN-9/SIN-11 is connected.

6. Ensure that the port parameters are selected as follows: bits/s: 9600, data bits 8, parity: none, stop bits: 1, flow control: hardware.

Sign-On

7. Ensure the DCB-261 is powered up. For the next steps, if using the AMS Cockpit software, you need to make sure that you are in the Dumb Terminal mode and not in the Party Line Mode. Dumb Terminal mode is the default upon start-up. See the Section Communication Modes for more details.

8. Strike the SPACE BAR key. The controller should sign on with the software version number (Vx.xx). If not, enter a (^C) (reset) and strike the SPACE BAR key again.

Striking the ENTER <CR> key should result in an echo of "# " characters, further indicating communication is established.

9. At this point it is necessary to set the Hold and Run current. The DCB-261 is shipped with a 0% hold current setting. The Hold current is based on a percentage of the Run current value that is set manually via the current adjust potentiometer (Current).

Set the defaults:

A. Enter the "E 2" command (sets hold current to 50% of run current). See the description of the "E" command in this manual for more details.
B. Enter the "S" command (save default to NV memory).

Adjust Potentiometer labeled "Current", using an amp meter, to set the Run current as follows:

C. Adjust "Current" fully CCW to minimum current,
D. Insert an amp meter in series with one phase,
E. Apply power,
F. Issue an "E 3" command. This will disable the hold current reduction,
G. Slowly increase the potentiometer setting until the current is at the desired value.

Note: Refer to the Addendum; "About Step Motor Current" for more information.

Axis Naming Procedure

10. If you are intending to connect more than one axis to one COM port (multi-axis operation), each axis must be assigned a unique "name" for proper operation.

A. Ensure only one axis is connected and reset the controller (^C), then type a single, valid (upper or lower case) name character. Recommended axis names are from upper case A through Z and lower case a through z. Non-valid axis names are shown in the chart below.

Non-valid axis names			
ASCII	HEX	ASCII	HEX
[	5B	^C	03
\	5C	CR	0D
]	5D	LF	0A
^	5E	@	40
-	5F		
‘	60		

B. Follow the name with a SPACE BAR. The sign-on message will appear.
C. Verify the Name by entering the "X" command <CR>. The last item of the first line should contain the "Name" character.
D. Enter the Save command (S) <CR>. The axis Name is now stored in non-volatile memory and will be maintained until changed by the user in the future.

The DCB-261 is ready to operate in the present single axis mode (dumb terminal mode) or be switched over to Party Line mode. It is suggested that the operator use single mode first to become familiar with command input. The single axis mode can be used with any "dumb" terminal device and is not dependant on using the AMS software. For multi-axis operation refer to "Party Line Mode" in the Communication Interface section of this manual.

Note, single axis mode (Dumb Terminal mode) operation does not require the name axis procedure.

11. Examine Command

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

Default Parameter	Value
K	5
I	400
V	5016

Where:
K= Ramp Slope
I= Initial Velocity
V= Slew Velocity

The values shown assume there are no input connections or special modes such as inverted limit switches.

12. Simple Command Execution Example

Enter the following commands:
A. +1000" <cr> (the motor should move 1000 steps),
B. "Z" <cr>. The position 1000 should be displayed.

Some Basic Rules:

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

Programs

13. The above examples were samples of immediate commands. The following is a sample of the sequences that are stored in non-volatile memory.

Note: when programming, the sequence is immediately written to non-volatile memory without any additional action required to save it.

	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	R-10000<CR>	Move 10,000 steps in the "-" direction, relative to Origin.
14	W0<CR>	Wait until complete.
17	J1 3<CR>	Jump to address 1, 4 times.
21	R500<CR>	Move 500 steps in the "+" direction, relative to Origin.
26	P0<CR>	End Program.
Now list the stored program		
	Q<CR>	Query command.

14. Verify the Program

The DCB-261 will respond with:

0	O	
1	R	10000
6	W	0
9	R	-10000
14	W	0
17	J	1 3
21	R	500
26		

15. Execute the Program

Enter	Remark
G0 1 <CR>	Programs start executing at location zero. If the Trace option is on, it will display each instruction, prior to execution.

Note: The program can be terminated at any time by hitting the ESCape key.

16. Edit Program

Example: It is desired to change instruction number 21 from 500 steps to 5,000 steps:

Enter	Remark
P21<CR>	Edit instruction 21.
R5000	Move 5,000 steps in the "+" direction, relative to Origin.
"ESCape"	Terminates Edit mode.

Note: When editing commands in dumb terminal mode, variations in Command byte length may affect subsequent command address locations and possibly cause corruption of the non-volatile memory storage.

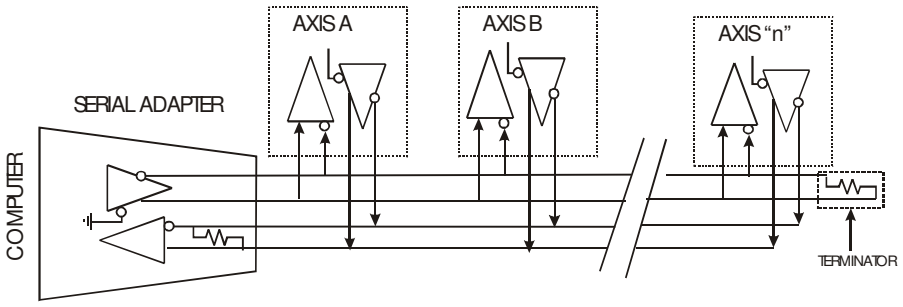
4) Communication Interface

RS-422 Hardware

AMS communication protocol 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; receive data is one pair of wires and transmitted data a second pair
- 0V to 5V differential signals for high speed and robust noise rejection over long distances
- Data speeds to 100K baud
- Up to 32 controllers from one COM port
- Cable network length to 1200 meters (4000 ft)
- Use for single controller "dumb terminal" mode

RS-422 Connect



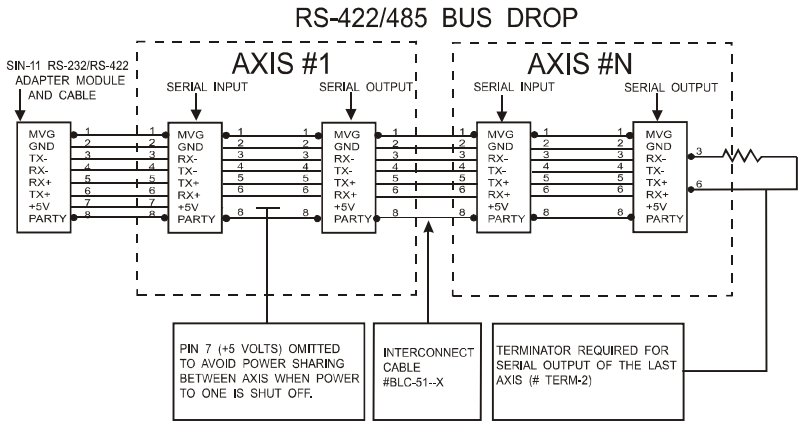
Communication hardware requires three components:

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

Other Party Line Signals

In addition to the 4 serial data bus wires, several other signals exist in the AMS party line interconnect.

1. **GND** (pin 2) is common for all devices (controller). 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 from the first controller.
3. **Party Select** (pin 8) is used for other products that require this input.



Note: Pin 8 Party is not used in products utilizing the ^N and ^P commands.

Adapters

AMS offers adapters suitable for a variety of applications, as follows:

SIN-9 Passive Adapter
The SIN-9 adapts RJ45 to DB-9. It is wired directly through with RS-232 levels passing to the appropriate RJ-45 pins. These will only interface to one controller. Application software must implement special character-by-character handshake protocol. This model supports only operation in Dumb communications mode and cannot be used in Party Line mode (These modes are discussed below). Also, it is not suitable for USB interface.

SIN-9, Serial Adapter

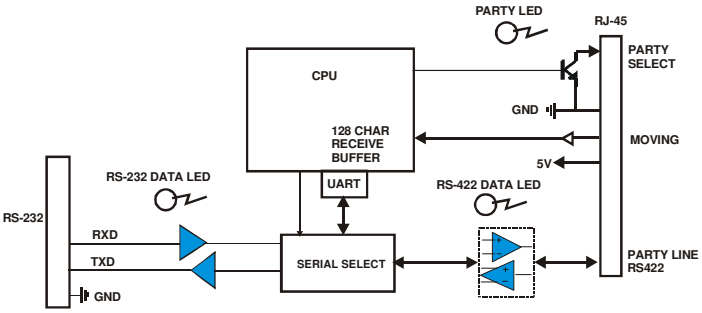
SIN-11 Intelligent Serial Adapter (Recommended)
The SIN-11 is an intelligent serial line converter that simplifies application software development and improves overall performance. Communication between connected controllers and the SIN-11 is 9600.

Specific operating instructions are contained in the SIN-11 Users Guide.

The SIN-11 has a built-in microprocessor that offers a number of features:

- Operates as hardware RS-232 to RS-422 adapter
- Diagnostic LED's
- 9600 baud rate
- DB-9 serial input connector
- RJ-45 party line connector
- 5 volt powered from controller
- 128 character buffers for multiple commands per line

Because the SIN-11 eliminates the need for special 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.



SIN-11, Intelligent Serial Line Converter

There are several commands that the SIN-11 can execute including: “Scan for controller present” (required initialization) and “Wait until motion is complete” (one or all controllers).

On power up the SIN-11 all start in the Single Controller mode where characters pass directly between the RS-232 and RS-422 bus. However, the SIN-11 monitors the ASCII stream for the presence of the special “&” character (several other trigger characters are also available).

When the “&” is detected, the CPU awakens and performs several actions:

1. Isolates input (RS232) from output (RS422).
2. Asserts the party select signal (pin 8) to the “on” condition –used by many.
3. Emits a software reset (^C) to the controllers.
4. Emits a ^P (control P) to the controllers which places the DCB-261 in party line mode.
5. Scans and maps party line controller into memory.
6. Reports the named controller as found.

The SIN-11 is now configured as a “line input” device, that is, the host computer can print a complete text line containing multiple commands. Once the line is received, it is processed starting with the first character received.

Assuming that there are two controllers named “A” and “B.” A typical command string to a system could be:

A+1000;B+1000;&W*;AZ;BZ

This would cause both axes to move the specified number of steps; wait until motion is stopped, then read back the two positions.

Communication Modes

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

1. “Dumb” Communications Mode

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: SIN-9 or SIN-11
At start-up:

1. Hit the SPACE BAR key to sign on.

In Dumb communications mode, you can do a number of useful things:

- Assign “name” character (not necessary if using daisy chain). The dumb terminal mode must be used for name assignment – it cannot be done in Party Line mode.
- Tweaking speed and acceleration parameters
- Experimenting with commands
- Development of program sequences
- Storing motion sequences for non-hosted applications

Note: Single axis mode should never be used in a computer or PLC hosted applications. If the design has a single axis then the daisy chain method can be used with either RS-232 or RS422. Single axis functions are suited for programming using the keyboard with visual screen “feedback.”

2. Party Line Mode
Party line mode is intended for computer-controlled designs. A computer (usually a PC) can address one or more axis 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. The SIN-11 adapter simplifies this protocol.

Serial adapter used: SIN-11
At start-up:
1. Issue an "&" command to enter Party Line mode.
2. The host computer interrogates and records axis name(s).

Note that in Party Line mode every command issued needs to be preceded by the name of the axis which is intended to process the command. For example, if the user intends to issue the command M1000 to axis A, the following command line needs to be issued: "AM1000".

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

Serial adapter used: SIN-9
At start-up:
1. The host computer emits axis #1 name, receives ending axis name +1.

Party Line and Daisy Chain Line Commands

Note: If a SIN-11 is used, then the following rules will not apply because these adapters will perform the necessary handshake.

The SMC series controllers incorporate a buffered UART input, capable of receiving and holding ONE character at a time. The controller **must** read this character before another one is received; otherwise the UART will be over-run, resulting in missed character errors.

The handshake method used is a simple "echo" of the received character. The host computer MUST ALWAYS wait for the echo.

Fixes such as insertion of delays between characters may seem to work but will ultimately fail. Beware that many PLC manufacturers do not provide the serial software flexibility required for your application to make the proper communication.

(The SIN-11 adapter provides handshaking functions and other features to make life simpler and reduce software development time).

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 name must be preceded by an LF (presumably the terminator for the previous command), i.e., <LF>"n" xxxxxxx <LF>.

Note: An LF can be generated using a 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. If the axis does NOT detect a proper name and command, then the data is simply echoed back to the terminal. The designated controller re-issues the Line Feed after processing the command.

If the command is of the type that results in a data output (such as "Z"), then 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. The terminal can interrogate the motion status using the appropriate command to determine if a function is complete. Editing features are NOT supported in daisy chain or party line operation.

Note: the commands "Control C" and "ESCape" do NOT require the use of, and will NOT be qualified by, a "name" prefix. All devices will respond.

The party line sequence can be sent using the dumb terminal. Caution must be used because any typo's cannot be corrected with a backspace, as is possible in the single axis mode. You must cancel with the ESCape and start over. Remember ESC is a global abort character.

Party Line Startup

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 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. ESCape aborts 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 outputs 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 input and hard stop (ESCape).

External indication of PENDING PERIOD end, execution and result cycle of the pending instruction is the carriage RETURN or Line Feed in the party line mode. The GO command is regarded as a command that has a continuous pending (Instructions Queued) period.

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 the ESCape character.

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.

If more then one process is active then ALL are aborted.
Abort is Global – all axis halt.

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.

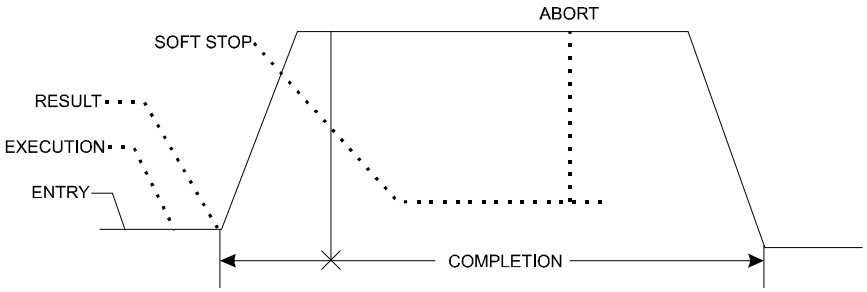
Process	Resulting Action
Pending period	Decelerate and cancel pending instruction.
Program execute	Decelerate then terminate execution.

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.

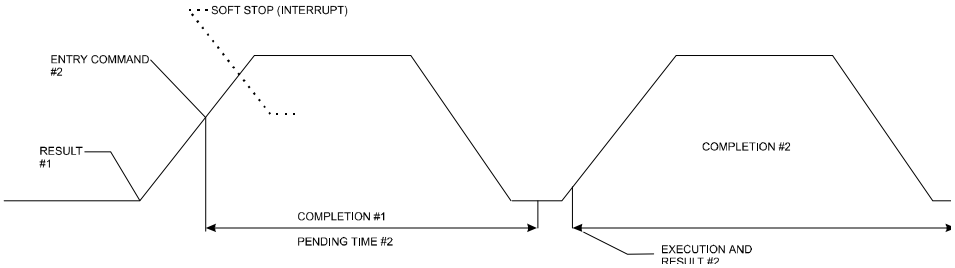
Jog Speeds, Homing
Jog input and home speed is a special case of the constant velocity command. Inter-speed ramping is used if the programmed jog speeds are above the initial velocity. Homing does NOT employ a deceleration ramp on reaching the home sensor.

Note: In any mode, 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.

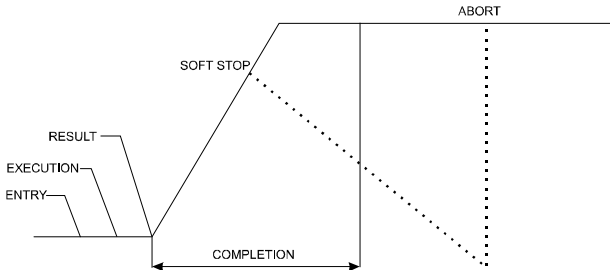
Command Cycle Examples
Index Cycle Resulting From +, -, R Commands



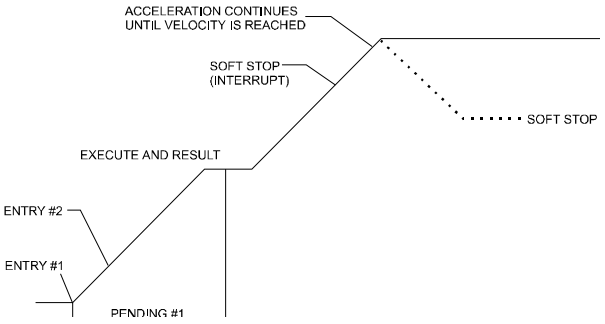
Queued Index Cycle Resulting From +, -, R Commands



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. One receipt of the 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

Communications Software

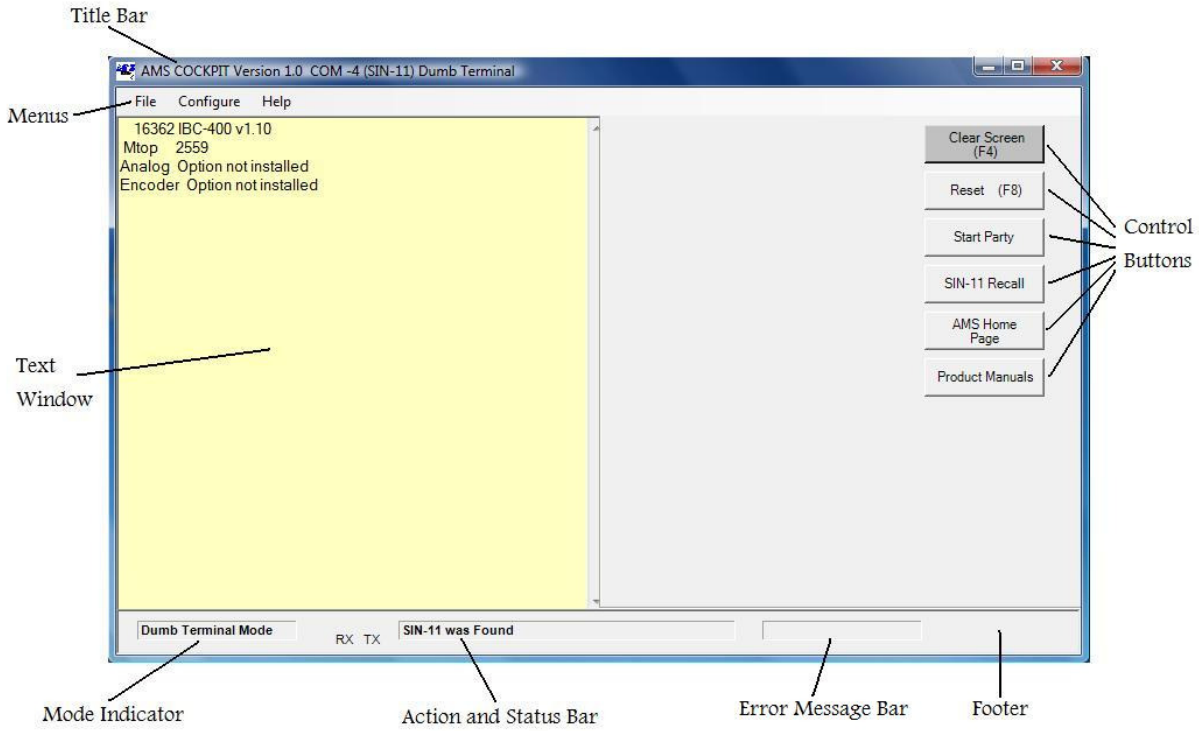
AMS offers the "AMS Cockpit" software that can be downloaded for free from the AMS website to assist customers in the implementation of their projects. It is compatible with Windows Operating systems. In addition to enabling communication, it includes some customized functionality, such as the downloading of programs into the non-volatile memory of the controller. This code is not intended to operate as an end user application program, but rather to allow familiarization, evaluation and programming of the AMS products.

When using the intelligent serial adapter SIN-11, it is possible to use virtually any program that enables transmitting and receiving of data via the serial port. An example is "HyperTerminal" by Microsoft which is delivered with Windows Operating systems up to Windows XP. Unfortunately, it is no longer included in Vista and Windows 7.

The default baud rate for all AMS products is factory set to 9600 baud. When using a third party software to communicate with the controller, it needs to be ensured that the port settings are as follows: data rate: 9600b/s, data bits: 8, stop bits: 1, parity: none, flow control: hardware. This represents the default for most PC's.

See the chapter entitled Getting Started for initiating the communication between computer and the AMS controller.

Also, there is an extensive manual for AMS Cockpit available from the AMS website.



Main Screen of AMS Cockpit Software

5) Memory / Parameters

Non-Volatile Memory Details

The SMC-26 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. Each time a cell is written a small number of electrons are trapped in the dielectric. After many write cycles the dielectric becomes less effective and the cell cannot retain its charge. The write life cycle endurance rating is constantly being improved. 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-26 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. The SMC-26 parameters are set so quickly, even in SERIAL mode, that you should let the host download them.

Changing parameters should NOT be done by writing directly to EEPROM. The SMC-26 won't recognize that it was changed and may over-write them. Use the commands available to set parameters. Reading on the other hand is non-taxing on the EEPROM. The DIVIDE factor is readable at 229 (0E5 hex). Trying to read and write Initial and Slew velocities from the EEPROM will be confusing as they are stored as timer reload values. Use the EXAMINE command ("X") in SERIAL mode.

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[1]	Trip routine
192-226	Unused
227	Configuration byte
228	Internal initial status byte (Do NOT modify)
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[1]	Branch area power up commands
1600-2047[2]	User program power up commands

[1]Committed only when specific command is being used, otherwise used as general-purpose storage. Locations 247 thru 255 are protected from the “Clear” command. Most of the data contained in these locations is in binary and should not be tinkered with.
[2]If a valid command exists at location 1600 through 2047 it will be executed on power up.

Default Parameter Table

The following default values are written to NV memory after the 'Clear'(C 1) command:

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)	3/20 (90-600)
Trip Point (T)	Off
Mode (H)	1/4 micro step
Auto Power Down (E)	Chop off (zero current)
Limit Polarity (H)	Low assert
Auto Position Readout (Z)	Off
Name (after reset)	Unchanged
User Programs (0-191)	Cleared

Turbo Ram

The SMC-26X2 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 Ms. 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

6) Commands

Command Format Description

Command	Function		Type	NV Bytes
	Mnemonic	Data 1 (Range)	Data 2 (Range)	Result

Where:

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)
	Information returned as a result of command execution or examination

Note: a comma separates two parameters.

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.

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. It is analogous to "Ctrl-Alt-Delete" - reboot the computer. All outputs are set high, defaults are reloaded from NV memory, and position is set to zero.

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)

Input data ranging between 0 and 63 is complemented then output to port 1 through port 6. Port 1 is the least significant bit. Binary combinations of bits will turn on more then one port. Example "A 7" will set ports 1, 2 and 3 to an ON condition. At hardware reset all outputs are set off (high). The command "A 128" will cause ports 1 through 5 to increment in a binary fashion. The command "A 129" will read and display the port data.

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

Reading the port data provides the following result information:

Data	Cause
1	Low input present on port 1
2	Low input present on port 2
4	Low input present on port 3
8	Low input present on port 4
16	Low input present on port 5

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 8	Turn on port 4.
W 500	Wait 500 milliseconds.
A 0	Turn off all ports.
P0	Exit program mode.

Note: the actual ports usable for output is determined by the hardware design. AMS products generally define ports 4, 5, & 6 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 J5-9) is held low. The values are multiplied by 30 to determine the actual step rate in steps per second. Setting values of 0 will disable the jog. Speeds are divided by the "D" value. 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) v1.11 only

The DCB-261 has been redesigned 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:

Threshold	1/2 SPS	1/8 SPS
0	Fast	
25	650	2400
50	1300	4800
100	2400	9600
150	3700	14000
200	5000	19200
255	Slow	

Because the step rate is actually measured, the decay detection functions for external step pulse input. Slow decay provides smooth operation with reduced resonance's at slow to medium speeds. Fast decay will generally enhance high-speed operation at speeds above 200-300 RPM. The actual settings usually must 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 all remains 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 affected.

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 specified number (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.

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.

Command	Function	Type	NV Bytes
E	Enable Control	Default, Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) E (n)	0-32	None	None

E (Enable Control)

Where "n" specifies the amount of hold current as a fraction of the running current. Each step motor winding current is a function of the 3 bit sine and cosine values generated by the SMC-26X2. Current reduction to the motors is achieved by reducing the binary output values. Note that the reduction can only be approximate.

E-Value	Hold Current
0	0% windings off
1	25%
2	50%
3	100%
16	50% (temp)
32	100% (temp)

"Windings off "prevents generation of EMI/RFI noise.

Note, the armature position can shift slightly while in a reduced current mode. The 100% value is determined by the external control and driver circuit, usually with a potentiometer.

The "E" value is stored as a default in the NV memory. The calibrate (16, 32) values are temporary and are not stored in NV memory.

The special calibrate mode is designed to assist in current calibration. Both phase currents are set to the same values:

E 16 will set the average (RMS) value.
E 32 sets both phases to maximum (PEAK) value.

This is a temporary condition, over riding the phase value. When stepping resumes the phase currents assume their normal values. The "ESC" key will cancel this mode.

The peak value corresponds to high torque full step values (H3, H4) while the RMS can be used with microstep calibration.

Current set procedure

1. Insert an ammeter in series with phase 1A.
2. Rotate both SIN and COS pots fully CCW (minimum) or off.
3. Apply power to driver.
4. Enter the E32 command.
5. Slowly increase the COS pot until the desired current is obtained. Never exceed the maximum drive rating.
6. Increase the SIN adjustment until the LED just switches on or off.
7. Enter "ESC" to enter the holding current state.

Now the currents in both windings will be balanced. Note that the H32 setting for a prolonged time can cause excess current (hence heating) in some motors.

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-26X2 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

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 location "a." Most programs will start at "0", however, you may wish to start at another address. The address MUST begin at a stored instruction address, i.e., "go to" data produces unpredictable results.

If "t" is 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 controller is factory set with the following program example:

```
P 0        Enter program mode.
+ 1001     Move 1001 steps in the plus direction.
W0         Wait until complete.
- 1000     Move 1000 steps in the minus direction.
W0         Wait until complete.
Z 0        Display step position.
G 0 0      Go to location 0 and run stored program.
P          Exit program mode.
```

The address range is 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-26X2 also features a special case for the "Go" instruction.

SPECIAL CASE "Go"

If the address is specified as 2048 (above the last NV memory address), the SMC-26X2 will read the Go input ports, then, branch to an address based on the state of input ports 1 through 4. 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

The physical input ports are internally inverted as part of the address computation. State 1111 is defined as a high or +5v on port 1 through port 4. Commands located in address space between 129-191 will execute much faster.

Command	Function	Type	NV Bytes
H	Set Step Size Resolution	Default, Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) H (n)	0-5	None	None

H (Step Resolution)

This command selects step size resolution. The SMC-26X2 has an internal lookup table of up to 32 bytes corresponding to 1/8 step. 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

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
I	Set Initial Velocity	Default, Immediate, Program	3

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 (stalling).

As with all velocity parameters, the initial velocity is divided by the divide factor (D). Using the examine (X) command displays updated velocities. 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 255 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.

This command may be implemented within a program. Following, is an 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) K (n1, n2)	Accel (0-255)	Decel (0-255)	None

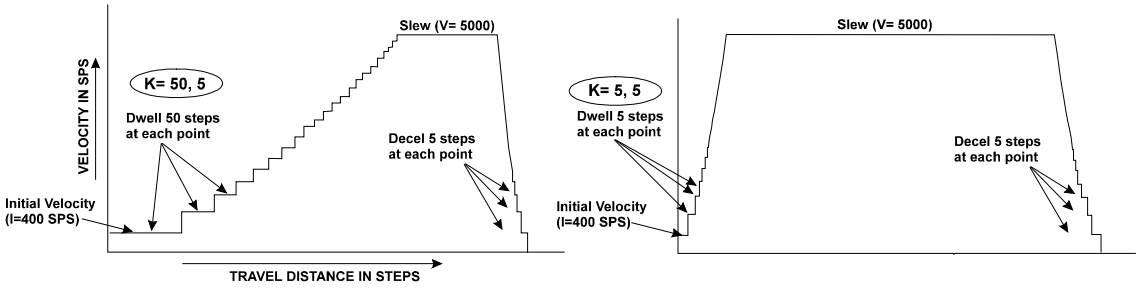
K (Ramp Slope)

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:



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 microstep 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.

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 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 best way to illustrate the power of these commands is with an example:

Enter as follows:

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.

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.

ONE IMPORTANT ADDITIONAL ACTION is performed. The position counter is reset to ZERO and causes the repeat of trips 200, 400 and 600.

Notes:

1. The physical motor travel will be 6000 steps, even though the position counter has been reset 10 times.
2. The cycle will repeat 10 times.
3. The position counter ends up at zero.

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 to a previous instruction then the program will loop until it becomes the specified level. The program will then continue to the next step.

Input ports are available as follows:

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

The SMC-26X2 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-26X2 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
l	Hardware Options	Default, Immediate, Program	2

Mnemonic	Data 1	Data 2	Result
(Name) l (a, d)	Options		None

l (lower case L; Invert Limit Polarity/Create Step and Direction Outputs)

This command permits option control and permits inverting the sense (polarity) of the limit switch inputs and can re-define two outputs as a step and direction output.

Limit Polarity
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.

Create Step & Direction Outputs
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.

The internal option flags are set as follows:

Flag	Data	Function
0	1	Invert input limits. Both inputs must be held low to allow a move.
1	2	Converts output of port 4 and 5 to step and direction signals.
2	4	Converts output of port 5 to a hardware moving status signal.
3	8	Gentle jog stop (jog-decel). Large overshoot if K is high value.

Ports 4 and 5 reassignment:

Flag	Limit Polarity	Port 4	Port 5
0	Low input	P4	P5
1	High input	P4	P5
2	Low input	Step	Dir
3	High input	Step	Dir
4	Low input	P4	Mvg
5	High input	P4	Mvg

The step and direction outputs, when used programmatically, can be a very powerful command. 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 as the DCB-261. An example is as follows:

P0	Enter program mode.
l 2	Change ports 4 and 5 to step and direction outputs.
+ 1000	Move the DCB-261 and the stand-alone driver 1000 steps in the plus direction.
P	Exit program mode.

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

Mnemonic	Data 1	Data 2	Result
(Name) M	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 direction is specified by the sign preceding the velocity. The SMC-26X2 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) interrupt (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. During any index 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	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 character, causing immediate return to the 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

The program mode may also be terminated with the ESCape character, causing immediate return to the 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.

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 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 (l)
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-26X2. 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" i.e., when executing a trip subroutine and a new trip is set as part of the routine, the 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. Trip service routines should not contain index, wait or time consuming instructions.

Disable

To turn off the trip function, use 0 (zero) as the address parameter. The trip is not currently usable in the encoder mode.

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.

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. The maximum speed will be limited by the motor capability and/or power driver circuitry.

The final output velocity is divided by the value of "D." 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 SMC26X2 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" 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.
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. 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.

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 500 milliseconds.
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; Pre-energize)

The "w" command is a "pre-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 "pre-energize" time:

1. Automatic current setback ("E" command) is in effect.
2. If the delay between consecutive motion commands exceeds 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 re-energized almost instantly with the first motor step. Thus the motor must rapidly change from a "relaxed" position to the next step.

This pre-energize 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%. The drawback is that 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.
```

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 different 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, b= 30s, ½, n=C

Where:

K= Ramp up/ramp down
I= Initial velocity
V= Slew velocity
D= Divide factor
b= Decay
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
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-26X2 has the option of continuous readout via the serial interface. The "Z 1" enables this operation. Any change in position causes the position data to be sent to the serial output. The readout is terminated by a carriage return only.

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.
W0	Wait until complete.
- 1000	Move 1000 steps in the minus direction.
W0	Wait until complete.
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) [	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. At addresses 128-191 the NV memory is always Read (not the RAM). The data contained at the specified location is output as a decimal value.

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-26X2:	Lb	La	Hm	P5	P4	P3	P2	P1

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-26X2 does not use them. For instance, if the jog speeds are set to zero, the three jog inputs may 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-26X2:	J3	J2	J1	*	*	EF	EE	BD

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).

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 only in the opposite direction.

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-26X2. A non-zero value indicates 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. 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).

Command	Function	Type	NV Bytes
|	Terminate Program	Immediate	2

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

| (Selective Termination)

This command (pipend: vertical dash key-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. The Terminate 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)
```

of extra steps is due to (decelerate) ramp plus a few steps of overhead. The overstep difference is repeatable.

7) Addendum

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	ENABLE CONTROL	MODE	0-32			2	⊙	⊙	⊙
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							⊙	
Z	DISPLAY POSITION	CONTINUE	0-1					⊙	

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

ASCII Character Code

Ctrl	Char	Dec	Hex	Code		Dec	Hex	Char		Dec	Hex	Char		Dec	Hex	Char
^@		00	00	NUL		32	20			64	40	@		96	60	`
^A	☺	01	01	SOH		33	21	!		65	41	A		97	61	a
^B	☻	02	02	STX		34	22	“		66	42	B		98	62	b
^C	♥	03	03	ETX		35	23	#		67	43	C		99	63	c
^D	♦	04	04	EOT		36	24	$		68	44	D		100	64	d
^E	♣	05	05	ENQ		37	25	%		69	45	E		101	65	e
^F	♠	06	06	ACK		38	26	&		70	46	F		102	66	f
^G	•	07	07	BEL		39	27	‘		71	47	G		103	67	g
^H	◘	08	08	BS		40	28	(		72	48	H		104	68	h
^I	○	09	09	HT		41	29	)		73	49	I		105	69	i
^J	◙	10	0A	LF		42	2A	*		74	4A	J		106	6A	j
^K	♂	11	0B	VT		43	2B	+		75	4B	K		107	6B	k
^L	♀	12	0C	FF		44	2C	,		76	4C	L		108	6C	l
^M	♪	13	0D	CR		45	2D	-		77	4D	M		109	6D	m
^N	♫	14	0E	SO		46	2E	.		78	4E	N		110	6E	n
^O	☼	15	0F	SI		47	2F	/		79	4F	O		111	6F	o
^P	►	16	10	DLE		48	30	0		80	50	P		112	70	p
^Q	◄	17	11	DC1		49	31	1		81	51	Q		113	71	q
^R	↕	18	12	DC2		50	32	2		82	52	R		114	72	r
^S	‼	19	13	DC3		51	33	3		83	53	S		115	73	s
^T	¶	20	14	EC4		52	34	4		84	54	T		116	74	t
^U	§	21	15	NAK		53	35	5		85	55	U		117	75	u
^V	▬	22	16	SYN		54	36	6		86	56	V		118	76	v
^W	↨	23	17	ETB		55	37	7		87	57	W		119	77	w
^X	↑	24	18	CAN		56	38	8		88	58	X		120	78	x
^Y	↓	25	19	EM		57	39	9		89	59	Y		121	79	y
^Z	→	26	1A	SUB		58	3A	:		90	5A	Z		122	7A	z
^[	←	27	1B	ESC		59	3B	;		91	5B	[		123	7B	{
^\	∟	28	1C	FS		60	3C	<		92	5C	\		124	7C	|
^]	↔	29	1D	GS		61	3D	=		93	5D	]		125	7D	}
^^	▲	30	1E	RS		62	3E	>		94	5E	^		126	7E	~
^_	▼	31	1F	US		63	3F	?		95	5F	_		127	7F	

About Step Motor Current

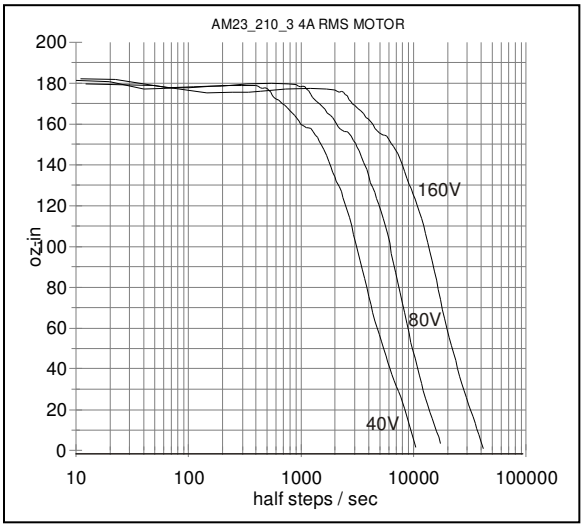
There is much confusion regarding the operation of step motors. Depending on your application, the step motor offers several advantages over servo motor designs, including lower cost and simplicity. The step (or stepper, or stepping) motor is a digital "synchronous" motor with a pre-designed number of "steps" per revolution. The most common motor has 200 full steps per revolution. Simple driver electronics can subdivide these steps into ½ step or more complex "microsteps."

Step Motor Characteristics

- The positional repeatability of each full or half step is very close to exact.
- While microsteps are repeatable, they tend to be somewhat non-linear.
- The torque is maximum at zero speed.
- The motor shaft RPM exactly correlates with the steps-per-second.
- Torque decreases with speed, eventually to zero or a "stall" condition.
- Resonance at certain speeds can cause undesired stalls or erratic operation.

There is little difference between today's step motor and the first generation of 60+ years ago. The magnetic materials and torque have been improved, yet it remains a simple, reliable workhorse of industrial motion control. Over time most improvements have been made to the drive and control electronics, i.e., microstep, solid-state components with higher voltage, current and switching speeds.

One insatiable hunger of a step motor is torque output at higher speeds. Winding inductance is the villain that limits speed. As the windings are switched on, the magnetic flux must be built up from current flow in the windings, producing mechanical torque. Higher step rates reduce the time available for flux to buildup and average current flow is reduced.



This reduced current results in reduced torque. The rate of current change depends on the voltage applied across it. High voltage applied across the coil will shorten the time constant.

Today's systems strive for low inductance motors and high voltage supplies. The above curves show the increased speed that might be obtained with higher supply voltages, up to 160Vdc. At standstill the average motor voltage is regulated to approximately 3Vdc.

A current sense circuit is used to switch off the current when it reaches the set value; hence the motor power is regulated. These "chopper" circuits operate at speeds above 20khz, well above hearing limits. The following is an abstract from "Control of Stepping Motors, a Tutorial" (linked from www.stepcontrol.com) by Douglas W. Jones, University of Iowa Department of Computer Science.

http://www.cs.uiowa.edu/~jones/step/index.html.

"Small stepping motors, such as those used for head positioning on floppy disk drives, are usually driven at a low DC voltage, and the current through the motor windings is usually limited by the internal resistance of the winding. High torque motors, on the other hand, are frequently built with very low resistance windings; when driven by any reasonable supply voltage, these motors typically require external current limiting circuitry.**"**

"There is good reason to run a stepping motor at a supply voltage above that needed to push the maximum rated current through the motor windings. Running a motor at **higher voltages** leads to a faster rise in the current through the windings when they are turned on, and this, in turn, leads to a **higher cutoff speed** for the motor and **higher torques** at speeds above the cutoff."

"Microstepping, where the control system positions the motor rotor between half steps, also requires external current limiting circuitry. For example, to position the rotor 1/4 of the way from one step to another, it might be necessary to run one motor winding at full current while the other is run at approximately 1/3 of that current."

Motor Choice
The discussion here relates to bipolar chopper motors. Internally, standard motors have 4 windings, resulting in a total of 8 wire leads. Motor manufacturers supply various configurations:

Leads	Application Connection	Comment
8	Bipolar (series or parallel), unipolar	All 8 leads are available. External interconnect can be cumbersome and untidy.
6	Unipolar or bipolar series	Can be used with 50% copper reduced torque but increased speed possible.
4	Bipolar series or bipolar parallel	Series: higher torque but reduced speed capability. Parallel: higher speed with lowered torque.
5	Unipolar only	Not suitable for bipolar drives. See AMS model CCB-25 with programmable phase sequencing.

Determining the Current Value
Question: What is the right current value?
Answer: The minimum value to operate reliably.

As the step motor current is reduced below the rated current, the torque output is reduced and eventually the motor will stall. The ideal current setting minimizes heating of motor and electronics, increases reliability, and reduces power supply requirements. Motors run more quietly and resonance effects can be reduced. One drawback from low current operation is that some microstep size linearity may be reduced, but full or half step accuracy is not adversely affected.

AMPS and Wire Count and Power

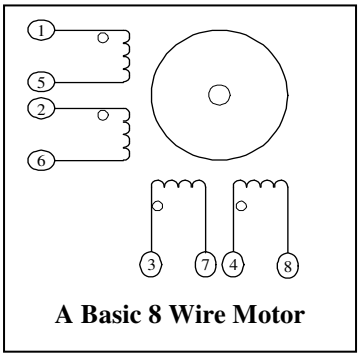
The rated current is specified based on the rated power input (watts) of a given motor.

A. Basic 8 Wire Motor

While never actually used as 8 individual coils, virtually all permanent magnet motors have 4 internal coils. All common configurations can be constructed from the 8 wire motor.

Let us assume that each winding of the 8 wire motor has the following specifications:

Current = 2 amps
Resistance =1.0 ohm
Voltage= 2.0 volts
Inductance = 4.4 mH



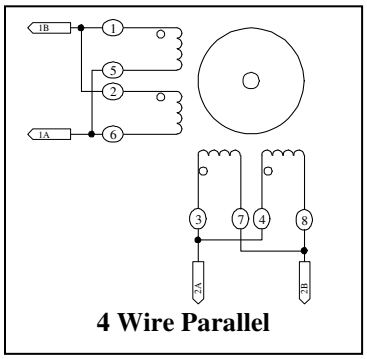
A Basic 8 Wire Motor

The power per winding is:

I^2R or 2 x 2 x 1= 4 watts,
x 4 coils = 16 watts total for this motor.

These values correspond closely with a NEMA size 23, 4 wire motor designs.

These following examples will configure the basic 8 wire motor into four real life connections:



4 Wire Parallel

4 Wire Parallel

The high-speed model implements parallel coil connection. Two coils connected in parallel result in the following for each of the two phases:

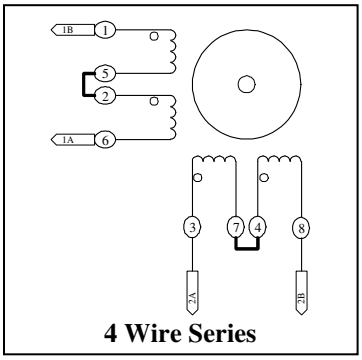
Parallel Resistance= 0.5 ohms
Parallel Inductance= 2.2 mH
Current= 4 amps (2 volts)
Watts per phase=8 (x 2 phases) = 16 watts total

B. 4 Wire Series

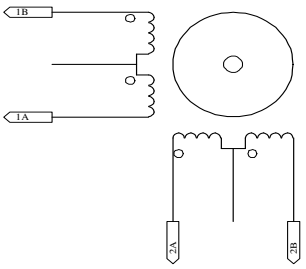
Changing to a series design, we have two pairs of two coils connected in series. Each has:

Series Resistance= 2 ohms
Inductance= 17.5 mH
The rated current is now 2 amps (4 Volts)
Watts per phase=8 (x 2 phases) = 16 watts total

Note that the series inductance is FOUR times the parallel design. Inductance limits the obtainable speed, since the time constant limits the amount of flux (hence torque) when step-to-step time is short.



4 Wire Series



6 Wire Motor

C: Adapting Available 6 Wire Motors

A 6 wire motor is equivalent to the 4 wire series motor.

Series Resistance= 2 ohms
Inductance= 17.5 mH
Rated current= 2 amps (4 volts)

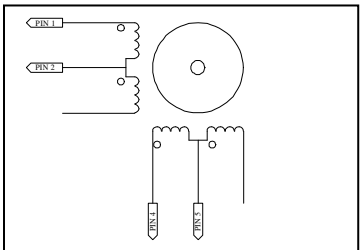
In practice the two coil ends are connected, while no connection is made to the center tap.

Half Copper or 50% Winding

The maximum speed can be increased by using ½ the coil. To do this, connect the driver between the center tap and one end of the winding.

The tradeoff is a loss of torque. The RMS current is the manufacturer's unipolar amperage rating with the same wattage per phase.

Often a 6 wire design is being upgraded or the size, features, availability or cost dictate the 6 wire motor. Some characteristics can make the motor impossible to use. Many motors are rated at voltages in excess of 5 volts. This means that 10 volts is necessary in the series (100% copper) configuration.



6 Wire, Half Copper

Aside from having excessive inductance, proper chopper operation dictates operation from voltage sources much higher than the motor rating. The minimum recommended value for VMM (DC supply) is 2 times the winding rating (the higher the better, until excessive heating occurs or insulation breakdown).

The RMS current rating for series operation is:

The manufacturer's unipolar amperage rating divided by 1.414. The lower current will reduce the average voltage slightly (about 7 volts).

Summary

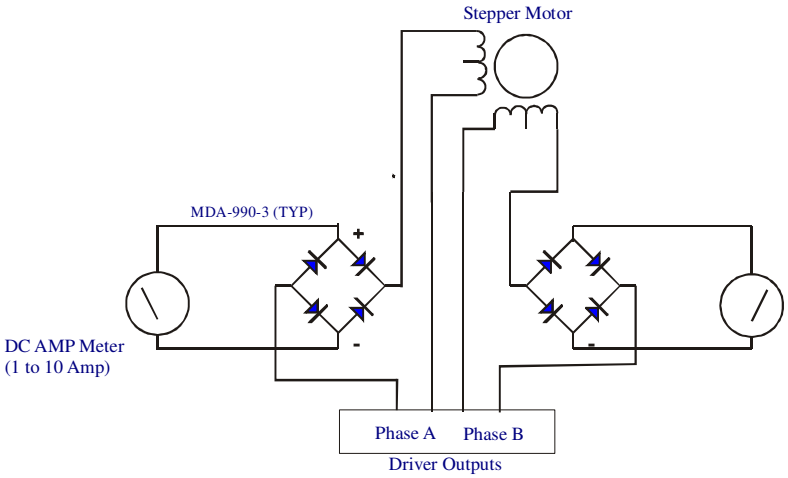
# Leads	Winding Connection	Per Phase Bipolar Current	
		Full Step (Total)	Microstep RMS (Peak)
4 wire	Parallel	4.0A (8.0A)	4.0A (5.66A)
4 wire	Series	2A (4.0A)	2.0A (2.8A)
6 wire	Unipolar	4A per phase	NA
6 wire	Series	2.8A (5.6A)	2.8A (4A)
6 wire	50% Copper	4A (8A)	4A (1.7A)

- Peak Current= One phase on and the other phase off.
- Peak Current =1.414 times RMS.
- RMS= Current per phase with both phases driven on (full step).
- RMS Microstep (or full step)= Both phases operating at equal currents.
- RMS = .707 times peak current.
- Total = Entire motor current.

Set-up for Current Calibration

The following is the basic setup and diagram for 2 phase current measurement:

A. The Amp meter can be digital or (preferably) analog.
B. The bridge rectifier must be rated above the maximum expected voltage and current.
C. A small capacitor (filter) may be needed across the meter.
D. A single meter circuit can be used, but two meters will indicate proper operation.
E. Additional meter protection circuitry may be desired (not shown).



Current Set-up Techniques

There are several basic methods used in establishing the initial motor current settings. The method used depends on the product model.

The following is a matrix of AMS products with adjustable current and the recommended (initial) current set-up techniques:

Model	Type	Adjustment	Method (See Below)
MAX-410/420	Microstep	Programmable	A1
CMAX-410/810	Microstep	Programmable	A1
SAX/DAX	Full/Half Step	Programmable	A2
CCB-26	Microstep	Single Potentiometer	B
DCB-241	Half Step	Single Potentiometer	B
DCB-261	Microstep	Single Potentiometer	B
DR-4M/PS	Microstep	DIP Switch	B
CDR-4/8MPS	Microstep	DIP Switch	B
DCB-264	Microstep	Dual Potentiometer	C
DCB-612	Microstep	Dual Potentiometer	C
ALL			D

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

LIVE CONNECTING/DISCONNECTING MOTORS WILL CAUSE DAMAGE THAT IS NOT COVERED BY WARRANTY.

General Procedure for all Methods

Assume a 2 amp bipolar motor (4 wire, parallel connection). The RMS value is 2 amps per phase, thus the peak (only one phase on) is 1.414 x 2 (amps), or 2.8 amps. Before proceeding, make sure the power is off and let any residual power supply capacitors discharge whenever motor circuits are connected or disconnected.

1. Adjust the output current to zero, either by pot adjustment, or serial command (depending on the product model/features).
2. Connect an amp meter(s) and motor as shown above.
3. Apply power.
4. Enable drive (method depends on model. See "E" command). The enable should eliminate "hold" reduction.
5. Increase the current setting until some amperage reading is obtained. Do not exceed the RMS current rating (2 amps in this example).
6. Adjust the "run" current. This is done at standstill. Methods for adjusting the current vary depending on the product model, as follows:

Method A: Programmable Current

AMS "programmable current" products have a digitally controlled potentiometer that is used for both hold and run current settings. The range is between 0 and 100 representing 0% and 100% of the full-scale drive current. Two "Y" command parameters control the hold and run values. For this procedure, set both values the same, i.e., "Y 40 40." Generally the preferred method is use of the peak value (one phase maximum) for micro step models and RMS (both phases on) for full/half step models such as the SAX or DAX.

1. Microstep Models with Programmable Current:
   - 1A. Set the resolution mode to "fixed" resolution (H 0).
   - 1B. Single-step until a maximum current on one phase is reached.
   - 1C. Use the "Y" command to obtain the desired current.

2. Full/Half Step Models with Programmable Current:
   - 2A. Single-step until equal currents on both phases is reached.
   - 2B. Use the "Y" command to obtain the desired current.

Fine tune using the Empirical Method (D) as required.

Method B: Peak Current, Single Potentiometer Models

The single turn potentiometer's position is proportional to the full current rating of the product. If necessary the driver is stepped until one phase is maximum and the other is at zero current (¼ step resolution is convenient).

1. Adjust the "run" current to the peak value of the product and matching motor. Fine tune using the Empirical Method (D) as required.

Method C: Peak Current, Dual Potentiometer Models

This procedure is implemented on dual-potentiometer products (DCB-264). Separate potentiometers are used to adjust the Sine (SIN) and Cosine (COS) outputs. When microstepping, the current values vary, reaching alternate "peaks" at two positions.

The general procedure is to position the motor "microstep" to the highest (peak) value of one phase, then adjust to the desired current.

Steps:

1. Before applying power, adjust both pots to full counterclockwise position (minimum current).
2. Attach a motor with a dual Amp-meter inserted.
3. Attach a power supply and apply power.
4. Start communication in single axis "dumb" terminal mode.
5. Enter the "E3" command.
6. Set microstep resolution to half step "H2."
7. Slowly increase the COS pot until some current reading is obtained.

8. Pre-adjust the SIN pot to an equal position (meter may not change).
9. Step the motor as required (+1 or -1) to insure maximum (peak) current.
10. Adjust the COS pot to the desired current.
11. Repeat steps 9 and 10 for the SIN current setting.
12. Enter several step indexes to assure reliable operation*

While half step resolution is recommended for simplicity, any step resolution may be used. Maximum rated output current for the DCB-261 is 1.2 amps/phase and must not be exceeded.

*Optimum amperage is the lowest current where the application indexing is reliable. Sometimes higher currents (still below the motor ratings) will decrease reliability. The motor temperature and driver heat sink temperature limit permissible currents.

13. Fine tune using the Empirical Method (D) as required.

Method D: Empirical Method, Minimum Current

The "empirical" method is the best approach for "final-tuning" the system and can/should be used for all AMS products. This technique is generally used for "final tuning" complete system configurations. When the best values are determined they can be used in future production, providing tolerances are sufficiently close. Once the system is assembled in its final form and the motion commands are sent to the motor:

1. Reduce the current by CCW rotation of the potentiometer(s) (by equal increments in dual potentiometer models, or by using the "Y" command available in programmable units).
2. Reduce the current setting until operation becomes erratic or undesirable.
3. Increase the current gradually until reliable operation is obtained, then increase the current equally by 10 to 20%.

In any of these adjustments, monitor motor temperature and insure that excessive heating does not occur. Larger motors require more time for temperature to stabilize. When a low hold current and short run cycle is used, heating effects are reduced.