DCB-242
USERS GUIDE

Revision Date: 03/02/2010

1) Introduction

Congratulations!

.....on your purchase of a DCB-242 Stepper Motor Driver-Controller Board. The DCB-242 will provide years of reliable, accurate and cost-effective motion control. As with all AMS products, the DCB-242 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-242. 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 Connections, Installation, Communication, Operating Parameters, Commands, and Specifications. We recommend that each section be reviewed prior to installation.

Although the DCB-242 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

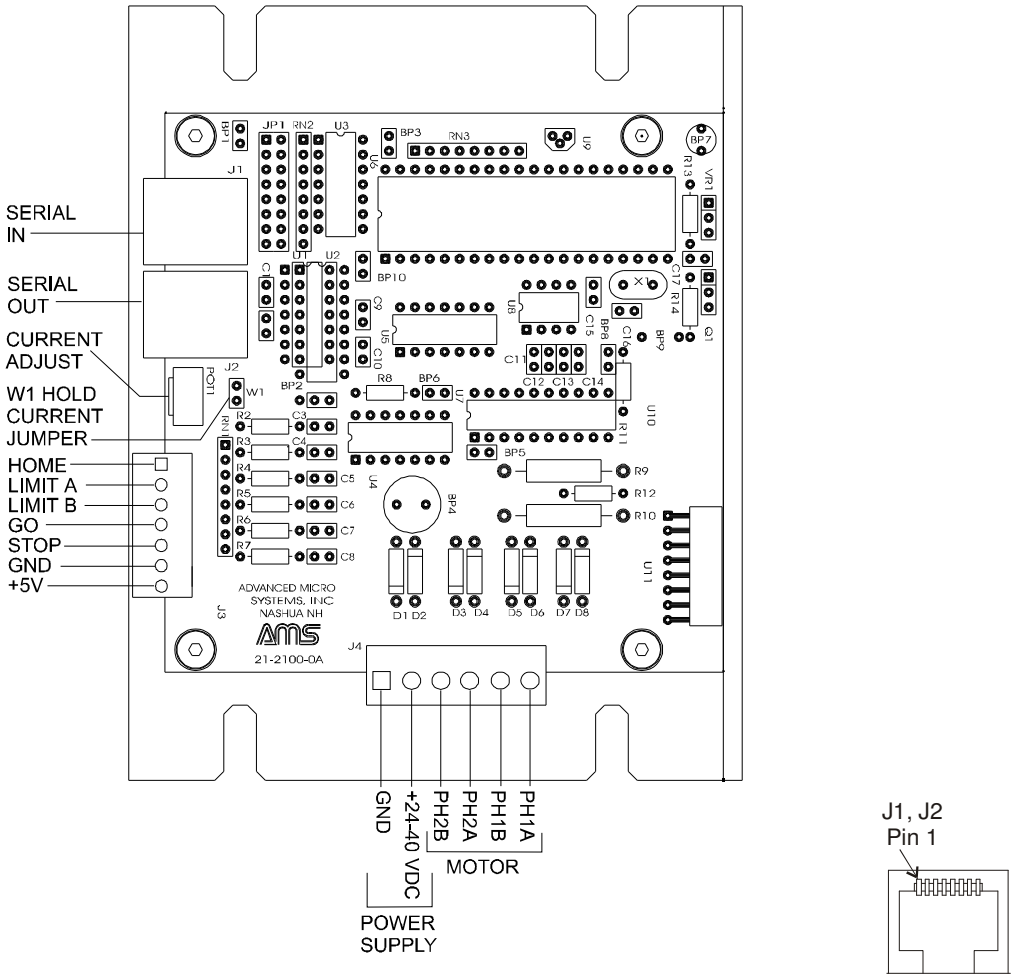
Product Overview

The **DCB-242** is a combination bi-polar chopper ***Driver*** and intelligent ***Controller*** board for operating small stepping motors. It is designed for low cost O.E.M. applications, yet it includes many enhanced operating features found in products costing much more:

- ***2 amp/phase chopper drive output***
- ***AMS' Award Winning controller***
- ***Single 24 to 40 volt power supply input***
- ***1/2 step resolution to 24k SPS***
- ***2k bytes of non-volatile memory***
- ***Limit, Home, Go and Stop inputs***
- ***Serial communication (1-32 axes)***
- ***Adjustable run current setting***
- ***Automatic hold current adjust***
- ***Programmable accel/decel ramping***
- ***Constant velocity commands***
- ***Heat-sink mounted***
- ***Mating connectors included***
- ***Free development software***

2) Hardware Connections

Assembly Drawing/Pin Description



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.

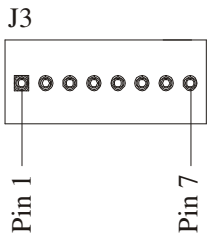
J1			J2		
Pin	Signal	Comment	Pin	Signal	Comment
1	J2-1	Not used	1	J1-1	Not used
2	GND	Power Gnd	2	GND	Power Gnd
3	RX-	+Data in	3	RX-	+Data in
4	TX-	+Data out	4	TX-	+Data out
5	TX+	-Data out	5	TX+	-Data out
6	RX+	-Data in	6	RX+	-Data in
7	5V	Power for serial adapter	7	N/C	Not used
8	PARTY	Enable party 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.

I.O. Connections (J3, JP1)

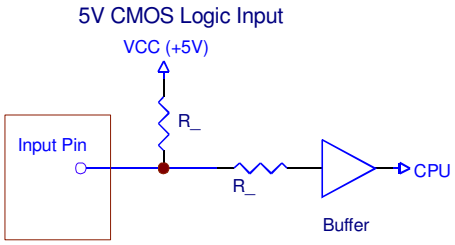
A seven contact connector (J3) provides inputs for the most commonly used I.O. signals. This modular connector is mapped as follows:

Pin	Signal
1	Home*
2	Limit A*
3	Limit B*
4	Go*
5	Soft Stop*
6	Gnd
7	+5V



*All these signals are inputs with a 5-volt range.

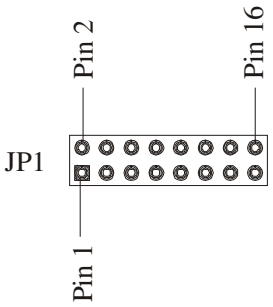
Typical Input Circuit



Optional

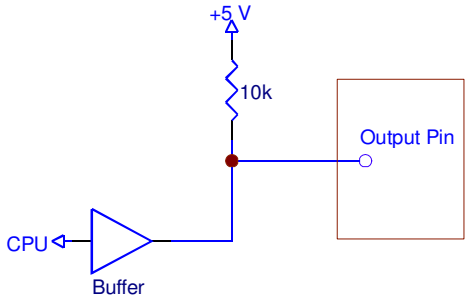
The DCB-242 has provision for some additional I.O. that can be provided by the user. An optional 16 contact header (JP1) is available to provide inputs and outputs as follows:

Pin	Signal	Type	Pin	Signal
2	Port 1	Input	1	Gnd
4	Port 2	Input	3	Gnd
6	Port 3	Input	5	Gnd
8	Port 4	Output	7	Gnd
10	Port 5	Output	9	Gnd
12	Trip	Output	11	Gnd
14	Reserved		13	Reserved
16	+5v		15	Reserved



Inputs are TTL with 10k pull up resistors.

Typical Output Circuit



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-242 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)

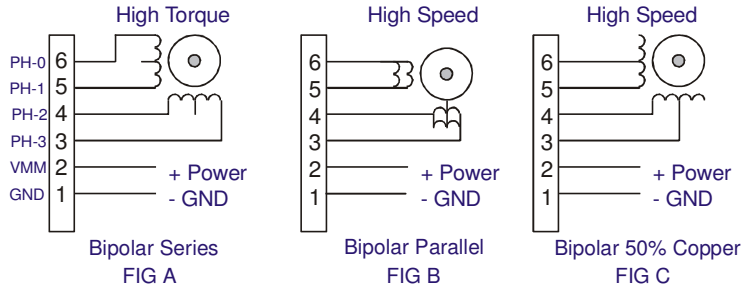


Fig. A: Series winding for higher torque and lower current. The inductance is 4 times that of the parallel mode, reducing the maximum obtainable speed.

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.

For more information, refer to the Addendum; "About Step Motor Current."

3) Getting Started

Required Hardware for Operation

Qty	Unit	Model #	Description
1	Axis	DCB-242	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-242.

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 POT1 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-242 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-242 is shipped with a "reduced" Hold current setting. The Hold current is based on a percentage (approximately 30%) of the Run current value that is set manually via the current adjust potentiometer (Pot1).

Set the defaults:

A. Enter the "E 2" command (enables reduction of hold current versus run current),
B. Enter the "S" command (save default to NV memory).

Install jumper W1 if zero Hold current is desired or remove W1 for the reduced Hold current feature.

Adjust Pot 1, using an amp meter, to set the Run current as follows:

C. Adjust Pot 1 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 Pot 1 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-242 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-242 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.

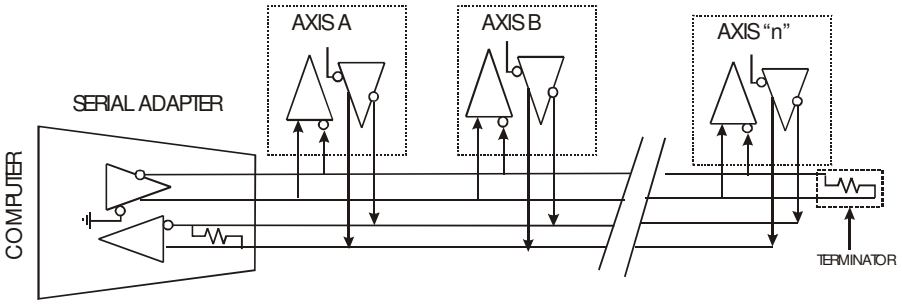
4) Communication Interface

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



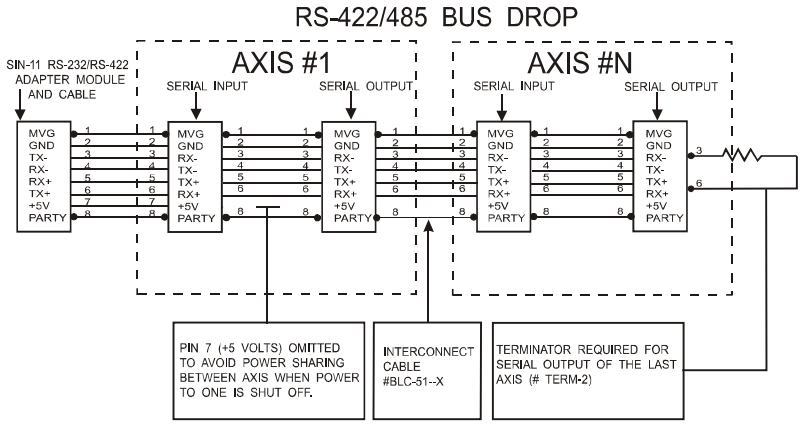
Communication hardware requires three components:

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

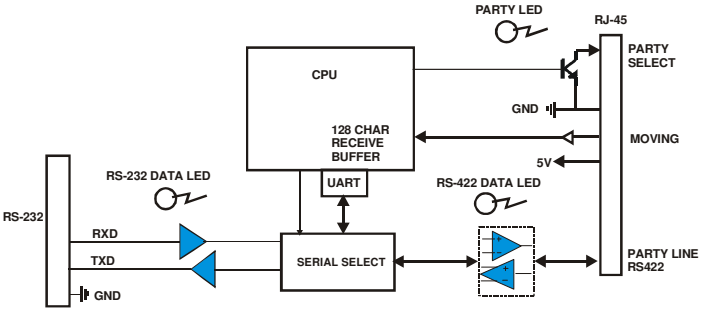
SIN-11 Intelligent Serial Adapter (Recommended)
The SIN-11 is an intelligent serial line converter that simplifies application software development and improves overall performance. Communication between connected controllers and the SIN-11 is 9600.

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.

Communications Software

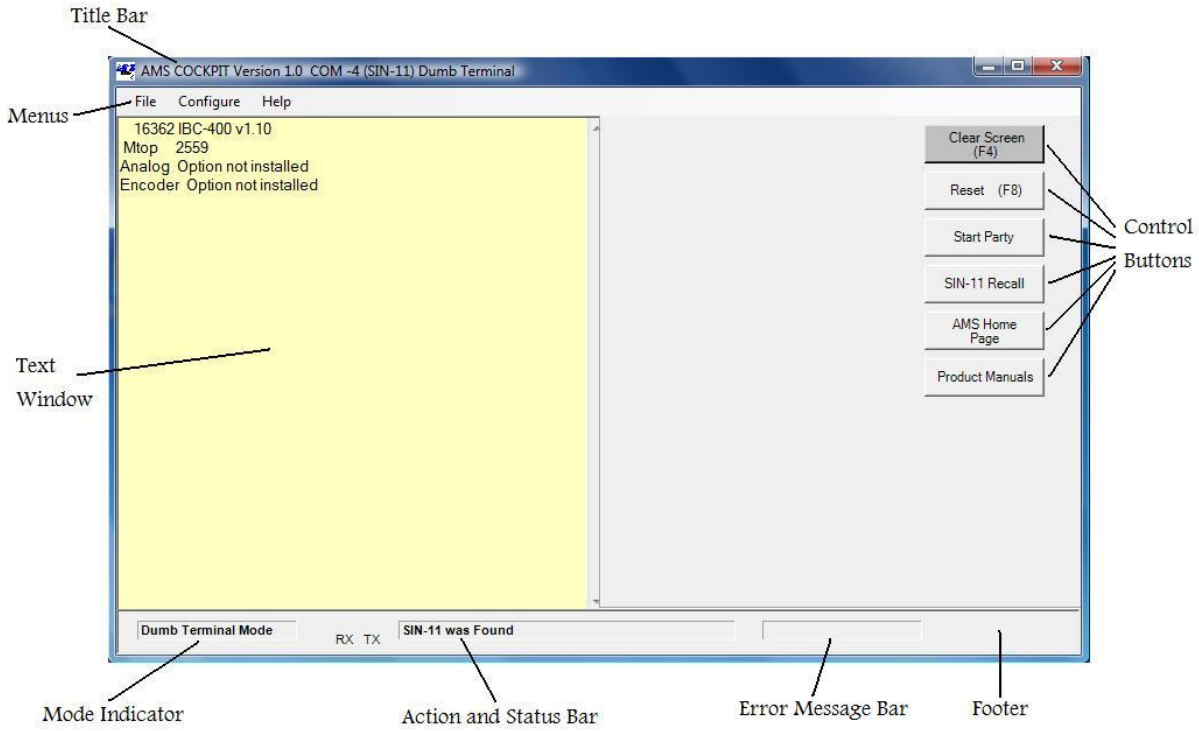
AMS offers the "AMS Cockpit" software that can be downloaded for free from the AMS website to assist customers in the implementation of their projects. It is compatible with Windows Operating systems. In addition to enabling communication, it includes some customized functionality, such as the downloading 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) Parameters / Memory / Command Execution

Reset/Initialize

Automatic hardware reset occurs each time the power is turned on or a command "^C" is issued from serial communication. During reset all inputs and outputs will be at a high state. After hardware reset all parameters are initialized to factory set default values. The communication mode is established at 9600 Baud. Additionally, checks for Party Line operation are initiated.

Resident non-volatile memory is accessed, and the parameters most recently stored by the "S" command are downloaded, replacing the standard defaults in the working registers of the controller. The following block of parameters are stored and recovered as a set:

Parameter	Standard Defaults
Initial Velocity (I)	400 (steps per second)
Slew Velocity (V)	5016 (steps per second)
Divide Factor (D)	1
Ramp Slope (K)	5/5
Auto Power Down (E2)	Reduced Hold, High Run
Limit Polarity	Low
Auto Position Readout (Z)	Off
Name (after reset)	Undefined

Note: Commands that modify these parameters use the working registers inside the controller. Actual non-volatile memory storage is initiated by the "Save" command. Once initialization is complete, Jog and Go inputs are active to allow jogging or a low pulse on the Go input to execute a program previously stored in non-volatile memory. A terminal or host is NOT required for these functions and may be initiated from the Auxiliary Input/Output connector.

Result Data

Some commands result in a numerical display. These consist of whole numbers that may have preceding spaces and are followed by a Carriage Return and Line Feed character. Negative numbers are preceded by the minus "-" sign.

Party Line Commands

During Party Line operation characters will NOT be echoed to the host until the proper "Name" (preceded by a ^J or ^Enter) is detected. All axes monitor concurrently the common TXD line from the host. Once the Name is received, the target axis will wake-up and start echoing as described above.

Instruction Execution

For each Motion command there are four cycles:

1. Entry
2. Execution
3. Result
4. Completion

All other commands have three cycles:

1. Entry
2. Execution

3. Result

In the idle state the controller continually tests for Jog, Go or Command input. The following describes each sequence operation that takes place on receipt of a command:

Cycle 1. Entry
The input commands and data are loaded via RS-232 or RS-422/485 interface. Command and data information is placed in a command line buffer as received. Editing is permitted in Single mode. ESCape aborts operation and returns to an idle state. A Carriage Return (^J or ^Enter for Party Line) 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/or 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
Half/Full Step	Until previous action is complete
Wait	Until previous action is 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. Another action command will be queued-up during the Completion cycle of a preceding Action command. The Execution and Result cycle of this Pending command is delayed until the Completion phase is finished. This interval is called the Pending Period.

During this Pending Period the command accepted is the one character interrupt (abort) command, limit switches (J2 pins 6, 8) soft stop input (pin 16) and home switch (J2 pin 4).

External indication of Pending Period end, Execution and Result cycle of the pending instruction is the Carriage Return. 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 command.

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

Note: All processes are aborted upon ESCape.

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 and Constant Velocity commands (inter-speed ramping) deceleration will be delayed until the previous ramp-to-speed has been completed.

Jog Speeds, Homing (Optional)

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 is mutually exclusive. A command canNOT be loaded while jogging and jogging can NOT be performed until the last command is complete. A command starts with the reception of the first command character.

High Speed Considerations

Step rates are controlled with a high degree of accuracy. As a result, step control is given priority over other processes. At high step rates this will manifest itself as a slight latency. The execution time increases when high step rates are active during command cycles.

An example might be reading positions while moving at a high speed. Usually this latency has little affect at step rates below 10,000 steps per second. At speeds approaching the maximum step rate the processing latency may have to be taken into account.

Non-Volatile Memory

The DCB-242 hosts a 2048 byte non-volatile memory for each axis. The non-volatile memory may be used to store User programs for future execution via the "Go" command.

Any number of programs may coexist, limited only by the available memory space.

The below memory map indicates that address locations are segmented into 8 pages and are accessible through direct read/write commands or cleared using the appropriate "C" command.

A special memory "Scratch Pad" of 64 bytes is accessed in location 128 to 192 for use in the working registers of the controller.

The EEPROM has a finite life of approximately 400,000 "write" cycles. Care should be used when writing to non-volatile memory to exclude unnecessary write cycles. For example, the Restore command ("^C" from a terminal) 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. Should you require a sequence of motions to be done without host attention, you may breakup the motions into subgroups rather than repeatedly programming the EEPROM. Use the Go from address command to execute the subgroups in the required sequence.

Use the Save command sparingly. Parameters are set so quickly that it is sufficient to just let the host download them.

Location	Description
(0-226)	User program or data storage
(256-2048)	User program or data storage
(227)	Configuration byte
(228)	Internal initial status byte. (Do NOT modify)
(229)	Divide factor (D)
(230,231)	Initial velocity low and high bytes
(232,233)	N/A
(234,235)	Slew speed (V) low and high bytes
(236,237)	N/A
(238)	Low speed jog value
(239)	High speed jog value
(240)	Ramp factor acceleration (K)
(241)	Ramp factor deceleration (K)
(242,243,244)	Trip
(245)	Hold current (not used on DCB-242)
(246)	Run current
(247)	Name for Party Line

Changing parameters should NOT be done by writing directly to the EEPROM, as the DCB-242 will not know that it was changed and may initiate an overwrite. Use the commands available to set parameters. Unlike writing, reading is non-taxing on the EEPROM.

Command/Program Mode

In the Command mode, commands are normally executed as soon as they are entered. The use of non-volatile memory allows storage of a list of commands. These stored program(s) can be triggered at power-up for automatic or repetitive operations by issuing a "G" command or by strobing the Go (input J1-4) to a logic low.

In the Program mode, the entered commands (now called instructions) are directed into the non-volatile memory. After leaving Program mode, the stored program(s) may be subsequently executed by entering the "G" (Go) command.

The following procedure assumes single mode communications with a standard (RS-232) serial interface using a common terminal:

The Program mode is initiated by entering "P aa" (Carriage Return). The desired start address "aa" is chosen by the User. Generally, address 0 is a good choice for the main program because a program located at address 0 can be started with a simple "G" (Carriage Return) or by strobing the "Go". Also the Go input begins execution at address location 0.

Once in the Program mode the current memory location is displayed on the terminal and instructions may be entered. As each instruction is entered, the location is displayed. All instructions have the same format as in the Command mode.

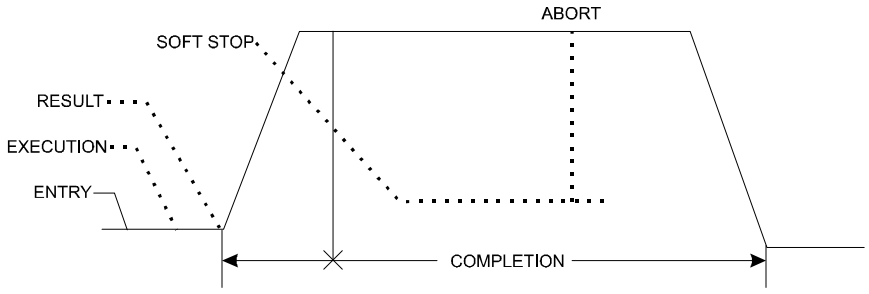
Terminating the Program mode is done by entering a "P." This will cause the end of program flag to be inserted and the controller will echo the pound (#) character. It will then return to the Command mode.

Several programs may co-exist in memory. Each program may be executed independently by issuing a "Go" command with the appropriate address. The length and quantity of programs may occupy the full 2k byte of memory space.

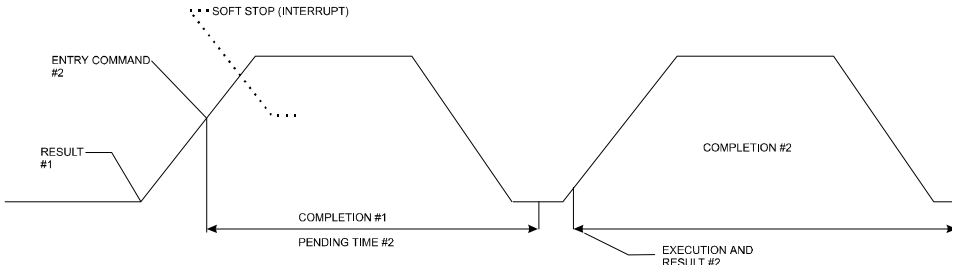
Note: The end of program indicator occupies one additional byte. A program sequence that will be "called" when a trip point is passed may be located at an address defined by the trip point.

Command Cycle Examples

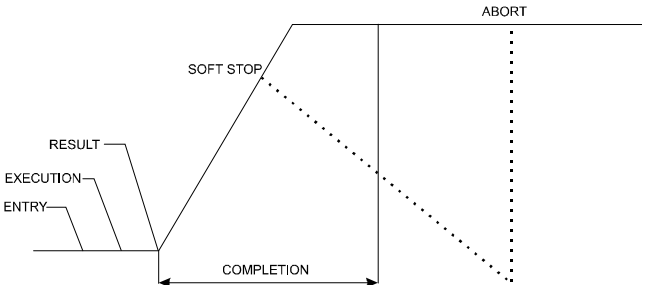
Index Cycle Resulting From +, -, R Commands



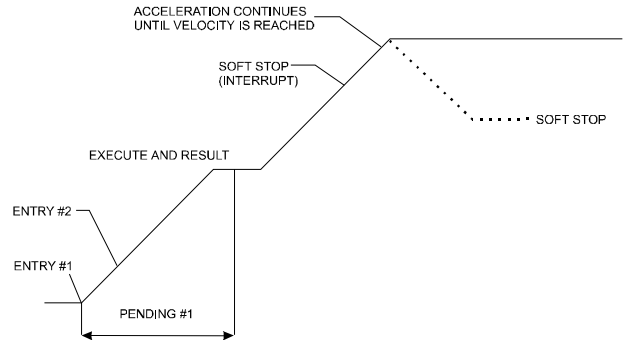
Queued Index Cycle Resulting From +, -, R Commands



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. The following times may be used as a start point for determining time requirements.

One receipt of the line feed, most commands execute in less than 1 additional millisecond, the exceptions are:

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

6) Commands

Program Command Description

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

Where:

Command	ASCII character (Keystroke)
Function	Functional description of command
Type	D= Default; initial parameter setting I= Immediate (direct execution) P= Program command; executable in stored program
NV	Non-volatile memory byte requirements in program
Mnemonic	Single character prefix used in multi-axis protocol; (prefixed by axis "Name" assignment in Party Line mode)
Data/Range 1	Affected parameters and valid numerical range of parameters
Data/Range 2	Same as Data Range 1 (as required)
Result	Information returned as a result of command execution or examination

Commands

Command	*Function*	*Type*	*NV Bytes*
ESC	Terminate Operation	Immediate	0

Mnemonic	*Data 1*	*Data 2*	*Result*
(Name) Esc. Character	None	None	Echo

ESC (Global Abort)

Global Abort terminates any active operation and forces the controller to revert to the idle state. Output drivers or ports are not affected. Stepping and position counter update will cease immediately without deceleration. Any program "running" will be terminated.

Any axis in the program mode will exit the program mode without creating the "end of program marker," therefore the escape character is useful in editing non-volatile program segments. In single mode a pound (#) sign is returned.

Note: Because the deceleration is immediate (without ramping) mechanical overshoot may result, especially with high speeds and/or inertia loads.

Command	*Function*	*Type*	*NV Bytes*
@	Soft Stop	Immediate, Program	1

Mnemonic	*Data 1*	*Data 2*	*Result*
(Name) @	None	None	#

@ (Soft Stop)

The Soft Stop command is useful as a gentle stop. It behaves differently, depending on how it is used. If the axis is moving it causes an immediate deceleration to a stop, based on the established deceleration K value. If one or more axis is running a program when this command is sent via the serial port, the running program(s) will terminate after deceleration. The soft stop may be embedded

within a program (in Program mode). During program execution, the encountered soft stop will not cause termination and is functionally equivalent to the "M 0" command.

Command	*Function*	*Type*	*NV Bytes*
^C	Reset Controller	Immediate, Program	0
	Mnemonic	***Data 1***	***Data 2***	***Result***
	(Name) ^C	None	None	None

^C (Reset)

Software Reset is a global RESET command. All axis ABORT immediately and a reset, equivalent to the power-up condition, is executed:

1. Down load default values from the non-volatile memory.
2. Set origin(s) to zero.
3. Calibrate motor current to hold value.
4. Assume an idle state waiting for GO pulse input, Jog input or serial command input.

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

A (Port) (optional hardware required)

The DCB-242 has 5 user ports. The onboard input or output buffers define the signal direction as follows:

Inputs
Port 1, port 2, and port 3 are logic inputs. The branch “G 2048” and loop on port “L” commands can use these inputs as part of a NV program. Each input is read using the “A 129” command. The number returned to the “Host” contains the states of all ports, and additional information. The voltages are inverted, so a 5 volt input level (unconnected) returns a zero.

Reading the port data provides the following information:

“A 129” Results

Data	Cause	Data	Cause
1	Low input to port 1*	16	Port 5 is low
2	Low input to port 2*	32	High if moving
4	Low input to port 3*	64	Trip port
8	Port 4 is low	128	Direction level

*The “host” can logically dissect this number to obtain the state of the 3 inputs by “AND”ing the value with 7.

Outputs
Ports 4 and 5 are buffered as general-purpose outputs; the Trip port (sometimes referred to as port 6) is usable as a general-purpose output. Caution must be used because it can only be toggled via the A64 command.

The command "A0" will reset all ports to their in-active (5 volt) state.

A (n)	Port 4	Port 5	Trip
A 0	Off	Off	No change
A 8	On	Off	No change
A 16	Off	On	No change
A 24	On	On	No change
A 64	No change	No change	Change state (toggle)

Command	***Function***	***Type***	***NV Bytes***
C	Clear and Restore	Immediate	0
	Mnemonic	***Data 1***	***Data 2***	***Result***
	(Name) C (n)	0-8 Page	None	Version

C (Clear and Restore)

The C1 command clears the entire non-volatile memory and restores the default parameter settings. This command should be used sparingly since the non-volatile memory has a finite life of approximately 400,000 write cycles.

Parameter	Standard Defaults
Initial Velocity (I)	400 (steps per second)
Slew Velocity (V)	5016 (steps per second)
Divide Factor (D)	1
Ramp Slope (K)	5/5
Auto Power Down (E2)	Reduced Hold, High Run
Limit Polarity	Low
Auto Position Readout (Z)	Off
Name (after reset)	Unchanged

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

To complete the restoration of Current control mode for the DCB-242, the appropriate E command (either E2 or E3) must be entered and saved (S) immediately following the use of the C1 command.

The following commands clear the corresponding page of non-volatile memory from the indicated address locations:

Command	Page#	Address
C2	2	256 to 511
C3	3	512 to 767
C4	4	768 to 1023
C5	5	1024 to 1279
C6	6	1280 to 1535
C7	7	1536 to 1291
C8	8	1792 to 2047

Command	*Function*	*Type*	*NV Bytes*
D	Divide Resolution	Immediate, Program	2
	Mnemonic	***Data 1***	***Data 2***	***Result***
	(Name) D (n)	Resolution 1-255	None	None

D (Divide Resolution)

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 3 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 or 3 may be helpful in producing smoother acceleration characteristics at slower slew speeds. The "D" command should NOT be changed while moving. Using a value of 2 or 3 may be helpful in producing smoother acceleration characteristics at slower slew speeds. The "D" command should NOT be changed while moving.

Command	*Function*	*Type*	*NV Bytes*
E	Enable Current Control	Immediate, Program, Default	2
	Mnemonic	***Data 1***	***Data 2***	***Result***
	(Name) E (n)	2, 3, 6, 7	None	

E (Enable Control)

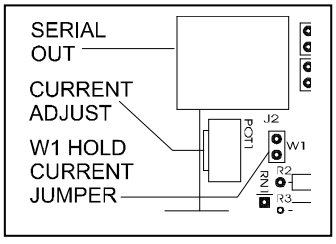
The E command controls the limit switch activation polarity and permits disable of the driver current, as follows: Refer to the Addendum; “About Step Motor Current” for more information.

E2: Limits the driver Hold current to the W1 setting (below) and is used for "normally open” limit switch activation.
E3: Enables the driver Hold current and is used for "normally open" limit switch activation.
E6: Disables the driver output and is used for "normally closed" limit switch activation
E7: Enables the driver output and is used for "normally closed" limit switch activation.

Limit Switch Activation
A limit switch that is normally open is called active low, i.e., a low voltage will inhibit motion in one direction.

“E” Command	Hold Current	Run Current	Limit Switch Polarity
E2	W1 setting	High	Activate low
E3	High	High	Activate low
E6	W1 setting	High	Activate high
E7	High	High	Activate high

W1 Jumper Setting	Hold Current
Installed	Zero
Not installed	Reduced



Command	*Function*	*Type*	*NV Bytes*
F	Find Home	Immediate, Program, Default	2

Mnemonic	*Data 1*	*Data 2*	*Result*
(Name) F (n,d)	SPS (18-23,000)	Direction (0-1)	None

F (Find Home)

The special Find Home algorithm is intended to eliminate mechanical hysteresis typically found in many switches and encoders. 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.

1. 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; 18 to 23,000 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.

2. 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	Direction 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

Command	*Function*	*Type*	*NV Bytes*
G	Execute Program	Immediate, Program	3

Mnemonic	*Data 1*	*Data 2*	*Result*
(Name) G (a,t)	0-2048	None	None

G (Go)

The Go command executes a User programmed sequence starting at a predefined address location. Although most programs will start at "0," the User can start at another address. That address however, MUST begin at a stored instruction address. The Trace option is useful in debugging single axis programs. If trace is a 1, 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. Address locations between 225 and 255 are reserved for parameter storage and may not be used in programs.

There is a special case for the Go instruction. If the address is specified as 2048 (above the last non-volatile address), the control signal will 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 at address 256, with 16 character (byte) intervals. This instruction is analogous to "on PORT go to."

Input Port State				Address
P1	P2	P3	P4	Location
1	1	1	1	256
0	1	1	1	272
1	0	1	1	288
0	0	1	1	304
1	1	0	1	320
0	1	0	1	336
1	0	0	1	352
0	0	0	1	368
1	1	1	0	384
0	1	1	0	400
1	0	1	0	416
0	0	1	0	432
1	1	0	0	448
0	1	0	0	464
1	0	0	0	480
0	0	0	0	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.

Command	***Function***	***Type***		***NV Bytes***
H	Set Calibration Flag	Immediate		
	Mnemonic	***Data 1***	***Data 2***	***Result***
	(Name) H	16	None	None

H (Calibrate Timing)

H 16

Earlier versions of the SMC-C24 (microprocessor) used in the DCB-242 had an 11 Mhz clock and the value used in the “W” (Wait) command was correct. New devices are specified at 14.7 Mhz and the calibration is off by 4/5. Normally, the designer has compensated for this error by adjusting the data value. Software V2.07 provides for calibration to 10 Ms, while being backward compatible with earlier versions (8 Ms).

Setting calibration on:

1. Enter the command H 16. This sets the calibration flag.
2. Issue the S (save) command. The mode is saved in NV memory.

The CLEAR (“C1”) command will cause the calibrate flag to be reset, and steps 1 and 2 must be repeated.

To determine the calibrate mode, the EXAMINE (X) command may be used. When in dumb terminal mode, a lower case “w” will appear as the first character of the display response.

Command	Function	Type	NV Bytes
I	Initial Velocity	Immediate, Program	3
	Mnemonic	**Data 1**	**Data 2**	**Result**
	(Name) I (n)	SPS 18-23,000	None	None

I (Initial Velocity)

The initial velocity specifies the start and stop speed for the motor. The first velocity of the acceleration ramp is specified by the I parameter. To achieve an optimum value the following variables must be taken into account: motor size (rotor inertia), system inertia, starting torque to overcome friction (stiction), and motor or system resonance's. Generally values between 50 and 1200 SPS are appropriate.

The initial velocity applies to:

1. All index commands (+, -, R).
2. Start speed used in constant velocity (M).
3. Decelerate to 0 in constant velocity or soft stop.
4. Final phase of home routine.

The Examine (X) command displays velocity information. This parameter is a user default. After setting the value and issuing the Save command, it will be stored for future power-ups. A value of 400 SPS is preset on issuance of the "C8" (clear) command.

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

J (Jump To Address)

Example: 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 may be used only within a program. This instruction may NOT be nested. Only one jump counter is available for use at any given time.

Command	Function	Type	NV Bytes
K	Ramp Slope	Immediate, Program	2
	Mnemonic	**Data 1**	**Data 2**	**Result**
	(Name) K (n)	Accel (0-127)	Decel (128-255)	None

K (Ramp Slope)

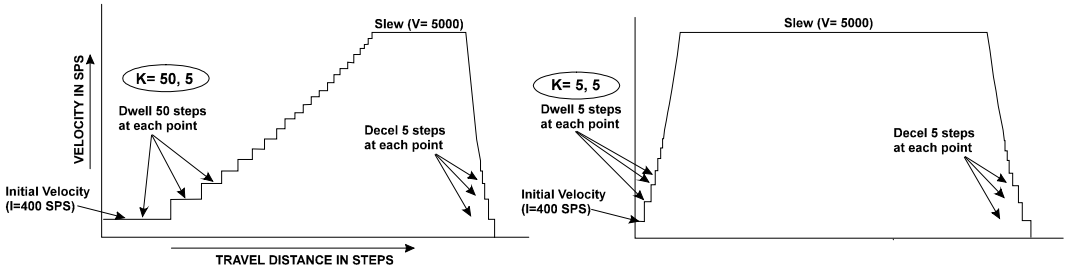
Specify the ramp acceleration and deceleration time.

The "K" command is used to adjust the ramp slope during the motor acceleration or deceleration. The profile or shape of the acceleration/deceleration curve is defined by an internal lookup table. Depending on the values of initial and slew velocities, 0 to 157 discrete velocities may be attained to adjust 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.

Examples: Two ramped indexes, each 2000 steps with I=400, V=5000, but different “K” values; K50 5 and K5 5.



Note: Values of "K" for deceleration mirrors the acceleration ramp. If a value of less then 127 is entered for "K", then both ramps assume the same slope. To alter the deceleration ramp, it is necessary to enter values between 128-255.

Example:
Entering a value of "130" would change only the deceleration slope and would cause the deceleration ramp to have a value of "2" steps on each point of the deceleration portion of the ramp table.

The K command can be issued:

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

Command	***Function***	***Type***		***NV Bytes***
L	Loop on Port	Program		3
	Mnemonic	***Data 1***	***Data 2***	***Result***
	(Name) L (a,c)	Address (0-2648)	Condition (0-9)	None

L (Loop On Port)

The Loop command (option with DCB-242-I) will test the specified input port for the required condition . If the port is NOT at the required level, the program will jump to the specified address. If the address is to a previous instruction, the program will loop until it becomes the specified level. The program will then continue to the next step. Input ports are tested according to the below table:

Condition	Wait For
0	Port 1 Low
1	Port 1 High
2	Port 2 Low
3	Port 2 High
4	Port 3 Low
5	Port 3 High

There is also an additional feature for implementing a "wait until" function. The standard loop tests the condition every 2-3 Ms. If the unique address is 2048, the controller executes a tight loop at this instruction while 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 non-volatile program execution.

Command	***Function***		***Type***	***NV Bytes***
M	Move at Constant Velocity		Immediate, Program	3
	Mnemonic	***Data 1***	***Data 2***	***Result***
	(Name) M (±)	SPS ± 23,000	None	None

M (Move At Constant Velocity)

Move at the specified velocity and direction “forever."

The motor will ramp up or down to a constant step velocity and motion will continue at the given speed until a new velocity is entered. The specified slew speed is in steps per second. Ramp parameters (K command) may be modified prior to each velocity command, allowing different ramp slopes.
The direction is specified by the sign preceding the velocity. Decelerating from full speed in one direction to full acceleration speed in the opposite direction can be accomplished with this single command.
Examples:

M 1000

Move in the + direction at 1000 SPS starting at the initial velocity then accelerating to the 1000 SPS slew speed.

M -4000

Decelerate to initial velocity (from 1000). Stop and change direction. Accelerate to 4000 SPS in the “-” direction.

Motion may be stopped 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. Position Trip points may be used. If the encoder feedback option is implemented, stall supervision is functional.

Command	Function	Type	NV Bytes
O	Set Origin	Immediate, Program	4
	Mnemonic	**Data 1**	**Data 2**	**Result**
	(Name) O (±n)	Position (± 8,388,607)	None	None

O (Set Origin)

The Set Origin command resets the internal 24 bit position counter to the specified value. Base position for the Relative mode is zero. Signed numbers are used. Hardware reset clears to zero.

Command	Function	Type	NV Bytes
P	Program Mode On/Off	Immediate	0
	Mnemonic	**Data 1**	**Data 2**	**Result**
	(Name) P (a)	Address (256-2047)	None	None, #

P (Program Mode)

This Program command allows composition and storage of “program sequences" in non-volatile memory. Program mode allows entering commands for future execution by use of the Go command or External go signal. Existing programs are overwritten as new instructions are stored. Entering a second "P" command will terminate the Program mode, and insert an end of program marker in the stored program before returning to the Immediate Command mode. While in Program mode, commands and data are directed into the non-volatile memory. The address specifies the start point in non-volatile memory where the application program will reside. As instructions are entered, the address counter is updated and displayed. Any number of independent program segments can coexist. These can be accessed via Jump, Loop, Go or other special instructions.

Special Locations, "Shadow Memory"
Sixty-four bytes of the program are configured as FAST memory. Locations 128 thorough 192 are downloaded from the external EEprom to internal RAM at power-up. Instructions executing in this segment run faster then other locations that fetch from relatively slow EEprom (1 Ms. per byte). Programmers should reserve this segment for time critical code. The shadow RAM is not actually written to non-volatile memory until the Store command is issued. Host computers may download subroutines without concern about wearing out the EEprom. Locations 256 through 511 are predefined if the "G 2048" command is used in an application.

The Trip command can jump to any location between 0 and 255. Editing of programs should be done WITH CAUTION. The general programming sequence is to (1) start programming at desired address, (2) enter new instructions, and (3) terminate programming with the ESC command. (This will cause a return to the Command mode without inserting the end of program marker).

Note: Do not attempt to use locations above 1792 for programs (default storage).

Example program in terminal mode (each command is terminated with Carriage return):

Response	Command	Comment
	P 0	Enter program mode.
0 (echoed)	+5000	Index 5000 steps.
5	-5000	Index -5000 steps.

10	P 0	End program, insert program marker One program has been entered and stored in non-volatile memory. You may now insert another routine.
	P 100	Start program at memory location 100.
100 (echoed)	O	Set origin to zero.
101	R 2000	Move to position 2000.
106	R - 500	Move to position –500.
111	J 101 9	Repeat sequence 10 times.
	P 0	End program. You may now list the programs.
	L 0 5	
	L 100 5	You may now execute the programs.
	G 100	You may now edit the programs.
101	R 3000	Change position from 2000 to 3000.
	P 101	
106	<ESC>	Terminate edit without “end” mark.

While most commands can be part of a program, some may not be necessary or impractical while some should just be avoided.

Command	Reason
(C) Clear	Never use; erases programs.
(D) Resolution	User default from non-volatile memory at power-up.
(E) Delay	User default from non-volatile memory at power-up.
(I) Initial Velocity	User default from non-volatile memory at power-up.
(K) Ramp Slope	User default from non-volatile memory at power-up.
(Q) Query	N/A
(S) Store	No application, non-volatile memory wear.
(T) Trip	User default from non-volatile memory at power-up.
(V) Slew Speed	User default from non-volatile memory at power-up.
(X) Examine	N/A

Command	***Function***	***Type***	***NV Bytes***
Q	Query (List) Program	Immediate	0

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

Q (Query Program)

This command will produce a list (disassembled) of the instructions stored in non-volatile memory using the format:

"ADDRESS" "INSTRUCTION" "DATA 1" "DATA 2"

1. The values will be displayed only if applicable to the particular instruction type.
2. One instruction will be listed at a time.
3. The space key will advance the address and list more instructions.
4. Listing is terminated when an "end of program" marker is found or the ESC character is received.

Command	*Function*	*Type*	*NV Bytes*
R	Index Relative to Origin	Immediate, Program	4

Mnemonic	*Data 1*	*Data 2*	*Result*
(Name) R (dd)	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).

Command	*Function*	*Type*	*NV Bytes*
S	Store Parameters	Immediate, Program	1

Mnemonic	*Data 1*	*Data 2*	*Result*
(Name) S	None	None	None

S (Store Parameters)

The S (Store) command will force a write sequence to non-volatile memory. This command should be used to save new operational parameters or to download motion programs. The S command should not be embedded internally as part of a program but rather used to save the program after loading.

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

Parameter	Standard Defaults
Initial Velocity (I)	400 (steps per second)
Slew Velocity (V)	5016 (steps per second)
Divide Factor (D)	1
Ramp Slope (K)	5/5
Auto Power Down (E2)	Reduced Hold, High Run
Limit Polarity	Low
Auto Position Readout (Z)	Off
Name (after reset)	Undefined

All of these parameters are saved as a block from the working registers in the SMC-C24. Frequent use of this command should be avoided, as memory longevity may be affected.

Note: Observe the precaution of writing too frequently to non-volatile memory.

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)	None	None

T (Trip Point)

The DCB-242 has a programmable "Trip Point" feature. During moves the current position is compared to the Trip position at each step. The Trip output will be alternated each time the Trip Point is reached or passed.

When in the Program Run mode, a User program located at address 200 will be automatically executed (called). On completion of this sub program, a return to the call point will be executed. When the Trip Point is enabled, available top step speed capability is reduced by approximately 10%. A value of 0 will disable the Trip Point trigger. A value of -0 (minus zero) will set the Trip Point to 0.

The Trip Point is displayed using the examine command. Default = 0 (off).

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 (18-23,000)	None	None

V (Slew Velocity)

This command sets final Slew Velocity after ramping up to speed. The final output velocity is divided by the value of "D." This value is independent of constant velocity (ramp completed), jog or home speeds and is used for indexing absolute or relative.

The following commands use this parameter:

R (Relative index)
+ (Plus index)
- (Minus index)

The following functions do not use or affect this parameter:

J (Jog)
F (Find home)

The DCB-242 also allows specifying speeds in step time values. The actual step clock period may be directly controlled, rather than in steps per second. This allows fine control of step rates, especially at lower speeds. The DCB-242 is placed in this mode by using a negative value with the “V” command, and restored to the SPS mode with positive values. Switching between modes will not affect speeds previously specified. In the period mode, all subsequent commands that relate to speed must use period values for data (positive data).

These include:

V (-65535 to -54) Slew velocity
I (65535 to 54) Initial velocity
M (65535 to 54) Constant velocity

F (65535 to 54) Find home
B Jog Speeds (minimum obtainable value is 160 SPS)

The data range is between 54 (23,000 SPS) and 65,535 (18.76 SPS).

The formula for determining step rate is:
N = 1228800 / SPS

As with all speed commands, the output rate is divided by the value specified with the "D" command.

Command	***Function***		***Type***	***NV Bytes***
W	Wait (n) Milliseconds		Immediate, Program	3
	Mnemonic	***Data 1***	***Data 2***	***Result***
	(Name) W (n)	0.0075 Sec. (0-65.535)	None	None

W (Wait)

WAIT n milliseconds. (See related command H16)

The controller will remain in an idle state for the specified amount of time. The Wait command, if issued while indexing (as a result of a R,+,- or F command), will NOT start until all motion is complete. Using this command with 0 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 operations, executed during wait commands, will increase the delay time by as much as 14 times normal value. The result will NOT be available until the delay is complete.

Command	***Function***		***Type***	***NV Bytes***
X	Examine Settings		Immediate	0
	Mnemonic	***Data 1***	***Data 2***	***Result***
	(Name) X	None	None	Display Setting

X (Examine Parameters)

The Examine command displays the parameter settings. This command will produce two different responses, depending on the mode of Motion command operation. When in the non-Party Line (Single) mode the display is as follows:

K=kk, I=ii, V=vv, (T=tt) nn [Carriage Return, LF]"

Where:
kk = Ramp factor
ii = Initial velocity divided by "D" (calculated)
vv = Slew velocity divided by "D" (calculated)
tt = Trip Point (if enabled)
nn = Name of Party Line

It is assumed the host computer will interpret the data string for processing.

Command	Function	Type	NV Bytes
Z	Read and Display Current Position	Immediate, Program	2
	Mnemonic	**Data 1**	**Data 2**	**Result**
	(Name) Z	Readout Mode (0-1)	None	None

Z (Read Position)

Read and display the current position. During motor move commands the value will change depending on the direction of travel. The position counter is reset by the "O" (Set Origin) command. The "Z1" command enables a continuous readout, via the serial interface. 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 "on" if a Save command is issued. This mode is only practical using a single axis protocol.

Command	Function	Type	NV Bytes
[	Read NV Memory	Immediate	0
	Mnemonic	**Data 1**	**Data 2**	**Result**
	(Name) [	Address (0-2047)	Number (0-255)	Displayed Values

[(Read Non-Volatile Memory)

The Read Non-Volatile Memory command allows the User to display any byte of the 2047 byte external non-volatile memory. The address specifies the desired location to access.

Command	Function	Type	NV Bytes
]	Read Limits, Hardware	Immediate, Program	1
	Mnemonic	**Data 1**	**Data 2**	**Result**
	(Name)]	0-1	None	None

] (Read Limits/Hardware)

This command allows examination of 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
Controller	Lb	La	Hm	*	Qs	Fb	En	Half

Where:

La= Limit "a" switch
Lb= Limit "b" switch
Hm= Home switch (32=low input)
*= Always low
fb+,fb-= Encoder feedback input level (n/a: SAX/DAX)
En= Enable output level
Half= Half step output level

If the motor is disabled and limit switches are inactive the result will be 0.

Command	Function	Type	NV Bytes
+	Index in the 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 a positive direction for the specified distance. The motor will ramp up, slew, and then ramp down per the previously set parameters. The range is 0 to 16,777,215 steps. The position counter will overflow at 8,388,607. A "+" direction index is defined as a clock-wise rotation of the motor as viewed looking toward the motor mounting flange.

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

- (Index in Minus Direction)

Step in a negative direction. Except for the direction, this command behaves exactly as the "+" command above.

A "-" direction index is defined as a counter clock-wise rotation of the motor as viewed looking toward the motor mounting flange.

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 Read Moving Status command is used to determine the current moving and mode status. These status bits are converted to a decimal number (0-255) in Terminal mode.

The status byte contains the current status of the controller as follows:

Bit	Decimal	Status
0	1	Moving - indicates axis moving
1	2	Constant - high in constant velocity
3	8	Homing - homing routine is active
4	16	Slewing - ramping complete

Note: Other bits in this result should be ignored.

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

\ *(Write To Non-Volatile Memory)*

The Write to Non-Volatile Memory command allows the programmer to directly modify any byte in the memory. The life expectancy of the non-volatile memory may be affected by excessive use of this command.

Note: Non-volatile memory has a finite life of approximately 10 years for data retention and 460,000 write cycles.

7) Specifications

Electrical

Output Current (max).....1.2 Amps
Chopping Frequency.....20kHz
Input Voltage.....+24 to 40Vdc
Motor Step Resolution.....Half Step
Non-Volatile Memory.....2k Bytes
Position Counter.....±8,388,607
Baud Rate.....9600

Input Signals: Limits, Home, Go and Soft Stop (J3)

Signals	Min	Typ	Max	Units
VCC Supply (internal)	4.7	5.0	5.5	Vdc
High Input Voltage	2.7	3.15	5.5	Vdc
High Input Current			40	Ua
Low Input Voltage	-0.7		0.9	Vdc
Low Input Current			.6	Ma

Note: These inputs use 74hct14 IC's

Signal Specifications Optional I/O Ports

I.O. Signals (JP1)	Min	Typ	Max	Units
Inputs (Ports 1,2,3)				
High Input Voltage	2		5.5	Vdc
High Input Current			1	Ma
Low Input Voltage	-0.7		0.8	Vdc
Low Input Current			2	Ma
Outputs, (Ports 4,5,6)				
Output Voltage			5	Vdc
Output Current			16	Ma (cont.)

Note: These I/O use 7407 IC's

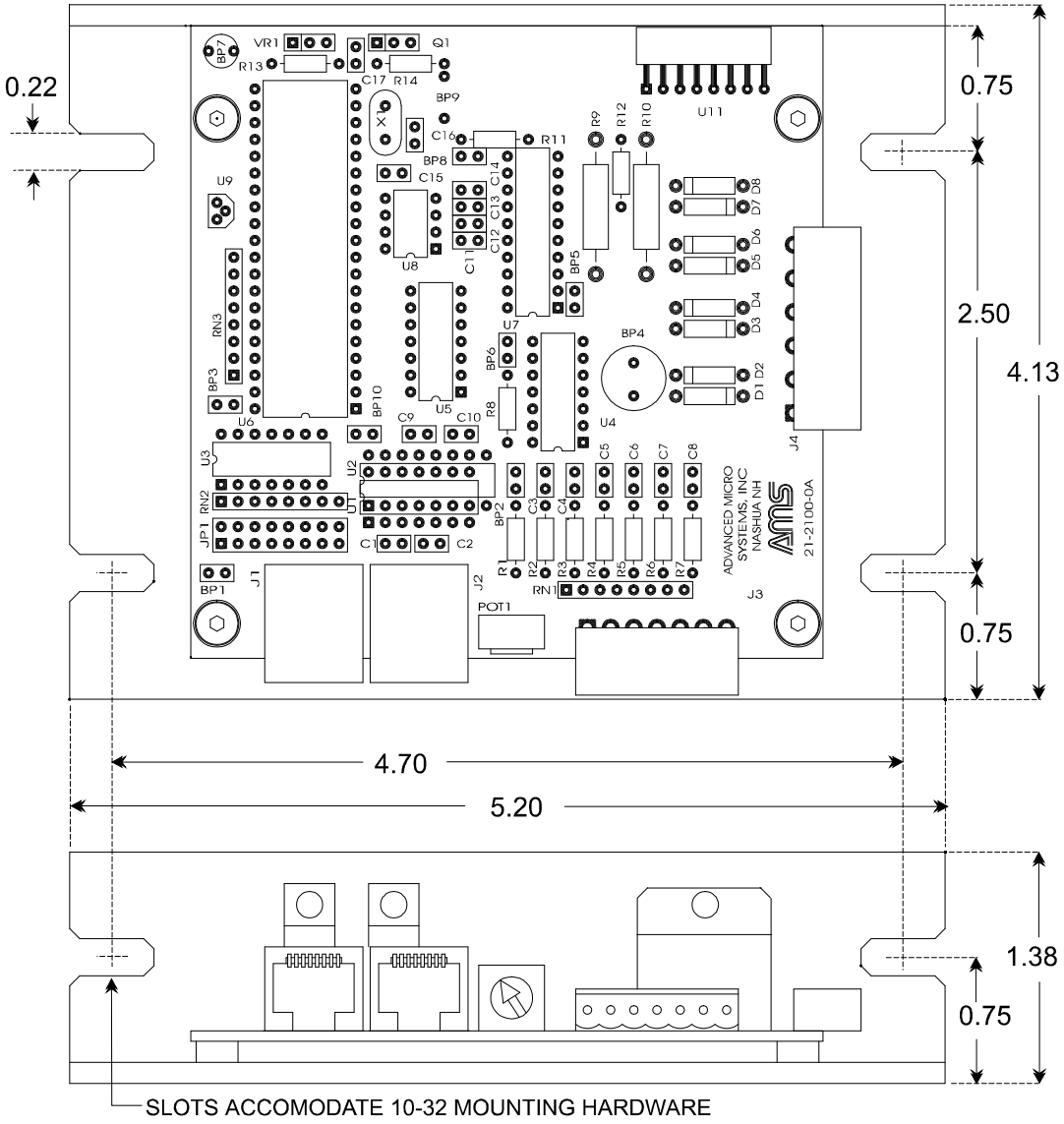
Environmental

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

Physical

Size.....4.13 X 4.26 X 1.364
Weight.....8.0 oz.

Dimensional Drawing



8) Addendum

Command Summary

MNEMONIC / COMMAND		DATA 1	RANGE	DATA 2	RANGE	NV	D	M	P	U
+	INDEX IN "+" DIRECTION	STEPS	0-16777215			4		⊗	⊗	
-	INDEX IN "-" DIRECTION	STEPS	0-16777215			4		⊗	⊗	
ESC	ABORT/TERMINATE									⊗
@	SOFT STOP					1			⊗	⊗
^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			1				⊗
^	READ MOVING STATUS					1				⊗
A*	PORT	BINARY	0-129			2			⊗	⊗
C	CLEAR AND RESTORE	PAGE	1-8				⊗			
D	DIVIDE RESOLUTION	RES.	1-255			2	⊗		⊗	
E	ENABLE CONTROL	N VALUE	2 or 3			2	⊗		⊗	
F	FIND HOME	SPS	18-23000	DIRECTION	0-1	3		⊗	⊗	
G	GO	ADDRESS	0-226,256-2048	TRACE	0-1	3			⊗	
H	CALIBRATE TIMING					1			⊗	
I	INITIAL VELOCITY	SPS	18,23000			3	⊗		⊗	
J	JUMP	ADDRESS	0-225,2047	N+1 TIMES	0-255	4			⊗	
K	RAMP SLOPE	ACCEL	0-127	DECEL	128-255	2	⊗		⊗	
L*	LOOP ON PORT	ADDRESS	0-26,256-2048	CONDITION	0-8	3			⊗	
M	MOVE AT CONST. VEL.	SPS	±18-23000			3		⊗	⊗	
O	SET ORIGIN		±8388607			4			⊗	⊗
P	PROGRAM MODE	ADDRESS	0-226,256-2048							
Q	QUERY PROGRAM	ADDRESS	0-2047							
R	INDEX TO POSITION	POSITION	±8388607			4		⊗	⊗	
S	STORE PARAMETERS					1	⊗			
T*	TRIP POINT	POSITION	±8388607			4	⊗		⊗	
V	SLEW VELOCITY	SPS	18-23000			3	⊗		⊗	
W	WAIT, (DELAY)	0.0075 SEC	0-65535			3			⊗	⊗
X	EXAMINE PARAMETERS									⊗
Z	READ POSITION	CONTINUOUS	0-1			2				⊗

*Available with optional Auxiliary I.O.

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.

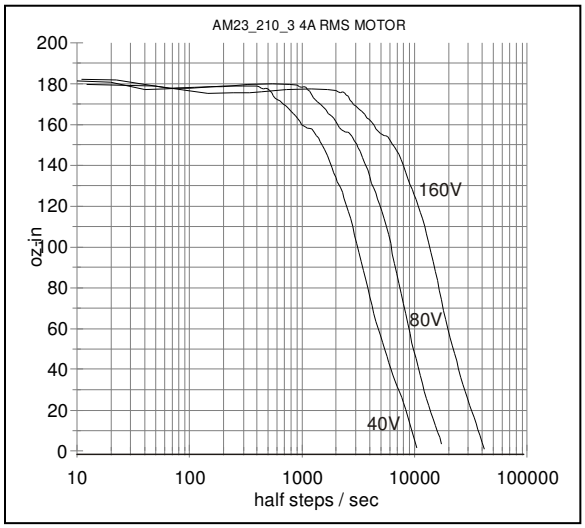
The step (or stepper, or stepping) motor is a digital "synchronous" motor with a pre-designed number of "steps" per revolution. The most common motor has 200 full steps per revolution. Simple driver electronics can subdivide these steps into ½ step or more complex "microsteps."

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.

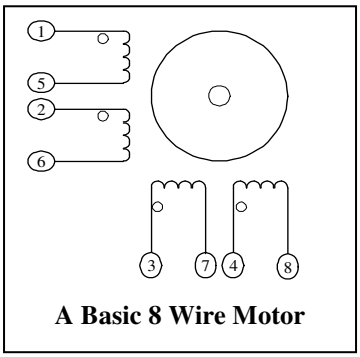
As the step motor current is reduced below the rated current, the torque output is reduced and eventually the motor will stall. The ideal current setting minimizes heating of motor and electronics, increases reliability, and reduces power supply requirements. Motors run more quietly and resonance effects 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

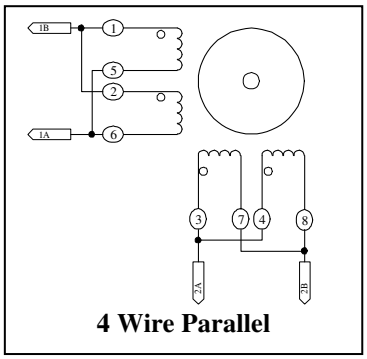


A Basic 8 Wire Motor

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

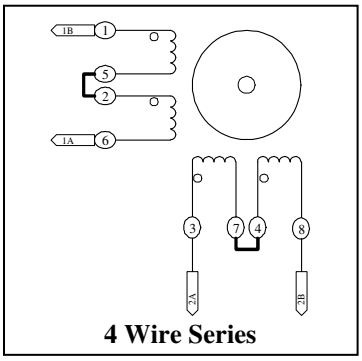
4 Wire Parallel
The high-speed model implements parallel coil connection. Two coils connected in parallel result in the following for each of the two phases:

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

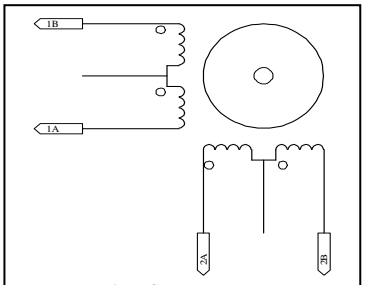
B. 4 Wire Series
Changing to a series design, we have two pairs of two coils connected in series. Each has:

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

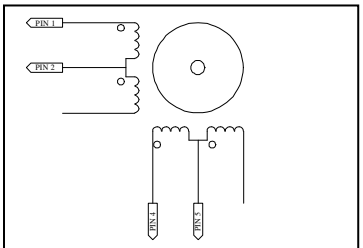
A 6 wire motor is equivalent to the 4 wire series motor.

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.



6 Wire, Half Copper

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.

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

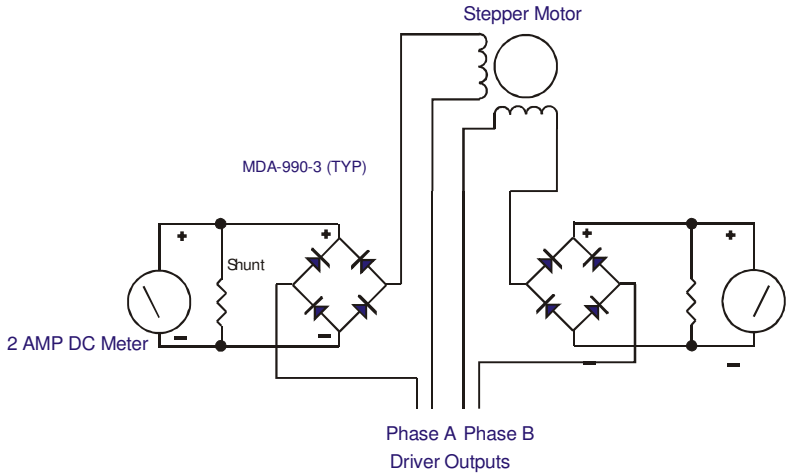
# Leads	Winding Connection	Per Phase Bipolar Current	
		Full Step (Total)	Microstep RMS (Peak)
4 wire	Parallel	4.0A (8.0A)	4.0A (5.66A)
4 wire	Series	2A (4.0A)	2.0A (2.8A)
6 wire	Unipolar	4A per phase	NA
6 wire	Series	2.8A (5.6A)	2.8A (4A)
6 wire	50% Copper	4A (8A)	4A (1.7A)

- 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-242	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, which is 2.8 amps in this example. 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 and DCB-612). Separate potentiometers are use 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-264 is 3.75 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%.

For dual potentiometer models, both potentiometers must be adjusted by equal amounts. Note that the "E3" command must be issued if the motor has been stepped to a non-RMS position. Periodically use the "E3" sequence to balance the two (SIN/COS) currents.

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.