

CMAX-410 / CMAX-810 Revision Date: 05/02/06

***CMAX-410 / CMAX-810
USER GUIDE***

CERTIFICATE

ELECTROMAGNETIC COMPATIBILITY

Manufacturer : **ADVANCED MICRO SYSTEMS Inc.**
Address : **2 Townsend West**
Postal code, Place : **Nashua, NH 03063-1277**
Country : **United States of America**

Applicant : **MOTION CONTROL TECHNICS**
Contact person : **Mr. T.S. van Duffelen**
Address : **Jan Glynisweg 32H**
Postal code, Place : **1702 PC Heerhugowaard**
Country : **The Netherlands**

Electrical apparatus : **Stepping motor controllers**
Trademark : **Motion Control Technics**
Type designation : **VPMC motor control system CMAX-410, CMAX-810
Microstepping driver CDR-4MPS, CDR-6MPS
Stepping motor AM23-210-3, AM34-620-3**

Environment : **Industrial**

Standards

EN 55011:1991 Emission - Class A
EN 61000-3-2:1995 Harmonic current emissions

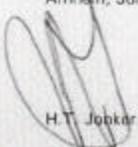
EN 50082-2:1995 Generic immunity standard, from which:

- EN 61000-4-2:1995** Electrostatic discharge (ESD) immunity
- EN 61000-4-3:1996** Radiated Electro-Magnetic field immunity
- ENV 50204:1995** Pulse modulated radiated EM-field immunity (IGSM)
- EN 61000-4-4:1995** Electrical fast transients (EFT) immunity
- EN 61000-4-5:1995** Surge transient immunity
- EN 61000-4-6:1996** Conducted Radio-Frequency disturbances immunity

Other standard(s) are found not applicable for this apparatus

The undersigned declares that the tested specimen of the described electrical apparatus meets the essential requirements of the **EMC Directive 89/336/EEC**. This Certificate is granted on account of tests made in the laboratory of **KEMA Nederland B.V.**
This Certificate does not include an assessment of the manufacturer's production.

KEMA Registered Quality B.V.
(Competent Body EMC)
Arnhem, June 29th 1998


H.T. Jobker

Certificate No. **81714-KRQ/EMC 98-4423**

*Publication or reproduction of this document in total is allowed.

KEMA Registered Quality B.V.
Utrechtseweg 310, 6812 AR Arnhem, P.O. Box 9035, 6800 ET Arnhem, The Netherlands.
Telephone +31 26 3 56 34 17, Telefax +31 26 3 51 01 78.



**IEC SYSTEM FOR
CONFORMITY TESTING
TO STANDARDS FOR
SAFETY OF
ELECTRICAL
EQUIPMENT (IECEE) CB
SCHEME**

Ref. Certificate
No.

NL-7043

CB TEST CERTIFICATE

Issued by: KEMA Quality B.V.

Registration date: 2003-09-11

Product: Precision step motor controllers/drivers

Applicant: Advanced Micro Systems 2 Townsend West,
Nashua NH,
03053, United States
Of America

Manufacturer: Advanced Micro Systems 2 Townsend West,
Nashua NH,
03053, United States
Of America

Factory: Advanced Micro Systems 2 Townsend West,
Nashua NH,
03053, United States
Of America

Rating and principal characteristics: CDR-4MPS, CMAX-410:240 V, 0,8 A, 90 VA
CDR-8MPS, CMAX-810:240 V, 1,6 A, 180 VA

Trade mark (if any): AMS

Model/Type reference: CDR-4MPS, CDR-8MPS, CMAX-410, CMAX-810

Additional information:

Sample of product tested to be in conformity with IEC: 61010-1(ed.2)

Test Report Ref. No: 2029551.50 (66 pages)

This CB Test Certificate is issued by the National Certification Body

KEMA Quality B.V.
Utrechtseweg 310
P.O. Box 5185
6802 ED Arnhem
The Netherlands

Signed by: T.J.M. Gaertner

Date of issue: 2003-09-12

KEMA

PRODUCT OVERVIEW

Congratulations! 1
Features 2
Required Hardware..... 3
Front Panel Description..... 3
Fault Protection..... 4
Serial Adapters 4
“EASI” Software 5

INSTALLATION NOTES

Guidelines For A Reliable Installation..... 7
Typical Wiring Diagram 8
Design Tips 8
Troubleshooting Guide..... 9

GETTING STARTED

Hardware Set-Up 11
Software Set-Up Using the AMS “EASI” Program 11
Immediate Command Example 12
Program Command Example 12

AUXILIARY I.O.

Pin Description..... 15
Inputs..... 15
Outputs..... 16
Supply/Threshold/Shielding 17
Typical User Connection 17

SERIAL COMMUNICATION

Serial Command, Single Mode..... 19
RS-422 and Party Line 19
Party Line Hardware..... 19
Party Line Start-Up..... 20
Program Considerations..... 21
Communication Flow Chart..... 22
ASCII Baud Rates..... 22

OPERATING PARAMETERS

Reset, Initialize and Default Parameters..... 23
Input Command Example..... 23
Command/Program Mode..... 24
Party Line Commands..... 24
Interrupt Commands 24
Instruction Execution 24
Editing Programs..... 25
Result Data..... 26
Soft Stop “@”..... 26
Soft Stop Interrupt..... 26
Jog Speeds, Homing 26
High Speed Considerations 26
Related Items: Trip and Loop..... 26
Non-Volatile Memory..... 27
Scratch Pad/Shadow Memory..... 27
Watchdog Functions..... 28

ENCODER FEEDBACK

Encoder Hardware	29
Encoder Schematic	29
Design Notes	30
Encoder Calibration	30
Encoder Operation.....	30
Indexing With Encoders	31
Encoder Indicators	32

PROGRAM COMMANDS

Program Command Description	33
(ESC) Global Abort	33
(@) Soft Stop.....	34
(^C) Reset	34
(A) Port Read/Write.....	35
(B) Set Jog Speeds	36
(C) Clear and Restore	36
(c) lower case c; Reset Driver.....	37
(D) Divide Resolution	37
(E) Settling Time Delay.....	38
(F) Find Home.....	38
(G) Go.....	39
(H) Resolution Mode	40
(I) Initial Velocity.....	41
(i) lower case I; Restart Special Trip.....	42
(J/j) Jump.....	42
(K) Ramp Slope.....	42
(k) lower case K; Trip Output Value.....	43
(L) Loop on Port	44
(l) lower case L; Limit Polarity.....	45
(M) Move at Constant Velocity.....	45
(N) Force Step Position.....	46
(O) Set Origin.....	46
(P) Program Mode	47
(Q) Query Program	48
(q) Query Program (20 Lines).....	48
(R) Index Relative to Origin.....	49
(S) Store Parameters	50
(T) Trip Point.....	51
(u) lower case u; Control Line Feed	52
(V) Slew Velocity.....	52
(W) Wait.....	52
(w) Driver Status	53
(X/x) Examine Parameters.....	54
(Y) Hold and Run Current	55
(Z) Read Position	55
(I) Read Non-Volatile Memory	56
(J) Read Limits/Hardware/Watchdog Timer	56
(+) Index in Plus Direction	57
(-) Index in Minus Direction	57
(^) Read Moving Status	58
(\l) Write to Non-Volatile Memory	58
(\l) Selective Termination.....	59
^N (Name Axis).....	59
^P (Party Line Mode).....	60

ENCODER COMMANDS

(a) Read Stall Count 61
 (d) Set Deadband and Enable..... 61
 (e) Enable Encoder..... 62
 (f) Find Encoder Mark 62
 (g) Hunt Mode 63
 (h) Hunt Resolution..... 63
 (m) Index Mode 64
 (n) Force Encoder Position..... 64
 (o) Set Origin to Zero 65
 (r) Set Stall Retry Count 65
 (s) Stall Detect..... 66
 (t) Stall Sample Rate 67
 (v) Hunt Velocity..... 67
 (z) Read Encoder Position..... 67

SPECIFICATIONS

Electrical 69
 Environmental 70
 Physical 70

ADDENDUM

Command Summary 71
 ASCII Character Code..... 72
 Firmware Revision History 73
 Serial Interface Connections..... 75
 SIN-8 Schematic..... 75
 About Step Motor Current..... 76

Congratulations!

.....on your purchase of a CMAX-410 and/or CMAX-810 Stepper Motor Control System with "VRMC®" (Variable Resolution Microstep Control). Your CMAX system will provide years of reliable, accurate and cost-effective motion control. As with all AMS products, the CMAX is backed by nearly three decades of manufacturing excellence and a commitment to quality and support that guarantees your satisfaction.

This User Guide will assist you in optimizing the performance of your CMAX system. Its purpose is to provide access to information that will facilitate a reliable and trouble-free installation. The User Guide is organized into the following sections:

- Product Overview
- Installation Notes
- Getting Started
- Auxiliary I.O.
- Serial Communication
- Operating Parameters
- Encoder Feedback
- Program Commands
- Encoder Commands
- Specifications
- Addendum

We recommend that each section be reviewed prior to installation.

In addition to this User Guide, systems integration software (EASI) is supplied. This programming tool is a DOS based, menu driven utility file with on-line help screens and available source code (Microsoft "C"). Used together, the User Guide and the EASI software will enable you to quickly take advantage of the advanced programming features and system capabilities inherent in the CMAX design.

Although the CMAX systems 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 questions. Please feel free to take advantage of our technical expertise in this area by calling one of our support personnel to discuss your application- (603)-882-1447.

Thank You!
Your AMS Team

Features

The CMAX stepper motor control systems are microstepping drives with built-in indexer and power supply capabilities. They are designed to operate two-phase, permanent magnet stepper motors. Product configurations include the CMAX-410- 4 amps/40 volts, for general-purpose automation requirements and the CMAX-810- 8 amps/64 volts, (80 and 160 volt configurations also available) for heavy-duty industrial applications. A unique programmable current feature provides performance options that allow operation with the full line of stepper motor sizes. For export, these units are certified to meet EMC and Low-voltage Safety Directives.

The CMAX systems incorporate AMS' proprietary Variable Resolution Microstep Control (VRMC[®]) indexing technology with high output, bi-polar chopper drivers and state-of-the-art toroidal power supplies for maximum performance and reliability. Packaged in compact, heat-sinked and fan cooled enclosures, each unit provides serial communication, 2k bytes of non-volatile memory, extended I/O and a convenient AC power entry module to facilitate a quick and simple installation. Encoder feedback is also available for closed loop operation.

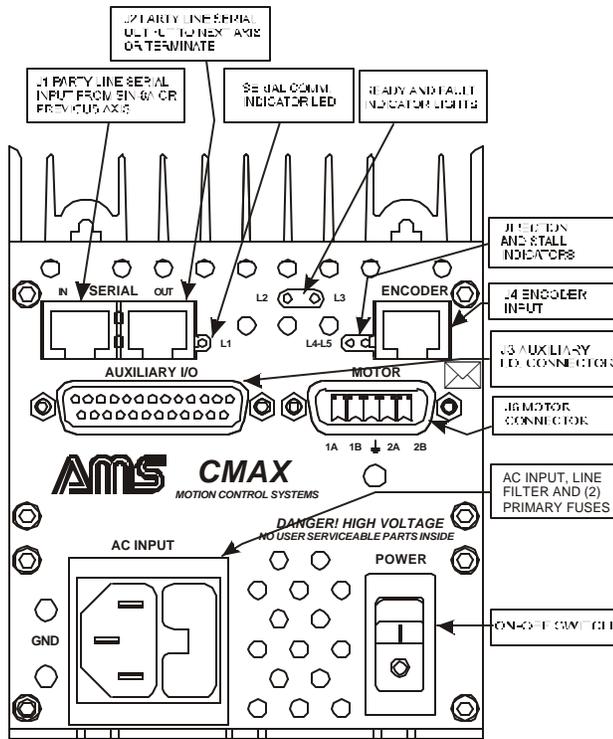
Features

- *Integral indexer/driver/power supply*
- *CE certified*
- *Output currents of 4A/40V to 8A/64V (non CE certified- 80 and 160 volts available)*
- *1/256 (fixed or variable) motor step resolution*
- *Short circuit, over temp, and under voltage protected*
- *Special "Watchdog" software for improved reliability*
- *Programmable run/hold current*
- *RS-422 "Party Line" operation*
- *Programmable acceleration and deceleration ramps*
- *2K bytes of non-volatile memory*
- *Four buffered/filtered user definable input ports*
- *Home/limit inputs*
- *Dual speed jog inputs*
- *Go and soft stop inputs*
- *Programmable trip point*
- *Encoder feedback option*
- *Free development software*
- *Full 2 year warranty*

Required Hardware

Qty	Unit	Model #	Description
1	Single Axis	CMAX-410/810	One axis step motor controller.
1	Axis	SIN-7 or SIN-9	RS-232, 25 pin or 9 pin serial adapter. Single axis only.
1	System	SIN-8	RS-232/RS-422 serial adapter and cable.
1	System	SIN-10/SIN-11	Intelligent serial adapter and cable.
1	Axis	BLC-44	Motor connector kit (included when purchased with motor)
1	Model	BLC-40	AC power cord (included).
1	Added axis	BLC-51-3	Interconnect cable, Cat5 (3 ft.).

Front Panel Description



Status Indicator LED'S

#	Label	Description
1	L1 (Green)	"On" when serial communications has been established.
2	L2 (Green)	Ready: (See Fault Protection/Operating Parameters section)
3	L3 (Red)	Fault: (See Fault Protection/Operating Parameters section)
4	L4 (Yellow)	Direction: "ON" when indexing in the "-" direction.
5	L5 (Red)	Stall: "ON" when motor position differs from encoder position dead-band or when in "low seek" mode.

Fault Protection

The CMAX is internally protected against phase to phase and phase to ground short circuits, over temperature and under voltage. Two LEDs are provided to indicate operating conditions and status.

The CMAX is packaged in a specially designed heatsink with cooling fan to help avoid over temperature conditions. Should an over temperature condition occur however (between 55° C and 65° C), the driver will automatically shut down. If the DC voltage to the driver drops below the minimum specification, the driver output stage will be disabled. For added safety, the driver outputs will not automatically re-enable when the proper voltage and/or temperature condition is restored but rather requires the driver to be reset.

The short circuit protection consists of phase-to-phase, phase to ground, and +V to phase. If a phase short to ground fault is detected, the outputs will be disabled and cannot be re-enabled without resetting or powering down the driver.

Diagnostic LED's

Two LEDs (one Green and one Red) are provided to indicate operating conditions and status as follows:

1. Normal Power Up (reset): The Green LED will blink 3 or 4 times then remain on. If the flashing does not occur when power is applied, something is malfunctioning.
2. During Stepping: The Green LED will be off when the motor is not on a full phase and on when it is on a full phase. The Red LED should normally be off.

Faults

A fault condition always involves the Red LED:

Condition	L2 Green LED	L3 Red LED
Over temperature*	Blink	On
Short outputs*	Off	Blink
Under voltage*	Off	On

* Latched; requires power cycle reset to clear.

A host computer may interrogate and signal the faults via the “w” command.

Serial Adapters

All CMAX series products communicate with a host computer or dumb terminal using a serial interface. AMS offers several serial adapter and cable assemblies suitable for a variety of applications, as follows:

Single Axis Operation

SIN-7 and SIN-9 Passive Adapters

The SIN-7 adapts RJ45 to DB-25 serial port and the SIN-9 adapts RJ45 to DB-9 serial port. They are wired directly through with RS-232 levels passing to the appropriate RJ-45 pins. These devices will only interface to one controller and cannot be used in multi-axis applications.

Multi-Axis Operation

SIN-8 RS-232 to RS-422

This adapter contains RS-232 transmitter/receivers and RS-485 transmitter/receivers and can be used with party line network designs. Any number between one and thirty-two axes can be connected in a mini-drop network system using the SIN-8.

SIN-10 and SIN-11 Intelligent Serial Adapters

The SIN-10 and SIN-11 are **intelligent** serial line converters (RS-232 to RS-422) with a built-in microprocessor that eliminates the need for special echoed character software. It is recommended for Windows applications where either the machine or software is slow and/or the operating system prevents direct programming of input or output instructions.

The SIN-10 has selectable baud rate speeds (up to 470k baud), while the lower cost SIN-11 is fixed at 9600 baud. Specific operating instructions are contained in the SIN-10 and SIN-11 User Guides.

“EASI” Software

EASI is a DOS program tailored to operate with AMS motion control products. It is not intended to operate as an end user application program, but rather to allow familiarization and evaluation of the products performance. In single axis mode, it is not necessary to use EASI, as any dumb terminal device will work.

Although EASI will operate from a floppy disk drive or CD, copying the contents to a sub directory (call it “easi”) will increase the speed and performance.

Guidelines For A Reliable Installation

Designed for Maximum Immunity

The CMAX-410 and CMAX-810 stepper motor control systems are designed to meet, and have been tested to, European (CE) standards for EMI emissions and immunity to RFI, ESD, and power surges. For added protection all CMAX units include robust noise tolerance software that monitors for brownouts, plus an independent watchdog timer that constantly supervises and prevents runaway CPU behavior.

Extensive power line filtering and surge suppression are designed into the CMAX systems. I/O signals pass through a ferrite “D” connector and RC filter before the isolation of a comparator with logic hysteresis. Most inputs can tolerate in excess of 40 volts. Chopped motor drive outputs are also EMI suppressed.

Chopping and Electromagnetic Interference (EMI)

While all AMS stepper motor control products use the latest, state-of-the-art technology, proper installation and wiring practice is still essential to ensure trouble free, reliable motion control. This section is provided to help system designers avoid problems with electromagnetic interference (EMI/electrical noise) emanating from the advanced technology itself, known as “chopping,” and other common sources of electrical noise.

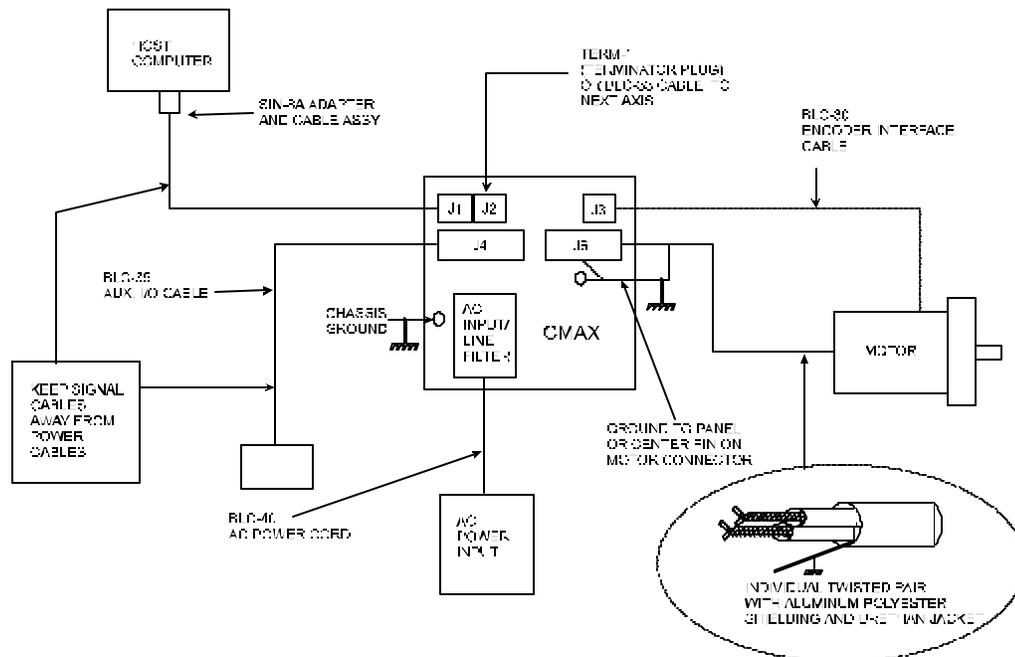
The motor driver technology uses a technique called “chopping.” Compared to older drive techniques, chopping gives improved motor performance while allowing the drive circuitry to dissipate less power. With this design, the voltage applied to the motor windings is turned on and off very rapidly, or chopped. The voltage level and chopping frequency are precisely controlled so that the desired current is produced. The instantaneous current in the drive circuit is sensed and is used to control the current to the motor.

High speed chopping, at rates above 15kHz, and nanosecond switching times will produce EMI that may cause the microprocessor logic in the system to fail. The system design engineer should be aware of the interaction between these conflicting technologies. Failing to do so can result in an intermittent and unreliable product design.

Other Common Sources of EMI

In addition to the noise generated by chopping, EMI is typically generated through ground loops and AC power line disturbances. External devices such as relays, coils, solenoids, arc welders, motors, and other computer-based equipment are also common sources of EMI. It is essential that these devices be isolated, as much as possible, from the CMAX.

Typical Wiring Diagram



Design Tips

The following design tips will help to prevent EMI from interfering with the system operation:

- Mount the CMAX enclosure as far away from noise sources as possible.
- Make sure that all power wiring (motor, AC, etc.) is away from the signal wiring (I/O, communications lines, etc.).
- Mechanical grounds should all be tied to Earth at a single point. Chassis and motor grounds should be tied to the frame and the frame to Earth.
- Ground all signal wiring to one point.
- Use solid state relays or opto isolators whenever possible to isolate remote signals. Suppress all mechanical relays with capacitors or MOV's.
- Use shielded, twisted pair cables for the motor, I/O and communications wiring.

Note: The decision to ground at one end (the controller/driver) or both ends is dependent on the possibility of ground loops. Proper grounding of shields is best accomplished empirically. The length of any ground drain wire should be as short and direct as practical.

Troubleshooting Guide

Diagnosing and correcting problems in today's systems can consume a major portion of the design time. Shortcuts for the sake of economics in the early phases of development will be lost in future troubleshooting expenses. In determining the cause or causes of a malfunction there is a need to analyze and consider the symptoms. The mixture of mechanics, electronics, and software allows the blame to be equally passed around. Often the problem originates from a complex interaction between all of these elements.

Evaluate

How often does the failure occur? The frequent failure can usually be located easier and quicker than one that occurs infrequently. Does a relay or motor switch on coincidentally with the failure?

Does the same thing happen each time? A software Bug tends to be repeatable. Random symptoms indicate a noise event, although improper software handshake or timing can appear random in nature. Electrical interference problems are common with today's computer based controls, and such problems are often difficult to diagnose and cure.

If a problem occurs with your system, it is recommended that the following checks be made to help locate the cause:

1. Check the AC line for fluctuations in line voltage. If needed, add a line filter or isolation transformer.
2. Re-inspect all mounting, grounding, and wiring for proper installation and connection.
3. Disconnect all sources of EMI from the CMAX and try operating in a stand-alone mode. If the problem goes away, it is most likely caused by one or more of the noise sources and/or a poor connection. Try connecting the noise sources one at a time to isolate the problem. When the problem is identified, inspect for good connections and adherence to previously stated design tips.

Remember; it is always best to suppress noise at the source, rather than attempting to prevent its entrance into your electronics.

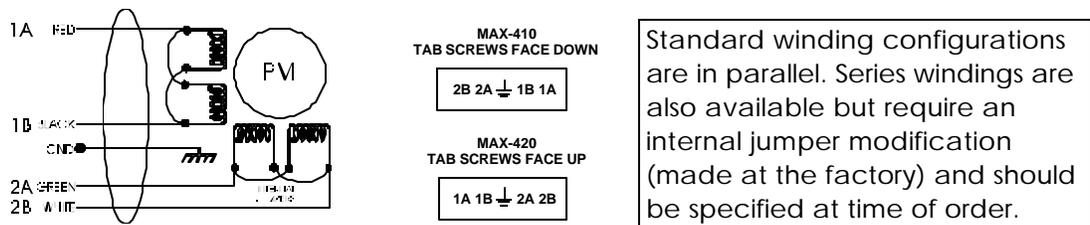
Note: Wait at least two minutes after turning off the power before connecting or disconnecting the motor. This will allow proper dissipation of voltage from the unit. Failure to do so may cause damage and void the warranty.

Hardware Set-Up

The following procedure is intended to familiarize you with the basic hardware and software set-up for single axis (one motor) operation. Information on multi-axis (Party Line) operation is provided in subsequent chapters. Information on serial interface connections (pin-outs) is contained in the Addendum.

- Step 1** Attach the selected serial adapter to either COM1 or COM2 of your PC/AT compatible computer. If you are using a SIN-8, make sure that the RED switch on the side of the connector is in the single (S) line mode. If you are using a SIN-10, make sure all switches are off. (The SIN-11 has no switches).
- Step 2** Connect the other end of the cable assembly (looks like a telephone connector), to the mating connector J1 “Serial Input”, on the front panel of the CMAX.
- Step 3** Install a terminator plug (TERM-2) into J2 “Serial Output” of the last axis. The controller will work without a terminator plug, however reliability may be compromised based on the environment (noise) and length of the interconnect cables.
- Step 4** Connect a motor to J6. (Refer to “About Step Motor Current” in the Addendum for more detail on setting the proper motor current).

Typical Wiring Diagram for AM23 and AM34 Series Motors



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.

- Step 5** With the Power Switch off (down), connect the AC power cord to the CMAX and to a correct AC power source (120V 50/60 Hz or 240V 50/60 Hz).
- Step 6** Turn the Power Switch on. The following actions should occur:
 1. The fan on the rear panel should start rotating.
 2. The READY indicator light will blink 3 to 4 times and may or may not remain on, depending on the motor phase.
 3. The ON-LINE indicator light (labeled OK) should be off.

Software Set-Up Using the AMS “EASI” Program

- Step 7** Install the diskette or CD and type: EASI <CR> at the prompt. At the opening screen enter “n” if you are operating from a monochrome terminal or hit the ENTER key for color.
- Step 8** Use the arrow keys to select COM1 or COM2. COM1 is the default setting. Follow the same procedure to select the correct baud rate. 9600 BPS is the default setting.
- Step 9** **Sign-On**
Select the “DUMB TERMINAL” mode. A blank window will appear.

Step 10 Strike the SPACE BAR key. The controller should sign on as follows:

```
xxxx   yyyy   ADVANCED MICRO SYSTEMS, INC.  
MAX-2001 v.z.z.z
```

Where:

xxxx is the encoder software date code,
yyyy is the primary controller software date code and
v.z.z.z is the software version number.

Note: The first four-digit number will only appear if the CMAX is configured with an encoder option.

Striking the ENTER (Carriage Return) key should result in an echo of “# “ characters, further indicating communication is established.

After sign-on, the READY indicator light (labeled “OK”) should come on, indicating that the characters were received. If the sign-on was not successful, reset the controller by entering a ^C (Control C), or cycling power off and on, and repeat the above procedure.

You are now ready operate in single axis mode.

Immediate Command Example

Immediate Commands are activated directly through the serial interface and not stored in non-volatile memory as a program.

1. Enter the command: R 1000 <CR>. The motor shaft should move 1000 steps in the CW (clockwise) direction, “Relative” to the origin.
2. Enter the Z <CR> command. The position (1000) should be displayed on your monitor.

More information regarding the “R” and “Z” commands is provided in the Program Command section.

Note: The motor shaft is turning based on a set of factory “default” parameters that are stored in the non-volatile memory of the CMAX. Information on how to change these parameters to suit your particular application is available in subsequent chapters.

Note: the CMAX commands are upper/lower case sensitive. Upper case is typically used for motion commands and lower case for encoder commands.

Program Command Example

The following is a sample of sequences that are stored in non-volatile memory. When in the Program Mode, the sequence is immediately written to non-volatile memory, without any additional action required to save it.

Some Rules

1. The command line may be edited using a backspace as characters are typed.
2. The line may be canceled using <ESC>.
3. The command line is limited to 12 characters.
4. Only one command may be entered per line.
5. A space is optional between the command and first number.
6. A space or comma must be used to separate two parameter commands.

Step 1 Enter the program

	<u>Enter</u>	<u>Remark</u>
	P0<CR>	Place in Program mode. Insert instructions at location 00.
<u>Address</u>		
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	W00<CR>	Wait until complete.
17	J1 3<CR>	Jump to address 1, 4 times (3+1).
21	R500<CR>	Move 500 steps in the "+" direction, relative to Origin.
26	P0<CR>	End Program.

Step 2 List the stored program

<u>Enter</u>	<u>Remark</u>
Q<CR>	Query command.

Note: An upper case "Q" displays one line at a time. A lower case "q" will display up to 25 lines at a time.

Step 3 Verify the Program

The CMAX will respond with:

0	O	
1	R	10000.00
6	W	0
9	R	-10000.00
14	W	0
17	J	1 3
21	R	500.00
26		

Step 4 Execute the Program

<u>Enter</u>	<u>Remark</u>
G0 1<CR>	The CMAX starts executing the program starting at location zero. If the Trace option is on (see "G" command in the program Command section), the CMAX will display each instruction, prior to execution.

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

Step 5 Edit Program

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

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

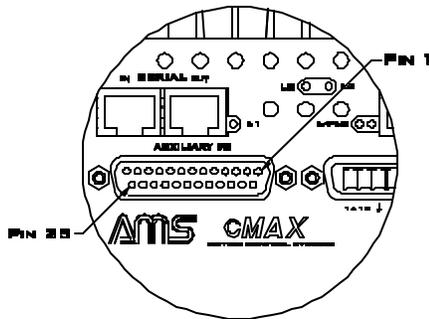
Note: Caution should be exercised when making Program Edits in dumb terminal mode due to variations in Command byte length that may effect subsequent command address locations and possible corruption of non-volatile memory storage. It is recommended that application programs be developed using the menu driven program (Party Line selection) in the EASI software, which includes a sophisticated Editor and Compiler

Pin Description

The 25 Pin DB style female auxiliary I.O. connector of the CMAX provides EMI/RFI shielding and filtering, noise filtering, and hysteresis for slowly changing inputs. It supports 4 general-purpose inputs, 5 general-purpose outputs, 8 dedicated inputs, 1 dedicated output, and flexible user support facilities.

Ports 2-4 are bi-directional inputs or outputs, and specific outputs reflect the state of specific inputs (P2 IN — P2 OUT; P3 IN — P3 OUT; etc), if the specific output is not turned-on by software. This can be useful to buffer a low-level input from another device, such as a PLC, through the CMAX, *as well as* using it as an input to the CMAX.

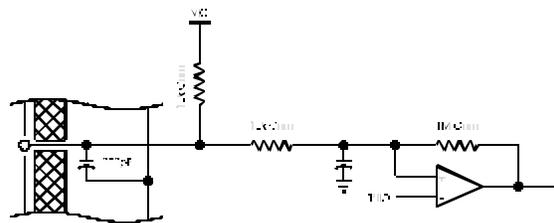
I.O. Orientation



Pin	Name	Function
1	P2 IN	Port 2 Input
2	P4 OUT	Port 4 Out
3	MVG_OD	Moving Open Drain
4	P5 OUT	Port 5 Out
5	P3 IN	Port 3 Input
6	P1 IN	Port 1 Input
7	JOGHS	High-Speed Jog Input
8	GND	Ground
9	GO	Go Input
10	V_OUT	Output Voltage Clamp
11	P2 OUT	Port 2 Output
12	D/U	Do Not Use
13	P3 OUT	Port 3 Output
14	VCC	On-board Logic Supply
15	HOME	Home Input
16	LIMA	CW Limit Input
17	LIMB	CCW Limit Input
18	JOG1	CW Jog Input
19	JOG2	CCW Jog Input
20	P6 OUT	Port 6 Out
21	SS	Soft Stop Input
22	P4 IN	Port 4 Input
23	VIO	Input Voltage Supply
24	GND	Ground
25	GND	Ground

Inputs

All inputs consist of a comparator circuit, which provides robust noise immunity; and supports a wide input voltage range. A typical input is shown in here. An internal 10k Ohm pull-up resistor to VIO is provided to force inputs normally high, and need to be pulled-low by the user to activate. An input R-C filter provides about a 10 kHz cut-off, and the comparator is configured for about 1% hysteresis.



General Purpose Inputs

P1-4 IN, The General Purpose Inputs provide 4 user programmable inputs supporting conditional input looping with the Loop On Port, or 'L' command; and conditional branching using the Vectors Go, or 'G2048' command. Port 2-4 are bi-directional I/O in that the port can be used as an input if the output bit is turned off. Inputs status can be read using the Read Port Command, 'A129'.

The Loop On Port command is useful for waiting for an input to become active before proceeding with a program. The vectored GO command is especially useful for selecting various programs with an external thumbwheel switch, or other logic devices, which can select various input, bit combinations. P2-4 are bi-directional I/O, in that it can serve as either an input, if the output is turned-off; or as an output, if the input is not needed. P2-4 OUT, further reflect the state of P2-4 IN, if programmed as an input.

Dedicated Inputs

The dedicated inputs: LIMA, LIMB, HOME, JOG1, JOG2, JOGHS, GO and SS support motion specific functions and cannot be programmed for other purposes.

LIMA and **LIMB**, the Clockwise Limit, and the Counter-Clockwise Limit, respectively, are used as an over-travel input to prevent equipment damage or personnel hazard from traveling beyond the bounds of the work envelope through a programming, or operational error.

The **HOME** input is used to reference the axis to a specific location and can be an independent switch, or it can be connected to one of the limits. Home is only examined in conjunction with the Find Home, or 'Fxxx' command.

JOG1 and **JOG2**, cause the motor to move, or jog clockwise, and counter-clockwise, respectively; both at a speed programmed by the Set Jog Speed command, 'Bxx xx'.

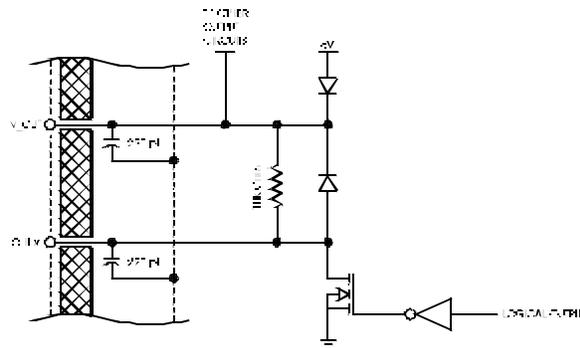
JOGHS causes the JOG1 or JOG2 inputs to operate in the high-speed mode, as programmed by the Set Jog Speed command, 'Bxx xx'.

The **GO** input causes program execution to begin at program address location 0. This function is useful for stand-alone operation, when a program is stored in memory.

The **SS**, or Soft Stop input. If executing a program, activating this input causes program execution to stop. If executing a move, activating this input causes the move to ramp to a stop. Since activity termination is not immediate, and because this functions under software control, **this input should not be used in place of an emergency stop.**

Outputs

All outputs consist of a 1 Amp/100 Volt MOSFET, with a 10k Ohm pull-up resistor to VIO provided to force off-state outputs to the high-state. An inductive clamp diode is provided to V_OUT to shunt flyback currents away from the MOSFET. To take advantage of these diodes, the user must connect the I.O. supply voltage to the V_OUT terminal.



General Purpose Outputs

P2-6 OUT, the general-purpose outputs provide 5 user programmable outputs. These outputs can be programmed using the Port Command ('A xxx'). P2-4 are bi-directional I.O., in that they can serve as either an input, if the output is turned-off; or as an output, if the input is not needed. P2-4 OUT, further reflect the state of P2-4 IN, if programmed as an input. On-the-fly trip output functions are supported using the Special Trip Command 'k xxx xxx'. P2-4 are bi-directional I/O, in that it can serve as either an input if the output is turned-off; or as an output if the input is not needed. P2-4 OUT, further reflect the state of P2-4 IN, if programmed as an input.

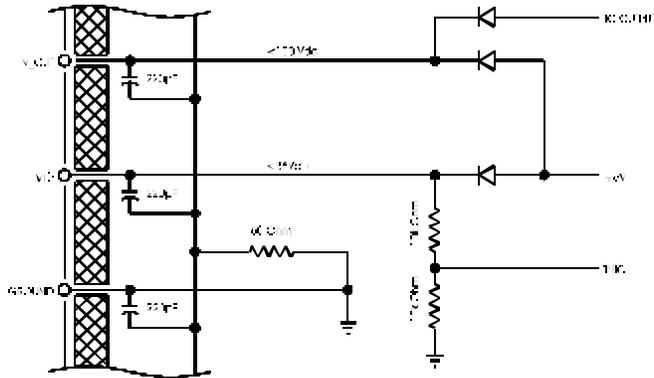
Note: Software version 1.28 and above allow ports 5 and 6 to be used for step and direction output to another driver. Reference "H2" command.

Dedicated Output

MVG_OD, or MoVinG Open Drain, is a dedicated output, which is active whenever the controller is stepping. This output is useful as a 'Moving Status' bit.

Supply/Threshold/Shielding

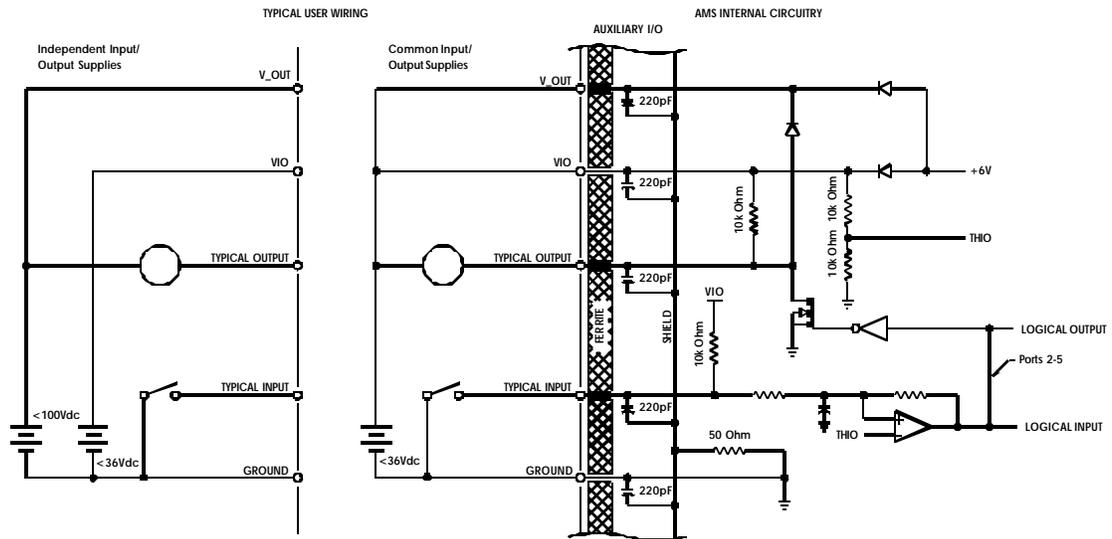
The I.O. power distribution provides considerable flexibility of connections and Voltages to loads and input devices. As seen, there is an on-board 6 Volt bias (about 5¼ Volts after the diode) supplying both the VIO and V_OUT terminals through diodes. This is suitable for low Voltage applications, and is 'Diode Or'ed with the VIO and V_OUT pins, if a higher voltage is required. The inputs can tolerate up to 36 Vdc, and the outputs can be wired for up to 100 Vdc. If the inputs and outputs are to operate with different voltages, then the output voltage supply should be connected to V_OUT, and the input voltage supply should be connected to VIO.



The input logic thresholds (THIO) are established using a voltage divider of 10k Ohm resistors. These are setup to provide a 50% switching threshold regardless of the input voltage.

The I.O. connector is fully EMI/RFI shielded and filtered by a ferrite block and 220pF shunt capacitors. The shield is connected to Ground through a 50 ohm resistor, to minimize the effects of ground loops.

Typical User Connection



Serial Command, Single Mode

The words “host” and “terminal” are used interchangeably and refer to any source of commonly available standard RS-422 interface including; terminals, and UARTs (COM ports). A character is a stream or packet of 8 consecutive bits (1’s and 0’s) with a defined sequence. Each unique character represents a letter, number, punctuation or unreadable control character as defined by an international set of standards called ASCII. The speed that characters are sent or received is defined by the Baud rate. The serial communication link refers to any two-wire (RS-232) or four wire (RS-422), full or half duplex communication link.

Command lines consist of an ASCII character followed by 0, 1 or 2 decimal ASCII numbers depending on command requirements. The User may edit the line prior to entry by using either the BACKSPACE or DELETE key. The command line may be up to 12 characters long, including spaces. Spaces are optional between the command character and first number. A motion command with two functions requires at least one space between the values; i.e., J100 9 (jump to location 100 and repeat sequence 10 times). Most motion command characters are issued as UPPER CASE, while all encoder command characters are lower case.

In the Command mode, the command is executed upon receipt of a <CR> (Single mode) or ^J, ^Enter (Party Line mode). The CMAX will respond with a <CR> and Line Feed upon acceptance of the command.

RS-422 and Party Line

Communication is via RS-422 protocol and may operate in either Single or Party Line mode. Each CMAX receiver monitors the host and responds when receiving a matching Name character. The Single mode provides user friendly, one axis communication for set-up and debug functions. Set-up usually involves optimizing operational parameters, writing and storing programs and assigning a unique Name for Party Line operation.

Party Line Hardware

“Party Line” products have a 4 wire RS-422/485 interface. Differential line drivers and receivers are used to provide reliable communication in noisy environments. This design allows a single Master (or Host) computer and up to 32 Slave controllers. The hybrid design retains all the desirable characteristics of both EIA RS-422 and RS-485 specifications. In general, the hardware implementation follows the extended RS-485 standard with higher voltage and receiver capacity. Rather than half duplex protocol however, a full duplex Party Line communication is provided.

Slaves

Each Slave unit contains the following:

1. RS-485 line receiver; always active.
2. RS-485 tri-state line driver; activated on receipt of “address received.”

The line driver is always enabled when Single mode operation is selected.

Master

The Master interface, consists of:

1. RS-485 line receiver; always active.
2. RS-485 line driver; always active.

Note: The RS-485/422 is rated for a maximum cable length of 4,000 feet. It is recommended that a second terminator (120 ohms) be used if runs exceed 15 feet or if operated in an electrically noisy environment. DO NOT bundle signal wires and motor wires together. The high current and frequencies generated by chopper drivers will couple, even if shielded wire is used (unless 100% shielding can be guaranteed). Avoid proximity to relays, motors and other RFI sources.

Party Line Start-Up

Note: The following information assumes the use of a SIN-8 serial adapter. Detail information on the use of a SIN-10 or SIN-11 serial adapter is available from AMS.

Step 1 Name Axis

Each axis must be assigned a unique “name” for proper operation when used in Party-Line mode. Reset the controller (^C), then type a single, valid name character.

Note: Software version 1.31 and above include a ^N command for an alternate method of assigning a name(s). (See ^N in Program Command section for more information).

Note: Make sure that the RED switch on the SIN-8 connector is in the single (S) line mode.

Recommended Names: Upper case A through Z Lower case a through z		Non-valid Names	
ASCII	HEX	ASCII	HEX
[5B	^C	03
\	5C	CR	0D
]	5D	LF	0A
^	5E	@	40
-	5F		
'	60		

Step 2 Sign On

Follow the name with a SPACE BAR. The sign-on message will appear.

Step 3 Verify the Name

Verify the Name by entering the “X” command (UPPER CASE), followed by a <CR>. The last item of the first line should contain the “Name” character.

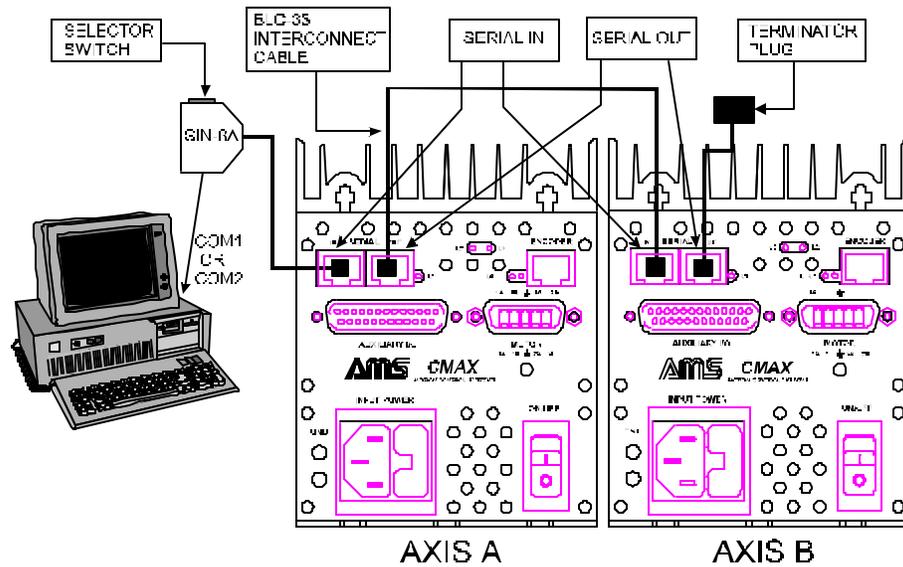
Step 4 Save the Name

Enter the Save command “S” (UPPER CASE) <CR>. The axis Name is now stored in non-volatile memory.

Once each unique Name has been assigned and stored in the non-volatile memory, multiple axes may be connected in parallel for Party Line mode operation. Single mode communication enables terminals or computers with serial communication capabilities to interface with the built-in command line editor. Party Line protocol requires the correct input character sequence to address an axis. It is recommended that a PC or other computer be used to simplify programming.

Step 5 Connect Axes

After assignment of a unique “Name(s),” connect the axis (1 to 32) as shown below. Move the red selector switch on the SIN-8 to Party Line (P) mode.



Step 6 Select Party Line Mode

Enter “Ctrl A” to return to the main menu of EASI and select “Party Line” <CR>. When starting in Party Line the CMAX reads its Name from the non-volatile memory and then skips any sign-on procedure. At this point the sleep condition (serial outputs tri-stated) exists, until a Name match occurs.

Note: Software version 1.31 and above include a ^P command for an alternate method of entering Party Line mode. (See ^P in Program Command section for more information).

Serial Protocol

Each Slave unit residing on the Party Line acts as a listener, waiting for its personal address (name) character. Once the proper name is received, the Slave enables the RS-485 driver onto the TX bus. The activated Slave then echoes the start character and receives and echoes the remainder of the command string until the terminator is received. The terminator must be a “line feed” (lf) (Ctrl Enter on most computers).

Once the line-feed character has been received, another axis may be commanded. In order for any axis to recognize a name, the name must be preceded with a line-feed (the previous terminator will do).

The echoed characters provide reliable handshaking and a means of receiving data from the Slave, i.e., a “Z” position register command.

Step 7 Enter a Simple Command

Party Line communication using a dumb terminal might go like this (assume two axis with address “X” and “Y”):

```
Master (Host PC): <lf> X+1000 <lf> Y-500 <lf> XZ <lf>
Slave (Axis):    X+1000 <lf> Y-500 <lf> XZ1000 <lf>
```

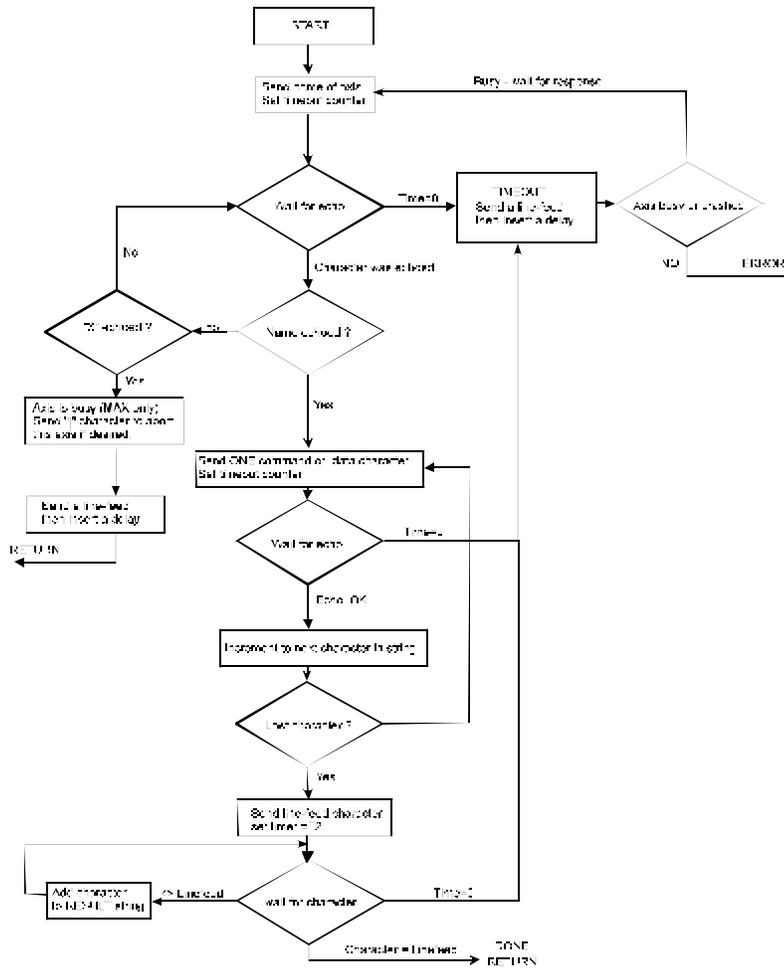
Program Considerations

The CMAX incorporates a buffered UART input. Because motion control is of the highest priority, processing of received information may be delayed if commands are sent while stepping at very fast rates. This condition may only occur at internal/external step rates exceeding 10,000 steps per second. In serial applications where commands are sent while motion is active, the User should monitor echoed data to avoid UART overrun.

Note: IT MUST BE STRESSED THAT ANY HOST COMPUTER MUST VERIFY THE CHARACTER ECHO AS EVERY SINGLE CHARACTER IS SENT. Inserting a delay between characters will inevitably fail and slow communications. Your application software must be capable of single character (as opposed to string) transmission without reception loss at 9600 Baud.

Use of the SIN-10 or SIN-11 serial adapter simplifies this process. User manuals for each are available from AMS.

Communication Flow Chart



ASCII Baud Rates

Baud rates for CMAX communication is fixed at 9600 BPS. Other options are available depending on the serial adapter being used and hardware configuration of the CMAX. Consult AMS for details.

Host/terminal settings:

Start Bits	Data Length	Stop Bits	Parity
1	8	1	None

Reset, Initialize and Default Parameters

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 BPS. Additionally, checks for Party Line operation are initiated.

When the on-board non-volatile memory is accessed, the parameters most recently stored by the “S” command are downloaded, replacing the standard defaults in the working registers of the controller. The Examine command (X) (UPPER CASE) <CR> causes the CMAX to display the parameter values that were last stored in non-volatile memory.

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)	3004 (steps per second)
Divide Factor (D)	1
Ramp Slope (K)	10/10
Jog Speeds (B)	3, 20
Trip Point (T)	Off
Step Resolution (H)	0 (fixed)
Auto Power Down	Yes
Hold/Run Current (Y)	5, 25
Limit Polarity	Low
Auto Position Readout (Z)	Off
Name (after reset)	Undefined
Encoder Option Only	
Encoder Resolution (e)	500 (ratio=10)
Deadband (d)	30
Hunt Velocity (y)	700/32
Stall Factor (s)	20
Stall Test Delta (t)	5
Set Stall Retry Count (r)	5 (lag=3)

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.

Input Command Example

Example #	Command	Single Axis Mode Operation
1	+1000 <CR>	Step 1000 steps in + direction
2	+ 1000 <CR>	Same as #1
3	H 0 <CR>	Set fixed step resolution
4	H0 <CR>	Same as #3
5	H <CR>	Same as #3 and 4 (0 is used by default)
6	R -1000 <CR>	Move to position -1000, relative to Origin

Commands such as Jump and Loop instructions are only valid when used in the Program, then Execute mode. The following sequence can only be executed from programs stored in non-volatile memory:

J 0 5 <CR>: Jump to location 0, 6 (n + 1) times.
 J0 5 <CR>, same as above.

Command/Program Mode

In the Command mode, commands are normally executed as soon as they are entered. The use of non-volatile memory also allows storage of these commands. These stored commands can be triggered at power-up for automatic or repetitive operations by initiating a command or by strobbing the Go (input J5, pin 12 or J3, pin 5) to a logic low, or auto starting at address 1600.

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

The following procedure assumes a standard (RS-422) serial interface using a common terminal:

1. The Program mode is initiated by entering "P aa" <CR>. 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" <CR> or by strobbing the "Go" input.
2. 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 the Immediate Command mode. Entering a "P" terminates the Program mode and the CMAX will return to the Immediate Command mode. Use the "S" (Save) command to retain the program in non-volatile memory.

Several programs may coexist 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 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 for the trip point.

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 concurrently monitor the common TXD line from the host. Once the Name is received, the target axis will wake-up and start echoing as described above.

Interrupt Commands

A program in process will be interrupted by a "Program Complete" (P), a Soft Stop command, or a hard stop on the I.O.

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.

Instruction Execution

For each Motion command there are four cycles:

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

Other commands have three cycles:

1. Entry
2. Execution
3. Result

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

Cycle 1. Entry

The input command and data are loaded via RS-422 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 <CR> (^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 <CR> and Line Feed. If the result does NOT produce numeric data then the <CR>, 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 CMAX has the capability to queue up another action command 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 (J3, pins 2, 3) soft stop input (J5, pin 13 and J3, pin 6) and home switch (J3 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.

Editing Programs

Existing program(s) may be modified at any time. The User can review the existing instructions by entering the “Q” command. This command produces a list of instructions along with the appropriate memory addresses. To edit an existing program enter “P”, along with the desired address, and proceed to enter the new instruction(s) as in the Program mode.

The edit session may be terminated in two ways. If the edit results in a program that is longer than the previous program or if the User wishes to discard the old instructions (shorten program), enter "P" to terminate edit and cause an end of program marker to be inserted. If only one or several successive new instructions are to be altered, entering ESCape will terminate the edit. Any instructions outside of the edit area will NOT be altered.

Note: If any instructions are of different byte lengths than existed previously, the program could wind up with invalid instructions in the middle of the program. Keeping track of the byte count will avoid this condition. The User may insert redundant or "dummy" one-byte instructions to fill the gap. If in doubt reenter the remaining portion of the program.

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.

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) and 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: A command should not be loaded while jogging and jogging should not be performed until the last command is complete.

High Speed Considerations

The CMAX is designed to control step rates 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.

Related Items: Trip and Loop

The Trip point output is activated on the exact step position specified. When running a program (from the GO command) several "fetches" from the non-volatile memory are required along with the service time. This latency may allow several motor steps to occur before the desired action takes place.

Loop on port may exhibit similar latency effects at high speeds. The port will require a longer "true" condition to be recognized. A faster method is to implement the "wait for port" condition using the "Go - Sub," (branch on port) condition.

Non-Volatile Memory

The CMAX hosts a 2048 byte non-volatile memory. 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 following 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.

Page	Address	HEX	Storage Allocation Type
1	0-255	0-7FF	Program, Trip, Go ¹
	128-192	80-C0	Program, k-Trip, Shadow
2	256-511	100-1FF	Program, Go ²
3	512-767	200-2FF	Program
4	768-1023	300-3FF	Program
5	1024-1279	400-4FF	Program
6	1280-1535	500-5FF	Program
7	1536-1791	600-6FF	Program
	1600		Program, Power-Up ³
8	1792-1893	700-765	Micro-position lookup table
	1894-2047	766-7FF	Operational parameter storage

- ¹ Aux. Input "Go" (hardware) activated.
- ² "Go Sub" branch on port condition.
- ³ Execute User instruction upon power-up at location 1600.

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 Non-Volatile Memory (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. The CMAX parameters are set so quickly that it is sufficient to just let the host download them.

Changing parameters should NOT be done by writing directly to the EEPROM. The CMAX 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.

Scratch Pad/Shadow Memory

Sixty-four (64) bytes of the program memory are configured as FAST "Shadow 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 within this area 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.

Watchdog Functions

The watchdog functions help to improve reliability in harsh environments (reference “]” command). Two enhancements are provided:

1. A watchdog timer
2. Power fail detection

Watchdog Timer

A special timer will interrupt and reset the processor. In order to avoid this timeout, the timer must be reloaded by software instructions. A microprocessor “crash” caused by noise or other failure will allow this timeout to occur. The error event will set a flag that may be read by the host computer.

Power Fail Interrupt

This feature will detect a logic power supply voltage that is out of tolerance. During power down or a severe power glitch the processor will be held at a reset condition until proper voltage (5v) is restored. This event is recorded by setting a flag, readable by the host computer.

Operation

Upon detection of a fault the watchdog flags are set and a CPU restart is initiated. All events that take place during power up are initiated, except reset of these flags.

1. Motion and running programs halt.
2. All position counters are reset.
3. Stored NV parameters are reloaded.
4. All ports set to off state.
5. The encoder controller is re-initialized.

In single axis mode a space character must be used to sign on. The sign-on message will contain the “!” character. In the party line mode the unit assumes the ready condition but will echo the “!” in response to the name address.

The host should issue the “[1” command to clear the watchdog flags and resume normal operation. Note, any command can be executed, even though the “!” is echoed. Once the flags are reset the assigned name will be used.

The watchdog flags (timeout and brown-out) are reset under only two methods:

1. By a hard reset such as at power up.
2. Using the “[1” command.

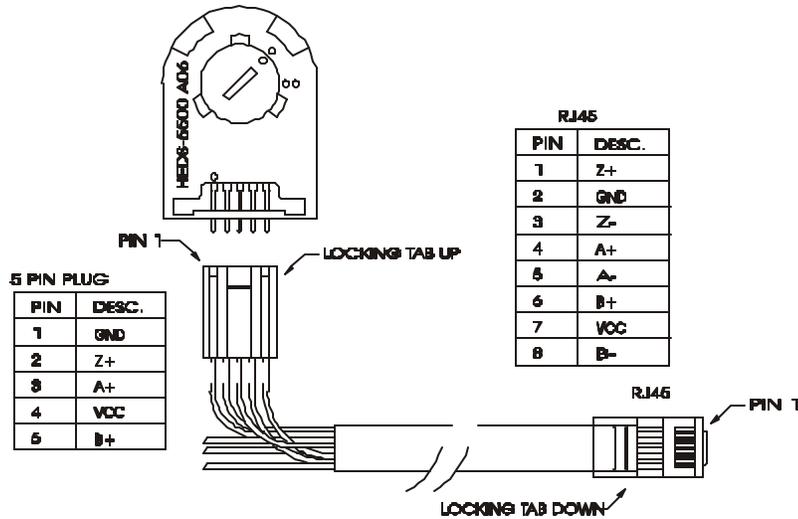
The host is made aware of the flags by echoing the “!” character in place of the name during party-line communications.

Encoder Hardware

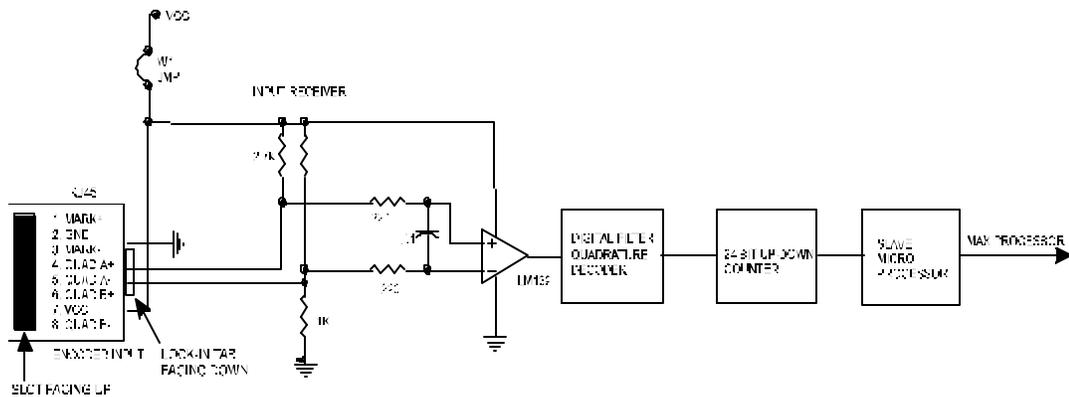
The CMAX is available with encoder feedback. This permits use of an optical encoder to provide position feedback. The following elements are used to provide this function:

1. Input buffers (two differential input comparators with R-C filters) to eliminate noise.
2. A digital filter to further sample and shape waveforms.
3. A quadrature decoder to convert inputs to count/direction and multiply count by four.
4. A 24 bit up/down counter and latch to track encoder position.
5. A slave processor to detect stall or position drift and notify host CMAX CPU.

Wiring Connections



Encoder Schematic



Most applications are reliable using the single-ended inputs (MARK+, QUAD A+ and QUAD B+). The standard 5-volt output can be used to supply power (80ma. maximum) to encoders such as the HP HEDS series. For special applications, an external supply, of up to 24Vdc, can be used if the internal jumper W1 is removed.

Note: Failing to remove W1 will destroy all of the electronics.

Design Notes

The ideal mechanical design is one with minimal backlash between the motor and encoder disk; that is the encoder mounted directly on the same shaft. Any gear reduction design must consider backlash and the encoder line count per motor revolution must be a multiple of 50. Backlash is compensated for through the use of the “s”, “t” and “d” commands.

Encoder Calibration (only necessary if the number of line counts on the encoder is unknown.)

- Step 1** Verify that the encoder is properly installed and connected.
- Step 2** Start the EASI program in the single axis, dumb terminal mode.
- Step 3** Strike the space bar and observe the presence of TWO numbers at the beginning of the sign-on.
- Step 4** Enter the following command sequence (observe the lower case for most encoder commands):
- | | |
|-------|--------------------------|
| D 0 | Full step |
| e 0 | Encoder off |
| o | Clear encoder position |
| R 200 | Index motor 1 revolution |
- Step 5** Verify that the motor shaft rotates 1 full revolution. If stalling occurs, evidenced by an incomplete index, tune the open loop parameters (I, K, V, and Y) for reliable operation.
- Step 6** Read the encoder position using the “z” (lower case) command.
- Step 7** Divide the value by 4. This should equal the physical encoder line count (compensate for gearing). The encoder quadrature resolution must be a multiple of the motor resolution.

Note: The encoder fabrication tolerances may result in 1 or 2 count uncertainties (or more in high resolution units). Once the encoder is enabled, all displacement commands are with respect to the encoder position counter, i.e., a 200 step motor, with a 2000 count encoder, is one revolution equal to “R200.” Whereas the same motor/encoder, with the encoder enabled, is one revolution equal to “R2000.”

- Step 8** Enter the command:
- | | |
|-----|-----------------------------------|
| R 0 | Index backward to start position. |
|-----|-----------------------------------|

The resulting encoder position should return to zero.

Note: In Encoder applications always use the “R” command. Never use the “+” or “-” command to position the motor, otherwise cumulative errors can occur.

Encoder Operation

The use of encoder feedback allows closed loop positioning system similar to a servo system. The CMAX contains two 24 bit position counters-the step counter and the encoder counter. The step counter will track the encoder counter until any encoder error occurs, then the position will be irrelevant. The step counter is never used in determining positions.

Example:

Assume the following conditions:

1. A 200 step/ revolution motor. These are natural full steps.
2. A 500 line encoder attached to the motor shaft.
3. The start “z” position = 0 (encoder counter).
4. The start “Z” position = 0 (step counter).
5. The “e” count (encoder count) per revolution is $4 \times 500 = 2000$.

Instructions

- Step 1** Enter the command:
e 500 Set the encoder mode. This zeros both position counters.
- Step 2** Issue the “X” command. The encoder related parameters should be displayed.
- Step 3** Set the step motor parameters:
H 0 Fixed resolution
D 1 Half step
- Step 4** Use the “S” command to save the encoder mode.
- Step 5** Issue an “R 1500” command. The following events are executed:
1. Compute the number of step motor steps required for 1500 encoder count = 150 full steps.
 2. Place the slave encoder processor in the stall detect mode.
 3. Initiate an open loop index of 300 half steps.

The 300 step index results in the motor reaching the desired 1500 encoder position. Upon successful completion, the slave processor will be instructed to switch to the position maintenance mode. If the position is within the Deadband range, the index is considered complete.

Stall Detect

Stall detect is used during the move portion of the index. The following parameters act upon this phase of an index:

1. “r” Retry Count; the number of retries to attempt when a stall is detected (the default is 5). The stall counter is reloaded at the start of an index and is decremented each time a stall is detected.
2. “s” Stall Factor; controls the distance the encoder **must** move for each sample (default is 20), i.e., $(s = (\text{encoder count}/\text{steps per rev.}) \times \text{fudge factor})$.
3. “t” Stall Sample Distance; sets the increments in motor steps when a stall condition is checked (default is 5). The stall is checked once for every 5 full motor steps.

The encoder processor is triggered every “t” motor steps (these are in full step motor steps, not encoder steps) or 40 times per revolution. The encoder movement (since the last sample) is computed. This delta must exceed the “lag” amount (in the correct direction) as determined by the “s” command. If not, a stall error signal is sent to the main processor, indicating insufficient shaft rotation, and a stall condition will exist. The main processor then would terminate the index, and if the retry counter is non-zero, re-compute and initiate a new index.

The computation of permissible “lag” in this case is $(e = 500 = 2000 \text{ counts/rev.}) / (200 \text{ motor steps per rev}) = 10$ “e” counts per motor step. The sample distance equals 5 full steps, and the encoder travel is 50 “e” counts.

The maximum allowable “lag” is determined by “s” as a percentage $(20/255)$ of the ideal encoder distance or $50 * 20 / 255 = 3$ steps. Although the value of “s” could be increased, it probably would not improve stall detection. When a step motor stalls the shaft stops.

Indexing With Encoders

An index command specifies, in encoder count, the target position to index to. The host computer/controller should always use the “R” command rather than a plus (+) or minus (-) command. This is because a +/- command will read the current encoder position then add the index size to compute a target position.

The encoder reading can vary by several counts and introduce an error into the expected position. The direction of the error is generally the same, resulting in a cumulative error.

Example:

Correct	Incorrect
R 100	+ 100
R 200	+ 100
R 500	+ 300
R 300	- 200
R 0	- 300

Encoder Indicators

Several LED's are provided as visual status indicators.

EA, EB and EC

Three yellow indicators are located just behind the encoder connector (J7). These LED's show the input signal levels generated by the encoder. The LED's are lit when the corresponding input is at a low.

1. EA – Channel A of the encoder
2. EB – Channel B of the encoder

The slow movement of an encoder should cause both LED's to blink "motion" in sequence with a 90 degree A-B quadrature. These are decoded and processed into the 24-bit encoder position counter.

Some encoders have an index channel, once per revolution. This LED can indicate a "home" position.

3. EC - Index input

ER and ST

The two LED's, close to the front of J7, represent the status outputs of the encoder microprocessor.

1. ER – Error signal (yellow)
2. ST – Stall detect (red)

ER indicates that the encoder has found an error condition. Depending on the operation, an error signal will be generated. The master CPU (when so enabled) uses this function to perform the corrective actions.

During an index the ER led will function in a "stall detect" mode. The Master CPU will then compute a new high-speed index to reach the target position. On reaching a point within the dead zone, motion will cease and the system will change to position maintenance mode.

Position Maintenance

During position maintenance – ER now indicates an outside dead zone condition. The Master CPU uses the "direction of error" (ST LED) to determine how to "tweak" the position. The CPU will attempt to position the motor to the exact target position (center of dead band).

Program Command Description

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
	<i>Mnemonic</i>	<i>Data 1</i> <i>(Range)</i>	<i>Data 2</i> <i>(Range)</i>	<i>Result</i>

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.

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
ESC	Terminate Operation		Immediate	
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) ESC			Echo #

(ESC) Global Abort

This command 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. Encoder auto-hunt functions are also aborted.

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.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
@	Soft Stop	Immediate, Program		2
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) @			Pound Sign (#)

(@) Soft Stop

This command is useful as a gentle stop. It behaves differently, depending on how it is used. If the axis is moving it will cause 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.

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.
- L 0 0 Loop to memory address location 0 until port 1 is low.
- @ Decelerate and stop program execution.
- P Exit program mode.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
^C	Reset all Axis			
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) ^C			

(^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. Determine encoder presence,
3. Calibrate power driver to on-phase position,
4. Set origin(s) to zero,
5. Calibrate motor current to hold value,
6. Test for and execute any User power-up program starting at location 1600,
7. Assume an idle state waiting for GO, Jog, or serial command pulse input,
8. Reset driver.

Encoder position maintenance will not be active until a motion function happens.

A	Function Read/Write to Ports		Type Immediate, Program	NV Bytes 2, 2
	Mnemonic (Name) A	Data 1 0-128	Data 2	Result Non/Read State

(A) Port Read/Write

The Port command provides access to the auxiliary I.O. ports. Six (6) general-purpose ports are available for User applications. The pre-configured hardware defines 3 inputs (ports 1, 2 and 3) and 3 outputs (ports 4, 5 and 6). All ports are read or written as a parallel 6 bit binary value.

Value	Port-4	Port-5	Port-6 (Trip)
0	Off	Off	Off
8	On	Off	Off
16	Off	On	Off
24	On	On	Off
32	Off	Off	On
40	On	Off	On
48	Off	On	On
56	On	On	On

An "On" condition is defined as an electrical low (inverted) at the output pin. Other values beside those listed may cause improper operation of the input ports.

Reading of the inputs is accomplished by using a value of 129 ("A 129"). A number containing 6 bits of information is returned in the result.

The physical port is read as a byte, and then inverted before being sent to the "host." After the host converts this number to a binary value, the following information may be obtained:

Input	Input	Input	Output	Output	Output
Port 1	Port 2	Port 3	Port 4	Port 5	Port 6 (Trip)
1	2	4	8	16	32

Inputs and outputs are the TTL compatible levels with 10K pull-up resistors.

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.
P 0      Exit program mode.
    
```

B	Function Set Jog Speeds, Slow/Fast		Type Default, Immediate, Program	NV Bytes 3
	Mnemonic (Name) B	Data 1 Slow 0-255	Data 2 High 0-255	Result

(B) Set Jog Speeds

Data range 1 and data range 2 represent the speeds to use when the jog inputs are utilized. The first is usually a slower speed. The second number is used when the high-speed jog (J5 pin 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 inputs. Jog resolution is determined by the microstep mode "H" and "D" values. Ramped acceleration using the "K" value is implemented. Ramped deceleration is used if the high-speed input is released while maintaining one of the two jog inputs on (low). Both CW and CCW inputs operate at the same speeds.

The Jog inputs are active:

1. After power-up.
2. When not running a program.

Jogging is not active during actual command line entry.

C	Function Reinitialize Non-volatile Memory		Type Immediate	NV Bytes
	Mnemonic (Name) C	Data 1 Page 0-9	Data 2	Result Version Number

(C) Clear and Restore

There are a total of 8 pages of program memory. The data values refer to the page of non-volatile memory to clear. Page 8 (1792-2047) contains the parameter default storage, pages 1 through 7 (0-1791) may be used for User programs. Reference, "Non-Volatile Memory" in the Operating Parameters section.

Clear and Restore reinitializes or clears the systems non-volatile memory. The page number defines the operation. All page numbers, except a zero, cause a 256-byte write cycle, taking approximately 200ms. Frequent use of this command should be avoided as memory longevity may be affected.

"C 0" reloads the last saved parameters from non-volatile memory. No write operations occur and operation is similar to what happens during a power-on reset. "C 1" through "C 7" will erase the corresponding page of non-volatile memory, 256 bytes at a time.

Note: Programming overwrites non-volatile memory as it proceeds. It is not necessary to use this complete erase, except to "clean up" a segment.

"C 8" forces complete non-volatile initialization to factory default values. This should be done only when new non-volatile memory is installed or existing memory is corrupted.

"C 9" forces initialization of the "micro look up table." This list of 100 eight-bit numbers starts at location 1792 in page 8 of the memory map. Each value corresponds to a micro position with a resolution of 1/256 micro step. The default values assume an ideal linear relationship between phase currents and actual armature positions. Designers may insert their custom tables here. Any new list must be monatomic in nature.

C	Function Reset Driver		Type Immediate	NV Bytes
	Mnemonic (Name) c	Data 1	Data 2	Result

(c) lower case c; Reset Driver

This command sends a reset to the driver module. It is useful to clear a fault condition. The driver is reinitialized to the "on phase" (0) state with this command.

D	Function Set Resolution in Fixed Mode		Type Immediate, Program	NV Bytes 2
	Mnemonic (Name) D	Data 1 Resolution 0-8	Data 2	Result

(D) Divide Resolution

This command is used to set the microstep resolution when operating in the fixed resolution mode. Indexes are made using the specified resolution.

Resolution Setting	Step Size	Max RPM
0	Full	6,000*
1	1/2	3,000
2	1/4	1,500
3	1/8	750
4	1/16	375
5	1/32	187
6	1/64	93
7	1/128	46
8	1/256	23

*Do not use this setting when operating in encoder mode.

This command may be used (with caution) when operating in the Variable Resolution mode. Its effect is to pre-scale the motor shaft speed to operate with higher resolution. The programmer should observe that the maximum velocity specified does not exceed the maximum shaft RPM available at a given resolution.

E	Function Settling Time Delay		Type Default, immediate, Program	NV Bytes 2
	Mnemonic (Name) E	Data 1 0.01 Seconds 0-255	Data 2	Result

(E) Settling Time Delay

The system features an automatic motor reduction capability. After an index, the winding current will be reduced or shut off as determined with the "Y" parameter.

The E command specifies the delay in 10's of milliseconds before activation of the auto-power-down feature. Successive indexes will activate the run current (as required) prior to the index. Auto-power-down will not occur if the encoder auto-hunt-mode is on.

F	Function Find Home		Type Immediate, Program	NV Bytes 3
	Mnemonic (Name) F	Data 1 SPS 20-23,000	Data 2 Direction 0-1	Result

(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 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

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.
```

G	Function Go from Address (n)		Type Immediate, Program	NV Bytes 3
	Mnemonic (Name) G	Data 1 Address 0-1791, 2048	Data 2 Trace 0-1	Result

(G) Go

The Go command executes a User programmed sequence starting at a pre-defined 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 Location
P1	P2	P3	P4	
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. Reference: BRANCH.SMC program structure on the EASI diskette.

The controller is factory set with the following program example:

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

H	Function Resolution Mode		Type Default, Immediate, Program	NV Bytes 2
	Mnemonic (Name) H	Data 1 FR/VR 0-2	Data 2	Result

(H) Resolution Mode

This command defines the primary step mode. The position counter is reset to 0.00 and motor phase is re-calibrated.

H 0 - Fixed Resolution (FR Mode)

Stepping occurs at a fixed resolution (full to 1/128) determined by the "D" command. The index distance for a given step count is also determined by the specified resolution, i.e., a +200 step index will result in 1 revolution at full step, 1/2 revolutions at 1/2 step, 1/32 revolution at 1/32 step, etc. The maximum shaft speed is limited to 20,000 steps/second at a given resolution. Fractional positioning is not available. Ramping is used.

H 1 - Auto Variable Resolution (VR Mode)

This is the most powerful index capability of the system. The controller determines the optimal step resolution for the specified initial and slew velocities. During acceleration or deceleration, step resolution is coordinated with step rate and changed at predetermined speeds to maintain smooth acceleration. This "gear shifting" always occurs on a motor full-step pole position, where torque is highest.

The following events take place in response to an index command. This example assumes an index in excess of 1 full step:

1. Any prior motion from a previous command is completed.
2. The next "TARGET" position is calculated in "N" whole steps and a fractional "F" remainder.
3. The motor is energized to the run current, as required.
4. A coarse index of "N" whole steps is executed.
5. The motor is "trimmed" to the final position "F" with 1/256 step resolution. If the encoder option is implemented this final trim sequence is "homed" on the encoder position instead. The target position is specified in terms of nnnnnn.ff, where n is whole steps and ff is in .01 steps.

H 2 Step and Direction Output

The H2 command was added that makes it possible to convert two output ports to step and direction output signals. This can be used to control additional "slave" drives.

The H2 command:

- A. Directs the step pulse to port 5 and direction to port 6,
- B. Places the controller in fixed resolution mode,
- C. Disables port commands that can affect ports 5 and 6.

When this mode is in use, several rules must be observed:

1. Ports 5 and 6 become step and direction outputs and cannot be used as general-purpose ports.
2. Fixed resolution mode must be used.
3. The step rate limit, with no options, is approximately 31,000 SPS.
4. The step rate limit, with step and direction enabled, is approximately 29,000 SPS.
5. The step rate limit, with step, direction and trip enabled, is approximately 26,000 SPS.
6. The low going step pulse width is approximately 12 microseconds.

The output sinking signals are open collector or drains. Weak pull-up resistors to VCC or VIO are incorporated. The actual output sink current and voltage ratings are dependent on the specific AMS model number.

Exceeding the maximum step rates manifests itself as a watchdog timeout error and the unit will halt and perform a hardware reset, leaving the watchdog timer flag set. The single axis sign on message will produce a “!” character. The unit must be reset by cycling power or using the “] 1” command.

The direction signal is set prior to the falling (leading) edge of the step pulse.

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 to variable resolution.
- P Exit program mode.

I	Function		Type	NV Bytes
	Set Initial Velocity Speed		Default, Immediate, Program	3
	Mnemonic	Data 1	Data 2	Result
	(Name) I	SPS 20-20,000		

(I) Initial Velocity

This command 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 and motor or system resonance. Generally values between 50 and 1200 SPS are appropriate.

The initial velocity command sets the parameters to be used in subsequent motion commands. In Variable Resolution Mode the speed is in equivalent full steps per second. Note that the criterion for selecting these velocities is base on the physical motor RPM. As microstep divide (D) factors are applied, the value of “I” may be increased to counteract the reduced shaft speed.

In Fixed Resolution Mode the speed is in pulses per second applied at the specified resolution, i.e., the shaft speed is halved for each smaller microstep setting (see the "D" command).

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.

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.

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

(i) lower case I; Restart Special Trip (Reference lower case “k” command).

*Actual values are determined by the hardware configuration.

J/j	Function Primary & Secondary Jump		Type Program		NV Bytes 4
	Mnemonic (Name) J/j	Data 1 Address 0-1791		Data 2 N + 1 Times 0-255	Result

(J/j) Jump

Jump to address; count +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. Only one jump counter is available for use at any given time. However, the Primary "J" command and Secondary "j" command can be nested. Use of the "G" command is recommended for single jump commands.

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.
```

K	Function Set Ramp Accel/Decel Time		Type Default, Immediate Program		NV Bytes 3
	Mnemonic (Name) K	Data 1 Accel 0-255		Data 2 Decel 0-255	Result

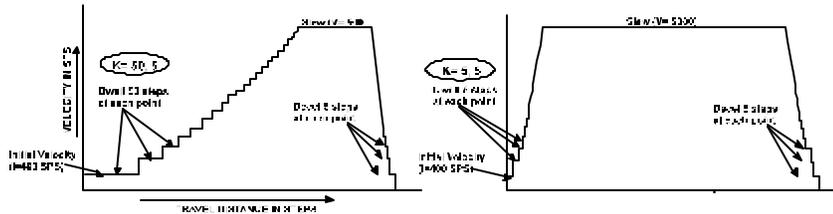
(K) Ramp Slope

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, 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.

The following is an example of two 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 10 (Accel) 10 (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 reduced if the microstep resolution (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	Special Trip	Program	5
	Mnemonic (Name) k	Data 1 Position ±8,388607	Data 2 Port Value 0-56

(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 the actions are performed in real time. The best way to illustrate the these commands is with an example:

Enter as follows:

```
P 0          Start programming mode
0          O 0          Set position to zero
4          k 200 8      Trip position 200, port 4 on, RAM=128
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 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 further describes the program sequence:

<u>Address</u>		<u>Description</u>
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. here 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

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
L	Loop on Port	Program		4
	<i>Mnemonic</i> (Name) L	<i>Data 1</i> Address 0-1791, 2048	<i>Data 2</i> Condition 0-8	<i>Result</i>

(L) Loop on Port

The Loop command 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 as follows:

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 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.

Note: Reference the special case of the Go command for branching on inputs.

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.
L 0 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.
    
```

I	Function Invert Limit Polarity		Type Default, Program	NV Bytes 2
	Mnemonic (Name) I	Data 1 Invert 0-1	Data 2	Result

(I) lower case L; Limit Polarity

A value of one will effectively invert the limit input polarity. A logic low must be maintained at the Limit switch inputs to allow motion.

M	Function Move at Constant Velocity		Type Immediate, Program	NV Bytes 3
	Mnemonic (Name) M	Data 1 SPS ±22,000	Data 2	Result

(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 sign preceding the velocity specifies the direction. The system has the capability of decelerating from full speed in one direction to full acceleration speed in the opposite direction 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).
4. "Go" command.
5. "Jog" command.

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.

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.
L 0 0 Loop to memory address location 0 until port 1 is low.
@ Decelerate and stop program execution.
P Exit program mode.

```

|          |                                        |                                      |                        |                      |
|----------|----------------------------------------|--------------------------------------|------------------------|----------------------|
| <b>N</b> | <b>Function</b><br>Force Step Position |                                      | <b>Type</b><br>Program | <b>NV Bytes</b><br>5 |
|          | <b>Mnemonic</b><br>(Name) N            | <b>Data 1</b><br>Steps ±8,388,607.99 | <b>Data 2</b>          | <b>Result</b>        |

**(N) Force Step Position**

This command forces the "Step" position (as opposed to encoder) to the value specified. The encoder value is not changed.

|          |                                                   |               |                        |                      |
|----------|---------------------------------------------------|---------------|------------------------|----------------------|
| <b>O</b> | <b>Function</b><br>Reset Position Counter to Zero |               | <b>Type</b><br>Program | <b>NV Bytes</b><br>1 |
|          | <b>Mnemonic</b><br>(Name) O                       | <b>Data 1</b> | <b>Data 2</b>          | <b>Result</b>        |

**(O) Set Origin**

This command resets the internal position counter to Zero. Units with encoder feedback are also reset. This command is not allowed while moving. The following events will also reset the origin:

1. Power on
2. (e) enabling the encoder

The micro trim position is re-calibrated to the "on pole" position as required (non-encoder mode).

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.
    
```

P	Function Program Mode		Type Immediate	NV Bytes
	Mnemonic (Name) P	Data 1 Address 0-1791	Data 2	Result

(P) Program Mode

This 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 through 192 are downloaded from the external EEprom to internal RAM at power-up. Instructions executing in this segment run faster than 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 pre-defined 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):

<u>Response</u>	<u>Command</u>	<u>Comment</u>
	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
	Q	(You may now list the programs)
	G 100	(You may now execute the programs)
	P 101	(You may now edit the program)
101	R 3000	Change position from 2000 to 3000
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
B Jog Speeds	User default from non-volatile memory at power-up
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
H Resolution Mode	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
Y Current Control	User default from non-volatile memory at power-up

All of the encoder parameters are user default from non-volatile memory at power-up. (Of course, the program may be used to modify User defaults).

<i>Command</i>	<i>Function</i>	<i>Type</i>	<i>NV Bytes</i>	
Q	List Instructions in NV Memory	Immediate		
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) Q	Address 0-1791		

(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.

<i>Command</i>	<i>Function</i>	<i>Type</i>	<i>NV Bytes</i>	
q	List Programs (20 Lines)	Immediate		
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) q	Address 0-1791		

(q) Query Program (20 Lines)

This command (lower case q) will list application programs from non-volatile memory address, 20 lines at a time. See the upper case "Q" command.

R	Function Index Relative to Origin		Type Immediate, Program	NV Bytes 4
	Mnemonic (Name) R	Data 1 Position ±8,388,607.99	Data 2	Result

(R) Index Relative to Origin

Move with ramping relative to the "0" origin. The distance of the index is determined by subtracting the last (or current) position from the target position. If an index is under way when the command is issued, the controller will wait until that index is complete.

The units used for target position can assume three meanings based on the mode of operation:

1. Non-encoder, Variable Resolution Mode, Range ±8,388,607.99 Full Steps

The target position is specified in terms of whole motor steps (typically 200 steps per revolution) and a fractional portion with .01 step resolution. Indexing will proceed at various resolutions, based on velocities, until the target "whole step" position is reached. A fine adjustment is then made to position the rotor to .01 to .99 step.

2. Non-encoder, Fixed Resolution Mode, Range ±8,388,607 Steps

The target position is specified in terms of steps at the specified resolution (see the "D" command). This mode permits step translations of 1/200 through 1/51,200 motor steps per revolution. Shaft speed is limited to 20,000 steps per second at a given resolution.

3. Encoder, Variable Resolution Mode, Range ±8,388,607 Encoder Position

The target position is specified in terms of encoder position. The shaft resolution is 4 times the encoder line count. Indexing will proceed at various resolutions, based on velocities until the target is reached. A fine adjustment is then made to position the encoder at the specified target. The resolution used for this phase is up to 1/256 motor/step. Automatic stall detect (with "n" retries) may be enabled. Automatic position maintenance after index may also be 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.
    
```

Related commands: D, E, H, I, K, O, V.

S	Function Store Parameters in NV Memory		Type Default	NV Bytes
	Mnemonic (Name) S	Data 1	Data 2	Result

(S) Store Parameters

Many parameters are automatically read from non-volatile memory each time power-up occurs, eliminating the necessity of inserting redundant initializing commands in programs. The following are User defaults that are stored in the non-volatile memory:

Parameter	Standard Defaults
Initial Velocity (I)	400 (steps per second)
Slew Velocity (V)	3004 (steps per second)
Divide Factor (D)	1
Ramp Slope (K)	10/10
Jog Speeds (B)	3, 20 (90, 600 steps per second)
Trip Point (T)	Off
Step Resolution (H)	0 (fixed)
Auto Power Down	Yes
Hold/Run Current (Y)	5, 25
Limit Polarity	Low
Auto Position Readout (Z)	Off
Name (after reset)	Undefined
Encoder Option Only	
Encoder Resolution (e)	500 (ratio=10)
Deadband (d)	30
Hunt Velocity (y)	700/32
Stall Factor (s)	20
Stall Test Delta (t)	5
Set Stall Retry Count (r)	5 (lag=3)

The Store command also saves the contents of the high speed "Shadow RAM." Locations between 128 and 192 are stored only when this command is issued, extending the life of the EEPROM. These 64 locations are automatically downloaded on power-up reset.

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

Command	Function	Type	NV Bytes	
T	Trip Point	Default, Program	5	
	Mnemonic	Data 1	Data 2	Result
	(Name) T	Position $\pm 8,388,607$	Vector 0-255	

(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 Carriage Return):

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.

U	Function Control Line Feed		Type Immediate, Program	NV Bytes 2
	Mnemonic (Name) u	Data 1 0-127	Data 2	Result

(u) lower case u; Control Line Feed

The “u 0” command inserts an “end of line” in the output and is useful when using the program trace options. Inserting decimal values 1 to 127 after the u outputs the corresponding ASCII character code, i.e., u 80 outputs a “P.” This can useful as an indicator that a sequence is complete or maybe for de-bug purposes.

V	Function Set Slew Velocity		Type Default, Immediate, Program	NV Bytes 3
	Mnemonic (Name) V	Data 1 SPS 0-20,000	Data 2	Result

(V) Slew Velocity

Set final Slew Velocity after ramping up to speed.

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)
- M (Move at a Constant Velocity)

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.

W	Function Wait (Delay)		Type Immediate, Program	NV Bytes 3
	Mnemonic (Name) W	Data 1 0.01 Seconds	Data 2 0-65,535	Result

(W) Wait

WAIT n x 10 milliseconds. The controller will remain in an idle state for the specified time. This time is derived from the crystal clock and is very repeatable and accurate. It is independent of motion in progress and times functions while moving. One timer is available.

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.
    
```

Note: SPECIAL CASE “wait” for motion completion: Using a 0 (zero) time value will delay the next command/instruction until any motion has been completed. In the Immediate Command mode the echoed line feed is delayed providing an alternative means for the host to determine motion complete (only practical in single axis designs).

W	Function		Type	NV Bytes
	Read Error Status of Driver		Immediate, Program	
	Mnemonic	Data 1	Data 2	Result
	(Name) w			

(w) Driver Status

This command (lower case w) will read the error status of the driver section. The host may determine the presence and cause of a fault. Faults include under voltage, over temperature and short circuit latch. A non zero result indicates the error condition:

- 1= Over temperature. The heat sink temperature has risen beyond ratings.
- 2= Short circuit. A short to ground or other fault has occurred.
- 4= Under-voltage. The motor supply is defective, a blown fuse or low line.

Any condition will cause a shutdown of the power drive circuitry. For safety reasons the driver outputs will not automatically re-enable when the proper voltage and/or temperature condition is restored but rather requires the driver to be reset (^c command).

A short condition triggers a latch that must be reset by a power off-on cycle or by issuing a clear command.

A reading of 7 represents an absolute failure of unit.

X/x	Function Examine Parameters		Type Immediate	NV Bytes
	Mnemonic (Name) X/x	Data 1	Data 2	Result Display Setting

(X/x) Examine Parameters

Single Line Mode Only

The Examine command displays settings of many parameters and provides useful information. This command will produce two different responses, depending on the mode of Motion command operation. Two lines of operational parameter data is displayed upon power-up. Up to five lines of data is displayed if the Encoder and Auto position function is enabled. When in the non-Party Line (Single) mode the display is as follows:

Line #1: Y = hold/run, E = delay, K = up/dn, H = vr{fr}, na=name

Where: hold = hold current, run = run current, delay = settling time before activation of power down, up = acceleration slope, dn = deceleration slope, vr = Variable Resolution mode, fr = Fixed Resolution mode, name = most recent name assignment for Party Line.

Line #2: I= iv {av/res}, V= sv {av/res}, {rl= nnn}

Where: iv = initial velocity, sv = slew velocity, av = actual velocity used, res = actual resolution used, rl = ramp length (multiply times K for actual).

Line #3: (if encoder is enabled) e=ll, {ratio = rr}

Where: ll = lines specified, rr = computed ratio (lines/full-step).

Line #4: (if auto position is enabled) d = dz, v = vh / hres

Where: dz = dead zone size, vh = hunt steps per second, hres = hunt resolution.

Line #5: (if auto stall is enabled) s = ss, t = tt, r = rr, {lag= lll}

Where: ss = stall percentage, tt = test delta, rr = retry attempts, lll = maximum allowable encoder lag distance (based on s and t values).

Items displayed in {parentheses} are computed internally, based on parameter settings.

Party Line Mode Only

In Party Line mode, the X command operates differently. The host may obtain the contents of registers specified by the address value. The returned values are obtained by "peeking" into the internal working registers. This feature should only be used in very specialized applications. The "C" application program contains additional information. Please note that the address, or definition of contents, is subject to change at any time. An address value of 0 will return a model number (32) to indicate the type of controller.

Y	Function Program Hold and Run Current		Type Default, Immediate, Program	NV Bytes 3
	Mnemonic (Name) Y	Data 1 Hold 0-100	Data 2 Run 0-100	Result

(Y) Hold and Run Current

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

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

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

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

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

Quiet Mode

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

<u>Enter</u>	<u>Result</u>
Y00 80	Initiates current Program mode, sets Hold current at 0% and Run current at 80% of maximum.

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

Z	Function Read Position		Type Immediate	NV Bytes
	Mnemonic (Name) Z	Data 1 Continuous 0-1	Data 2	Result Position

(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" command.

There is an option of continuous readout via the serial interface. The "Z 1" command enables this operation. Any change in position causes the position data to be sent to the serial output. The readout 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 single axis protocol. The decimal portion will be non-zero only when fractional positioning is enabled (non-encoder, Variable Resolution mode).

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

(I) Read Non-Volatile Memory

This command allows the User to display any byte of the 2047 byte external non-volatile memory. The address specifies the desired location to access. The optional count allows display up to 255 locations, 10 bytes per line.

]	Command	Function Read Limits/HW/Watchdog	Type Immediate, Program	NV Bytes 1
	Mnemonic (Name)]	Data 1 LM/HW/ Watchdog 0-2	Data 2	Result

(J) Read Limits/Hardware/Watchdog Timer

Note: This command is optional. Please consult factory for details.

"10" Query Limits and Watchdog

A Zero is returned if no limits or watchdog flags are on. The returned number contains the following information "bits:"

- 1 = Limit A
- 2 = Limit B
- 4 = Watchdog timed out
- 8 = Brown-out

"11" Query Hardware Status and Reset Watchdog Flags

This command permits the User to examine the status of the various signal inputs.

The result will contain the sum of the decimal values as follows:

Decimal Value	Signal Input
1	Home Input (cannot be read when executing home command)
2	Encoder Error Direction (not readable)
4	Encoder Error (not readable)
8	Go Input (not readable)
16	Soft Stop Input
32	Jog + Input
64	Jog - Input
128	Jog Speed Input

Some of these may be used as general purpose input ports if the User is careful to take appropriate actions to avoid conflicts.

"12" Set Watchdog Flag

This command emulates the Watchdog Timer function. It can be useful for test and de-bug purposes.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
+	Index in Plus Direction	Immediate, Program		5
	<i>Mnemonic</i> (Name) +	<i>Data 1</i> Steps 0.01-8,388,607.99	<i>Data 2</i>	<i>Result</i>

(+) Index in Plus Direction

Step in a positive direction for the specified distance. The motor will ramp up, slew, then ramp down per the previously set parameters. If an index is under way when the command is issued, the controller will wait until that index is complete. The units used for target position can assume three meanings based on the mode of operation:

1. Non-encoder, Variable Resolution Mode, Range ±8,388,607.99 Full Steps

The target position is specified in terms of whole motor steps (typically 200 steps per revolution) and a fractional portion with .01 step resolution. Indexing will proceed at various resolutions, based on velocities until the target "whole step" position is reached. A fine adjustment is then made to position the rotor to .01 to .99 step.

2. Non-encoder, Fixed Resolution, Range ±8,388,607 Steps

The target position is specified in terms of steps at the specified resolution (see the "D" command). This mode permits step translations of 1/200 through 1/51,200 motor steps per revolution. Shaft speed is limited to 20,000 steps per second (at a given resolution).

3. Encoder, Variable Resolution Mode, Range ±8,388,607 Encoder Position

The target position is specified in terms of encoder position. The shaft resolution is typically 4 times the encoder line count. Indexing will proceed at various resolutions, based on velocities until the target is reached. A fine adjustment is then made to position the encoder at the specified target. The resolution used for this phase is up to 1/256 motor/step. Automatic stall detect (with "n" retries) may be enabled. Automatic position maintenance after index may be enabled.

Related commands: D, E, H, I, K, O, V.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
-	Index in Minus Direction	Immediate, Program		5
	<i>Mnemonic</i> (Name) -	<i>Data 1</i> Steps 0.01-8,388,607.99	<i>Data 2</i>	<i>Result</i>

(-) Index in Minus Direction

Except for the direction, this command behaves exactly as the "+" command above.

Command ^	Function Read Moving Status	Type Immediate, Program		NV Bytes 1
	Mnemonic (Name) ^	Data 1	Data 2	Result Status Byte

(^) Read Moving Status

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

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

Bit	Decimal	Comment
0	1	Moving - indicates axis moving
1	2	Constant - high in constant velocity
3	8	Homing - homing routine is active
4	16	Hunting - encoder correction

Other bits in this result should be ignored.

Command \	Function Write to Non-Volatile Memory	Type Immediate		NV Bytes
	Mnemonic (Name) \	Data 1 Address 0-2047	Data 2 Number 0-255	Result

(\) 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.

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
/	Terminate Program		Immediate	2
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) ¹			Status Byte

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

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

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
^N	Name Controller		None	N/A
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	^N	None	None	None

^N (Name Axis)

Response to the single character "Ctrl N" (14 decimal, 0E hex) is immediate.

This command must be executed with one axis attached, otherwise a bus conflict will occur and all axes may receive the same name. On entry of the "Ctrl N" the controller responds with "name?" Type the desired name, the character will be stored in NV memory and the CMAX is ready to enter party line mode.

<i>^P</i>	<i>Function</i> Restart in Party Line Mode		<i>Type</i> None	<i>NV Bytes</i> N/A
	<i>Mnemonic</i> ^P	<i>Data 1</i> None	<i>Data 2</i> None	<i>Result</i> None

^P (Party Line Mode)

Response to the single character “Ctrl P” (16 decimal, 10 hex) is immediate.

This control character restarts the controller in party line mode. One, or up to 32 controllers may be connected. The “^C” (or cycle power) command is used to restart in single controller (space to sign-on) mode.

AMS - SIN-11 Serial Adapter

If you are using a SIN-11 serial adapter from AMS, the “&” command must be used to initiate party line operation. On receipt of the “&” command, the SIN-11 automatically transmits the ^P character within the party line startup routine. See the serial communication section for more information.

a	Function Read Stall Count		Type Immediate	NV Bytes 1
	Mnemonic (Name) a	Data 1	Data 2	Result 0-255 Stalls

(a) Read Stall Count

This command reads back a number that indicates if and how many stalls occurred during a move. At the beginning of an index (+, -, or R) command the user defined retry count (see “r” command) is loaded into the retry counter. If excessive encoder position lag is detected, a “stall” error will initiate a re-index attempt.

Each re-index attempt will increment the stall count by one until it equals the retry count. The value returned will range between zero (ideal) and your specified retry value (the factory default is 5). If your system is tuned properly this counter should never count up (zero retries). Stalls under “normal” conditions indicate problems that should be found and fixed before they reach the customer.

Among the possible causes are:

- Excessive load or undersized motor
- Wrong parameter settings
- Speed (I, V) settings too high or near a resonance point
- Acceleration (K) too slowly through a resonant frequency
- Current (Y) set too low
- Lag value too stringent (determined by s and t values)
- Mechanical lag or backlash
- System spring or wind-up

d	Function Set Deadband and Enable		Type Immediate, Program, Default	NV Bytes 3
	Mnemonic (Name) d	Data 1 0-255 Deadband Size	Data 2 None	Result

(d) Set Deadband and Enable

This command specifies the differential position distance, in encoder counts, that the motor shaft is permitted to differ from the encoder position register before automatic position correction (hunt) is executed. When a correction is required the position is re-homed to the desired position. The total dead zone value is double the specified distance, thus a value of 10 will maintain the position within ±10 encoder steps.

After completing a move using automatic position correction, further position corrections will be automatic. Full motor power is maintained and the moving output signal is asserted "on." The minimum practical value is affected by encoder resolution, backlash, and hunt step rate/resolutions.

A value of 0 disables this function. A non-zero value activates position maintenance immediately, locking the position to the current encoder position. An abort command (ESC) will shut off the maintenance.

Example:

The encoder position is 1100 and the “current position”(target) register contains 2000.

Note: the current position should be equal (or very close) to the encoder position.

This example assumes that the motor has stalled, without subsequent correction, leaving the two registers out of synchronism. When a “d 15” command is executed, the following actions take place:

1. The encoder position is placed into the “current position” register.
2. The position maintenance mode is turned “on.”

The “current position” now equals 1100 and the motor is “locked” into position 1100. The dead zone has a tolerance of plus or minus 15 encoder steps from the “current position” register. The encoder position is allowed to wander within this 30-count range. If the encoder position exceeds the specified range, i.e., 1116, an automatic hunt cycle (position correction) is triggered. This will “servo” on the encoder position, adjusting until the encoder is at 1100. See the “g” command for more information.

Note: A dead zone value that is too low can result in constant hunting or shaft oscillation.

If the Hold and Run currents are set to widely different values, a position shift will occur on auto-power down. This may trigger a hunt cycle that will power up the windings. This cycle will continue to repeat, resulting in a periodic shaft oscillation at a low rate. Aside from handling of positions, the “d” and “g” commands perform almost identical functions.

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
e	Enable Encoder		Immediate, default	3
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) e	0-2000 Line Count		

(e) Enable Encoder

This command specifies the encoder resolution, in lines, for one revolution of the motor. The controller multiplies this value times 4 for encoder position readings. The encoder used must have 50 lines per motor revolution or be a multiple of 50, i.e., 200, 250, 300, etc. A 500 line encoder will produce 2000 steps per motor revolution. The data is based on a standard, 200 step-per-revolution (1.8°), motor. Non-standard applications must be scaled appropriately.

A value of 0 disables all encoder functions, and subsequent indexing is in motor steps rather than encoder steps.

Note: The encoder used must be 50 lines per motor revolution or a multiple of 50; i.e., 200, 250, 300, etc.

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
f	Find Encoder Mark		Immediate, Program	
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) f	0-1 Direction		

(f) Find Encoder Mark

This command causes the motor to rotate in the specified direction until the optional encoder marker output (option on encoder) is detected. Searching is accomplished using the hunt speed (“v”) and resolution (“h”). To avoid missing the index signal, the micro step resolution should be high. Values between 1/32 and 1/256 are recommended.

Note: The mark is only usable in the fixed resolution mode.

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
g	Hunting On/Off		Immediate, Program	2
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) g	0-255 Dead Zone Size		

(g) Hunt Mode

This command will turn off the auto-hunt and initiate the hunt mode using the specified deadband value. If the target position does not match the actual encoder position a hunt cycle will be initiated.

The hunt cycle is a servo cycle that will trim the motor to match the value contained in the “current position” register. This register is sometimes called the “target position”, since it is loaded with the destination position at the initiation of an index.

The following events will take place during the hunt cycle:

1. Compare the encoder position with the “target position” register.
2. Determine the direction of the error.
3. Power-up the motor to the “Run” current.
4. Set constant velocity-fixed resolution mode with no acceleration ramp.
5. Start seeking at a speed set by the hunt resolution (h) and velocity (v).
6. Step and test until the encoder position equals the target position register.
7. When the encoder reaches the target position, the hunt cycle is complete.
8. Auto power down to hold current will function as in any other motion.

Monitoring of the encoder position will continue and any disturbances will trigger another correction cycle. The monitoring of encoder position errors can be stopped via the “g 0” command or the abort (ESC) command from a host. See the “d” command for more information.

<i>Command</i>	<i>Function</i>		<i>Type</i>	<i>NV Bytes</i>
h	Hunt Resolution		Immediate, Program, Default	2
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) h	0-8 Microstep Size		

(h) Hunt Resolution

The Hunt Resolution command defines the motor microstep resolution used during deadband re-calibrate operations. A value of 0 corresponds to a full step and an 8 will equal 1/256 of a step. See the "H" command for more information.

For moderate to high-resolution applications, resolutions of 1/16 and above are recommended. Very high resolutions will tend to increase the time required for large position error corrections. See Commands: "v" (Hunt Velocity) and "d" (Deadband Enable).

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
m	Index Mode	Immediate, Program, Default		2
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) m	0-1		

(m) Index Mode

This command sets the index behavior in response to the + command (index in plus direction) or the – command (index in minus direction). There are two index modes, m0 and m1.

m0

The standard index mode behaves as follows in response to a + (n) index command:

1. Read the current encoder position.
2. Add (or subtract) the distance (n encoder steps).
3. Place this number into the “target” register.
4. Compute and index to the target position.

If the deadband mode is on when the index is completed, the controller will start monitoring for position errors.

Note: No action is taken if the final position is within the dead zone boundaries, so the encoder is allowed to be off the exact target by n encoder counts.

If another + index is executed, steps 1-4 are repeated. In step 1 the encoder reading could be off by +3 counts. The target register will be off by 3 counts. If multiple + index commands are executed these small “errors” will accumulate and manifest itself in the form of position creepage. Thus, the + command does exactly what it is intended to do- move the specified distance from the current position.

Note: The + index commands are not intended to be used when the encoder mode is used. Only the “R” command will result in proper positioning.

m1

When the m1 option mode is implemented the “ideal” position is always computed using registers rather than encoder readings. Step 1 is changed to “read the current target position” (the last index). This internal tracking of position essentially converts the + index to the “R” command equivalent, always having an ideal target position.

The “m” command index mode is stored in NV memory as a startup parameter.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
n	Force Encoder Position	Immediate, Program		5
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) n	±8388607 Position		

(n) Force Encoder Position

This command will “jam” a 24-bit number into the encoder counter. It may be useful in a system with power fail and recovery capabilities.

O	Function Set Origin To Zero		Type Immediate, Program	NV Bytes 1
	Mnemonic (Name) o	Data 1	Data 2	Result

(o) Set Origin to Zero

This command sets both step and encoder position counters to 00.00. This command is identical to the upper case version.

r	Function Set Retries		Type Immediate, Program, Default	NV Bytes 2
	Mnemonic (Name) r	Data 1 0-255 Retry Count	Data 2	Result

(r) Set Stall Retry Count

This command will automatically re-attempt to execute a new index if, during the course of an index, a stall condition is detected:

1. Stop motion.
2. Read encoder position.
3. Automatically execute a new index based on the new computation.

Upon exhaustion of the retry count the controller will still attempt to acquire the desired position if the Hunt (Dead band) feature is enabled.

Limit inputs or Abort (ESC) commands will terminate retries.

Note: If the Deadband function is not enabled and stalls occur then the position will be completely lost if the + or - index command is used.

Use the “m” command to compensate for unintended position errors that are normally based on reading the encoder position.

IMPORTANT

When the auto deadband function is not enabled, the “g” command should be used to remove excessive position errors.

S	Function Stall Detect		Type Immediate, Program, Default	NV Bytes 3
	Mnemonic (Name) s	Data 1 0-255 Distance Factor	Data 2	Result

(s) Stall Detect

This feature determines if the motor has stalled, or slipped poles during a move. This is accomplished by periodically reading the encoder counter, and ensuring that it has changed at least by the value specified by “lag.”

Lag is established by the ‘s’, ‘t’, and ‘e’ parameters as follows:

$$\text{lag} = (\text{ratio} * \text{s} * \text{t}) / 256$$

Where:

- lag= must move distance
- e= encoder lines per revolution
- ratio= $(e * 4) / (\text{motor steps per revolution})$
- s= stall factor
- t= sample interval (steps)

Example:

$$\begin{aligned} \text{ratio} &= 10 \text{ (2000 count encoder/200 step motor)} \\ \text{s} &= 20 \\ \text{t} &= 5 \\ \text{Then lag} &= (10 * 20 * 5) / 256 = 3 \end{aligned}$$

The encoder will be sampled every 5 full motor steps (50 encoder counts, with t= 5), and the encoder must report at least 3 incremental encoder counts, or a stall error will be detected. The default values used above are adequate for detecting a stall condition with moves greater than 5 full motor steps. However, it will not detect a slipped pole, and it will not work with shorter moves. Therefore, you may wish to change these values to better suit your application.

Shorter moves require sampling the encoder more frequently than the distance of the move itself. That is, if a 3 step move is performed, then ‘t’ must be equal or less than 3. However, with moves this short, you may wish to just let the ‘d’, or dead-band logic correct for position errors.

Note: One slipped pole with this motor/encoder combination represents 10 encoder counts of error, but if the error is greater than 5, then it has slipped. This represents a ‘lag’ of 45 (5 counts short of a full 5 steps). Working algebraically backwards, we can establish “s” using:

$$s = (256 * \text{lag}) / (\text{ratio} * \text{t})$$

Upon stall detection, the controller can be made to re-establish synchronization, and continue with the move by programming the ‘r’, or Retry parameter.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
t	Stall Sample Rate	Immediate, Program, Default		3
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) t	0-255 Frequency		

(t) Stall Sample Rate

This command defines the sample rate or distance increment (in full motor steps) between stall tests. Reference the “s” command.

Example:

A 200 step/rev motor is coupled to a 500 line encoder, which yields 2000 encoder steps per revolution, or 10 encoder steps per motor step (2000/200). The sample distance (t) is 10 full steps. The encoder change each "t" step is 10 x 10 = 100 encoder steps. The actual sampling should allow for some backlash, as defined by the stall factor.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
v	Hunt Velocity	Immediate, Program, Default		3
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) v	0-8000 SPS		

(v) Hunt Velocity

This command specifies the hunt step rate (steps per second) to be used during dead-band repositioning. Motion is at a fixed resolution, specified by the "h" parameter. A value of zero disables stall detection.

<i>Command</i>	<i>Function</i>	<i>Type</i>		<i>NV Bytes</i>
z	Read Encoder Position	Immediate, Program		1
	<i>Mnemonic</i>	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i>
	(Name) z	0-1 Display Mode		Position ±8388607

(z) Read Encoder Position

This command will read and display the current encoder position. During motor move commands the value will change depending on the direction of travel. The position counter is reset by the "O" command. There is an option of continuous readout via the serial interface. The "z 1" command enables this operation. Any change in position causes the position data to be sent to the serial output. The readout is terminated by a Carriage Return only. The Readout mode will be defaulted “On” if a Save command is issued.

Electrical

AC Input Voltage..... 100 to 125 VAC, or 200 to 250 VAC, 50/60 Hz

	<u>CMAX-410</u>	<u>CMAX-810</u>
AC Input Power.....	90VA	180VA *
AC Fuse @ 120 Volts.....	1.6A	3.15A
AC Fuse @ 240 Volts.....	0.8A	1.6A
Output Current (Peak).....	4.0A	8.0A
Motor DC BUS**.....	40VDC	64VDC
DC BUS Fuse.....	3.0A	3.0A

AC Fuse Type: SB/IEC127-2
 DC Fuse Type: 2AG/SB – 250 Volts

*Maximum VA is determined by customer’s motor choice, parameter settings and speed. It is the customer’s responsibility to insure this VA value is not exceeded.

** DANGER! The motor BUS current may be present on the motor connector!

Applicable Directives

Environment: Industrial

- Standards: EN 55011:1991- Emissions Class A
 EN 61000-3-2:1995- Harmonic current emissions
 EN 50082-2:1995- General immunity standard
 EN 61010 Edition 2- Low Voltage Safety

Installation Category II
 Pollution Degree 2

All products are tested in accordance with Annex F, IEC-61010-1 Second Edition, 2001.

- F.1 Protective earth
- F.2 Mains circuits
- F.3 Other circuits

Microsteps Per Full Step..... Variable, 2, 4, 8, 16, 32, 64, 128, 256
 Encoder Resolution..... 50 to 12,750 Lines (in multiples of 50)
 Non-Volatile Memory..... 2k Bytes
 Position Counter..... ±8,388,607
 Baud Rate..... 9600 (Std.), 460k (Selectable)

Signal Specifications

I.O. Signals (J3)	Min	Typ	Max	Units
I.O. Supply (VIO)	5		28	Vdc
Inputs (Ports 1,2,3,4)				
Input Voltage	-0.7		VIO	Vdc
Input Current			500	µA @ 5V
Outputs, (Ports 4,5,6)				
Output Voltage			100	Vdc
Output Current			0.5	Amp (cont.)

Control Signals	Min	Typ	Max	Units
Encoder Supply Voltage		5		Vdc
Encoder Supply Current			50	mA
Encoder Inputs	-0.7		5.7	Vdc
Encoder Input Currents		5	10	mA
RS485 Inputs	-10		+15	Vdc
RS485 Outputs		5		Vdc

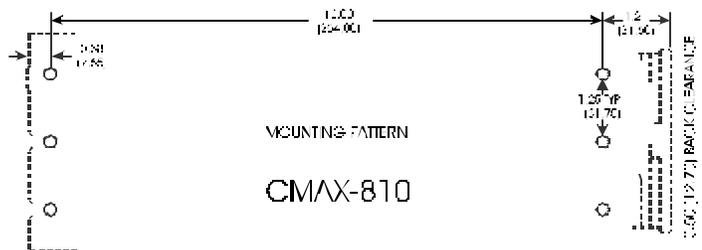
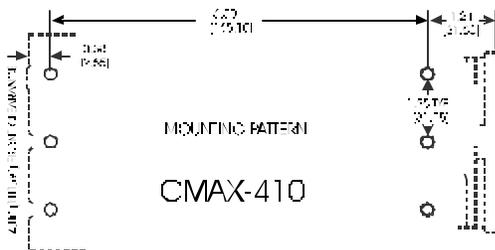
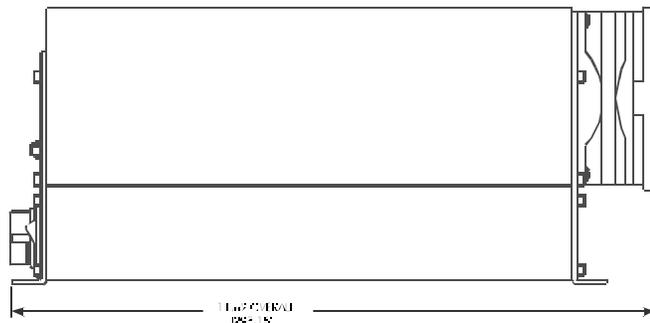
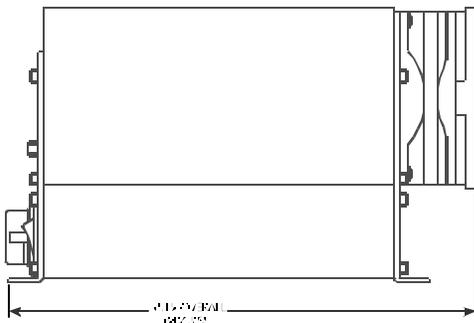
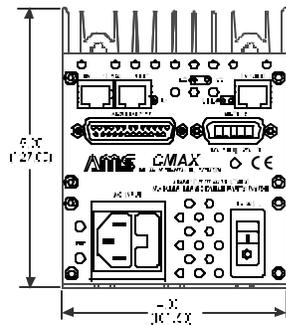
Environmental

Storage: -45 to 85 Degrees C
 Operating: 0 to 50 Degrees C
 Humidity: 0 to 95% (Non-condensing)

Physical

Weight:
 CMAX-410: 5.2lbs (2.4kg)
 CMAX-810: 9.4lbs (4.3kg.)

Size in inches (mm):



Command Summary

MNEMONIC / COMMAND	DATA 1	RANGE 1	DATA 2	RANGE 2	NV	D	I	P
PROGRAM COMMANDS:								
+	INDEX IN "+" DIRECTION	STEPS	0.01-8388607.99			5		⊙ ⊙
-	INDEX IN "-" DIRECTION	STEPS	0.01-8388607.99			5		⊙ ⊙
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	0-2					⊙ ⊙
^	READ MOVING STATUS							⊙ ⊙
	SELECTIVE TERMINATION				2			⊙ ⊙
^N	NAME CONTROLLER							⊙
^P	ENTER PARTY LINE MODE							⊙
A	PORT	BINARY	0-128			2		⊙ ⊙
B	SET JOG SPEEDS	SLOW	0-255	HIGH	0-255	3	⊙	⊙ ⊙
C	CLEAR AND RESTORE	PAGE	0-9					⊙
c	CLEAR DRIVER							⊙
D	DIVIDE RESOLUTION	RES.	0-8			2	⊙	⊙ ⊙
E	SETTLING TIME DELAY	0.01 SEC	0-255			2	⊙	⊙ ⊙
F	FIND HOME	SPS	20-23000	DIRECTION	0-1	3		⊙ ⊙
G	GO	ADDRESS	0-1791, 2048	TRACE	0-1	3		⊙ ⊙
H	MODE	FR/VR	0-2			2	⊙	⊙ ⊙
I	INITIAL VELOCITY	SPS	20-20000			3	⊙	⊙ ⊙
i	RESTART SPECIAL TRIP	NEXT TRIP	+8388607	PORT	0-63	5	⊙	⊙
J	JUMP	ADDRESS	0-1791	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	0-1791, 2048	CONDITION	0-8	4		⊙
l	LIMIT POLARITY	INVERT	0-1			2	⊙	⊙
M	MOVE AT CONST. VEL.	SPS	±22000			3		⊙ ⊙
N	SET POSITION	STEPS	+8388607.99			5		⊙ ⊙
O	SET ORIGIN					1		⊙ ⊙
P	PROGRAM MODE	ADDRESS	0-1791					⊙
Q	QUERY PROGRAM	ADDRESS	0-1791					⊙
q	QUERY (LIST) PROGRAMS	ADDRESS	0-1800					
R	INDEX TO POSITION	POSITION	+8388607.99			5		⊙ ⊙
S	STORE PARAMETERS							⊙
T	TRIP POINT	POSITION	+8388607	VECTOR	0-255	5	⊙	⊙
u	CONTROL LINE FEED		0-127			2		⊙ ⊙
V	SLEW VELOCITY	SPS	20-20000			3	⊙	⊙ ⊙
W	WAIT, (DELAY)	0.01 SEC	0-65535			3		⊙ ⊙
w	READ ERROR STATUS							
X/x	EXAMINE PARAMETERS							⊙
Y	RUN AND HOLD CURRENT	HOLD	0-100	RUN	0-100	3	⊙	⊙
Z	DISPLAY POSITION	CONTINUE	0-1					⊙
ENCODER COMMANDS:								
a	PRINT STALL COUNT							⊙
b	RESERVED							
d	DEADBAND ENABLE	COUNT	0-255			3	⊙	⊙
e	ENCODER RESOLUTION	LINE COUNT	0-2048					⊙
f	FIND ENCODER MARK	DIRECTION	0-1			2		⊙ ⊙
g	HUNT ON/OFF	DEADZONE	0-255			2		⊙ ⊙
h	HUNT RESOLUTION	MICRO. RES.	0-8			2	⊙	
m	INDEX MODE		0-1			2	⊙	⊙ ⊙
n	FORCE ENCODER POS.	COUNT	+8388607			1		⊙
o	RESET ORIGIN TO ZERO					1		⊙
r	SET STALL RETRY COUNT	RETRIES	0-255			2	⊙	⊙
s	STALL DETECT	STALL	0-255			3	⊙	⊙
t	STALL TEST DELTA	STEPS	0-255			3	⊙	⊙
v	HUNT VELOCITY	SPS	0-8000			3	⊙	⊙
w	DRIVER STATUS							
x	EXAMINE PARAMETERS							
y	CARRIAGE RET/LINE FEED							
z	READ ENCODER POSITION	CONTINUE	0-1					

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	

Firmware Revision History

V1.22 (02/18/97)

Software version 1.22 (02/18/97) implements “watchdog” functions to improve reliability in harsh environments. Two enhancements are provided:

1. The CMAX units incorporate a special timer. This timer is continuously running and counting down to zero. Periodically, the CMAX “reloads” the timer and the count down process is restarted. Should the timer reach zero, because the microprocessor has “crashed” for some reason, the timer will set a flag that causes an unconditional reset of the system.
2. A “Power Fail Interrupt” feature has been added that will detect a logic power supply voltage that is out of tolerance. During power down, or a severe power glitch, the processor will be held at a reset condition until the proper voltage (5V) is restored. This event is also recorded by setting a flag and is also readable by the host computer.

Operation

Upon detection of a fault, flags are set and a CPU restart is implemented. All events that take place during power up, except for reset of the flags, are initiated:

1. Motion and running programs halt.
2. All position counters are reset.
3. Stored NV parameters are reloaded.
4. All ports are set to an “off” state.
5. The encoder controller is reinitialized.
6. Go and Jog inputs are enabled.

In single axis mode a space character must be used to sign on. The sign-on message will contain the “!” character.

In the Party Line mode the unit assumes the ready condition but will echo the “!” in response to the name address. At this point the host computer should issue the “[] 1” command to clear the watchdog flags and resume normal operation.

Note: any command can be executed, even though the “!” is echoed. Once the flags are reset the assigned name will be used.

The watchdog flags (time-out and brown-out) are reset under only one of two methods:

1. By a hard reset such as at power up
2. Using the “[] 1” command

V1.23 (12/15/98)

Two new commands were added that allow control of encoder servo and absolute position calculation methods. The added “g” command triggers encoder hunt (servo) for position maintenance and the added “m” command modifies the position computation method. See full descriptions in the appropriate manual.

V1.24 thru 1.27 were minor bug fixes.

V1.28 (12/01/00)

A new feature (H2 command) was added that makes it possible to convert two output ports to step and direction output signals. This can be used to control additional “slave” drives.

The command H 2:

- A. Directs the step pulse to port 5 and direction to port 6,
- B. Places the controller in fixed resolution mode,
- C. Disables port commands that can affect ports 5 and 6.

V1.29 (11/15/01)

Added the following commands:

- Lower case “k”
- Lower case “i”
- Lower case “u” (replaces lower case “y”)

V1.30 (01/10/04)

Added virgin NV memory test on power up, ^N name assignment and ^P Party Line mode.

V1.31 (01/10/05)

Alternate software (axis) naming and party line mode commands were added. Both commands function only while in single axis mode.

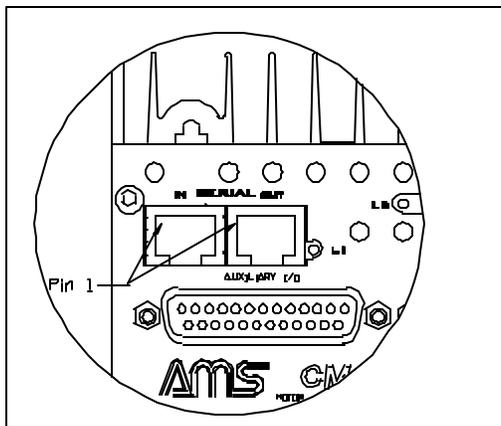
- ^N (control N) to name axis
- ^P (control P) to immediately enter party line mode

Compatibility with older versions is still present so the sign-on with separate “Save” command method can still be used.

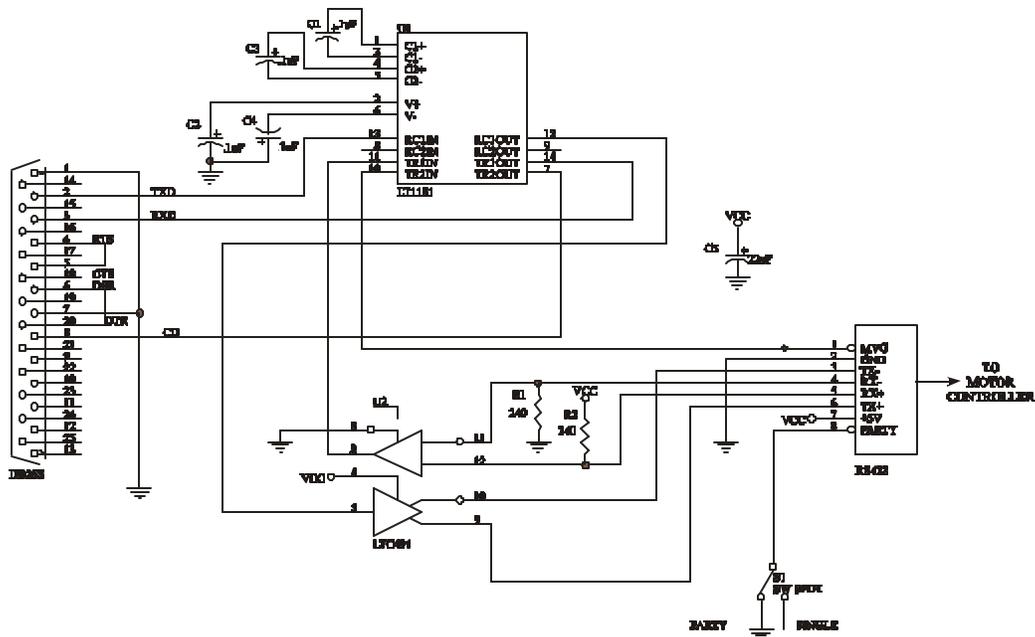
Serial Interface Connections

Serial Interface Connector (RJ45) On Serial adapter			Serial Interface Connector (J1) on CMAx-410/810		
Pin	Signal	Type	Pin	Signal	Type
1	Moving*	Output	1	Moving	Output
2	GND	Logic	2	GND	Logic
3	TXD-	Output	3	RXD-	Input
4	RXD-	Input	4	TXD-	Output
5	RXD+	Input	5	TXD+	Output
6	TXD+	Output	6	RXD+	Input
7	+5v	Logic	7	+5v	Logic
8	Party Enable	Input	8	Party Enable	Input

*The Moving signal is an open collector buffered output used on the SIN-8 only. The SIN-8 converts this signal to RS-232 level and supplies it to Pin 8 “Carrier Detect” of the 25 Pin “D” connector.



SIN-8 Schematic



About Step Motor Current

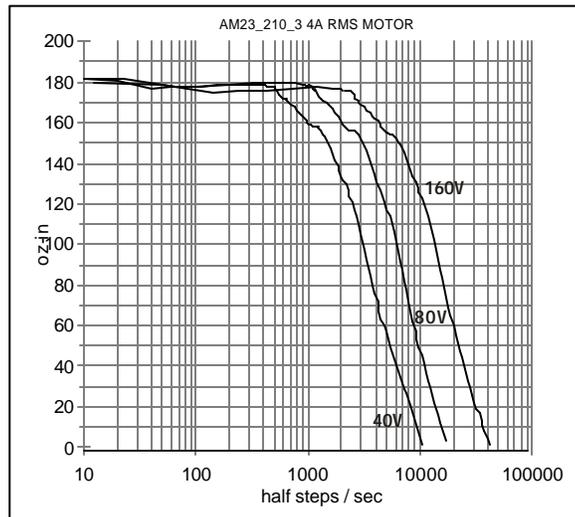
There is much confusion regarding the operation of step motors. Depending on your application, the step motor offers several advantages over servo motor designs, including lower cost and simplicity. The step (or stepper, or stepping) motor is a digital “synchronous” motor with a pre-designed number of “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.

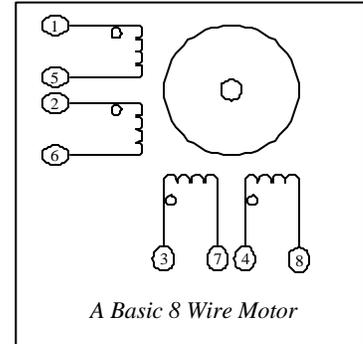
Let us assume that each winding of the 8 wire motor has the following specifications:

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

The power per winding is:

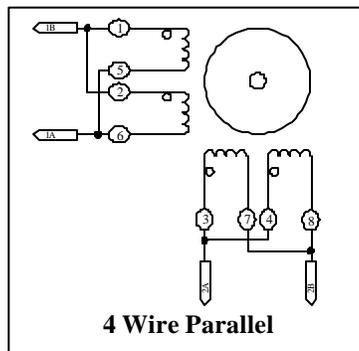
$$I^2R \text{ or } 2 \times 2 \times 1 = 4 \text{ watts,}$$

$$\times 4 \text{ coils} = 16 \text{ watts total for this motor.}$$



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

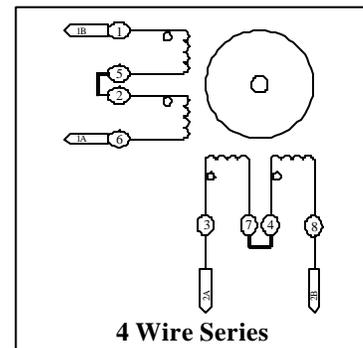
The high-speed model implements parallel coil connection. Two coils connected in parallel result in the following for each of the two phases:

- 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

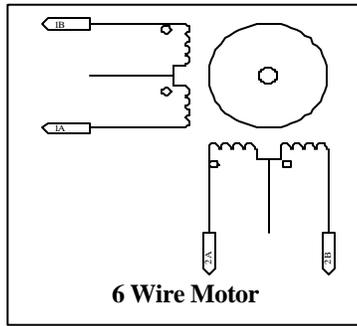
B. 4 Wire Series

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.



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)

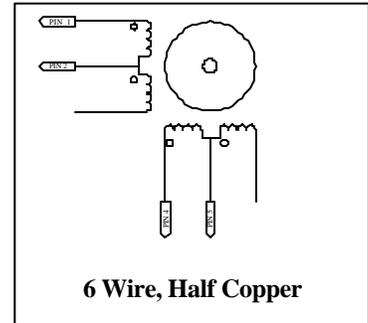
In practice the two coil ends are connected, while no connection is made to the center tap.

Half Copper or 50% Winding

The maximum speed can be increased by using 1/2 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.



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

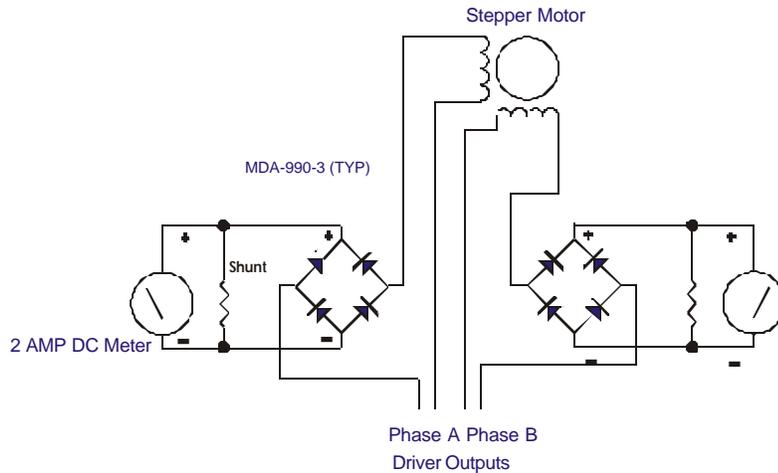
# Leads	Winding Connection	Per Phase Bipolar Current	
		Full Step (Total)	Microstep RMS (Peak)
4 wire	Parallel	4.0A (8.0A)	4.0A (5.66A)
4 wire	Series	2A (4.0A)	2.0A (2.8A)
6 wire	Unipolar	4A per phase	NA
6 wire	Series	2.8A (5.6A)	2.8A (4A)
6 wire	50% Copper	4A (8A)	4A (1.7A)

- 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 1.414×2 (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 ($\frac{1}{4}$ 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 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-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.