

***MAX-410 & MAX-420
USERS GUIDE***

Revision Date: 07/09/2013

1) <i>Introduction</i>	5
Congratulations!	6
2) <i>Installation</i>	7
Required Hardware	8
Front Panel Description (MAX-410)	8
Hardware Set-Up	9
Fault Protection.....	9
Diagnostic LED's	10
Software Set-Up	10
Sign-On.....	10
Axis Naming.....	11
Programming	12
Party Line Startup (Using a SIN-11 Serial Adapter)	14
Program Considerations.....	14
Communication Flow Chart	15
ASCII Baud Rates	15
3) <i>Communication Interface</i>	16
RS-422 Hardware	17
Other Party Line Signals.....	17
Adapters.....	18
Communication Modes.....	19
Communications Software.....	20
4) <i>Auxiliary IO</i>	22
J3 and J5 Connections	23
J5 Auxiliary I.O. Schematic.....	24
5) <i>Operating Parameters / Memory</i>	25
Reset/Initialize	26
Input Command Example	26
Result Data	26
Party Line Commands	27
Instruction Execution.....	27
Interrupt Commands	28
Watchdog Functions (Optional).....	28
Soft Stop “@”	29
Soft Stop Interrupt	29
Jog Speeds, Homing	29
High Speed Considerations.....	29
Related Items: Trip and Loop.....	30
Non-Volatile Memory	30
Scratch Pad/Shadow Memory.....	31
Command/Program Mode	31
Editing Programs	31
6) <i>Encoder Feedback</i>	33
Encoder Hardware	34

Encoder Schematic	34
Design Notes.....	35
Testing Operation	35
Encoder Operation.....	35
Indexing With Encoders	36
Encoder Indicators.....	37
 7) <i>Commands</i>	 38
Program Command Description	39
(ESC) Global Abort.....	39
(@) Soft Stop.....	39
(^C) Reset	40
(A) Port Read/Write.....	40
(B) Set Jog Speeds.....	41
(C) Clear and Restore	41
(c) lower case c; Reset Driver.....	42
(D) Divide Resolution.....	42
(E) Settling Time Delay.....	43
(F) Find Home	43
(G) Go.....	44
(H) Resolution Mode	45
(I) Initial Velocity	46
(i) lower case I; Restart Special Trip See lower case “k” command.....	46
(J/j) Jump	47
(K) Ramp Slope	47
(k) lower case K; Trip Output Value	48
(L) Loop on Port.....	49
(l) lower case L; Limit Polarity.....	50
(M) Move at Constant Velocity	50
(N) Force Step Position	51
(O) Set Origin	51
(P) Program Mode	51
(Q) Query Program	53
(q) Query Program (20 Lines)	53
(R) Index Relative to Origin.....	53
(S) Store Parameters	54
(T) Trip Point.....	55
(u) lower case u; Control Line Feed	56
(V) Slew Velocity	56
(W) Wait.....	56
(w) Driver Status	57
(X/x) Examine Parameters.....	57
(Y) Hold and Run Current	58
(Z) Read Position.....	59
(I) Read Non-Volatile Memory	59
(J) Read Limits/Hardware/Watchdog Timer.....	59
(+) Index in Plus Direction	60
(-) Index in Minus Direction.....	61
(^) Read Moving Status	61
(\) Write to Non-Volatile Memory.....	61
(/) Selective Termination	62
^N (Name Axis).....	62
^P (Party Line Mode)	63

8) Encoder Commands	64
(a) Read Stall Count	65
(d) Set Deadband and Enable	66
(e) Enable Encoder	67
(f) Find Encoder Mark	68
(g) Hunt Mode	68
(h) Hunt Resolution	69
(m) Index Mode	69
(n) Force Encoder Position	70
(o) Set Origin to Zero	70
(r) Set Stall Retry Count	70
(s) Stall Detect	71
(t) Stall Sample Rate	72
(v) Hunt Velocity	72
(z) Read Encoder Position	72
9) Specifications	73
Electrical	74
Environmental	75
Physical	75
10) Addendum	77
Command Summary	78
ASCII Character Code	79
Firmware Revision History	80
About Step Motor Current	82
AMPS and Wire Count and Power	84
Set-up for Current Calibration	86

1) Introduction

Congratulations!

.....on your purchase of a MAX-410 (Single Axis) and/or MAX-420 (Dual Axis) Stepper Motor Control System with "VRMC[®]" (Variable Resolution Microstep Control). The MAX will provide years of reliable, accurate and cost-effective motion control. As with all AMS products, the DCB-241 is backed by over three decades of manufacturing excellence and a commitment to quality and support that guarantees your satisfaction.

This Users Guide will assist you in optimizing the performance of your MAX system. Its purpose is to provide access to information that will facilitate a reliable and trouble-free installation. The Users Guide is organized into the following sections: Installation, Communication Interface, Auxiliary I.O., Operating Parameters, Encoder Feedback, Commands, Encoder Commands, Specifications and Addendum. We recommend that each section be reviewed prior to installation.

Although the MAX and supporting documentation were designed to simplify the installation and on-going operation of your equipment, we recognize that the integration of motion control often requires answers to many complex issues. Please feel free to take advantage of our technical expertise in this area by calling one of our support personnel at (603) 882 1447 to discuss your application.

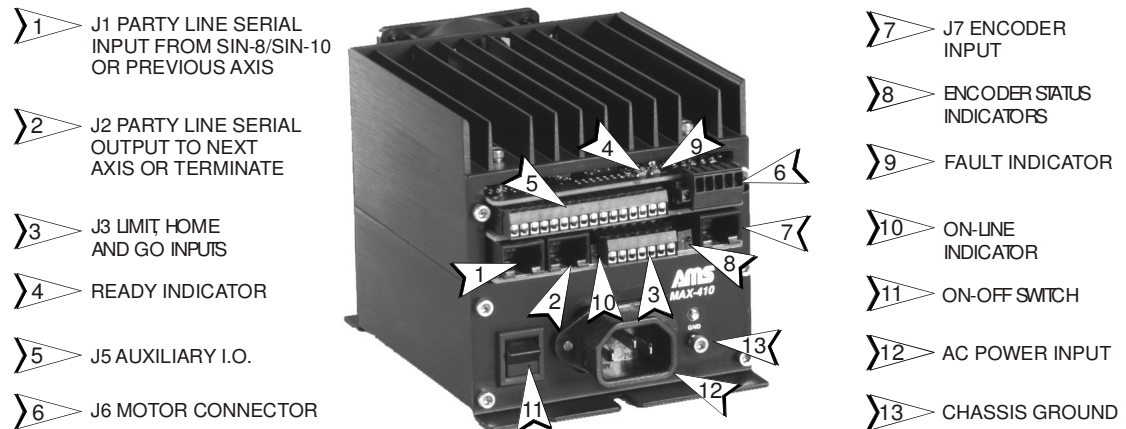
Thank You!
Your AMS Team

2) Installation

Required Hardware

Qty	Unit	Model #	Description
1	Axis	MAX-410/420	Driver-Controller assembly
1	System	SIN-9 OR	RS-232 serial adapter (single axis)
1	System	SIN-11	Intelligent serial adapter and cable
1	Added axis	BLC-51-3	Interconnect cable, Cat5 (3 ft.)
1	System	TERM-2	Terminator plug (included with SIN-11)

Front Panel Description (MAX-410)



Status Indicator Light Description

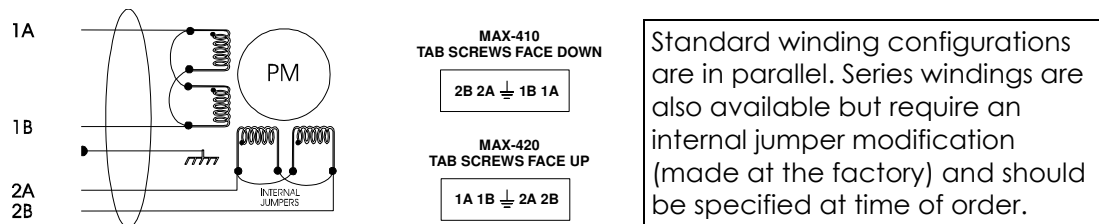
#	Label	Description
1	EA (Yellow)	Encoder channel "A" input (behind ER, ST).
2	EB (Yellow)	Encoder channel "B" input (behind ER, ST).
3	EC (Yellow)	Encoder channel "C" input (behind ER, ST).
4	ER (Yellow)	Direction. "ON" when indexing in the "-" direction.
5	ST (Red)	Stall. "ON" when motor position differs from encoder position deadband or when in "low seek" mode.
6	OK (Green)	"ON" when serial communications has been established.
7	FAULT (Red)	See "Fault Protection" below.
8	RDY (Green)	See "Fault Protection" below.

Hardware Set-Up

Prepare the MAX for operation by referring to the illustration on the preceding page and by following these simple instructions:

- Step 1** Connect the DB9 end of the SIN-9 or SIN-11 to an available COM port of your computer.
- Step 2** Connect the other end of the SIN-9 cable assembly (looks like a LAN connector), to the mating connector J1 "Party Line Serial Input", on the front panel of the MAX.
- Step 3** Install a terminator plug (TERM-2) into J2 "Party Line Serial Output" of the last axis. The controller will work without a terminator plug, however reliability may be compromised, based on the environment (noise) and length of 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 MAX and to a correct AC power source (120V 50/60 Hz or 240V 50/60 Hz depending on the settings of your unit; see serial number sticker).
- 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, then remain on.
 3. The ON-LINE indicator light (labeled OK) should be off.

Note: If the ON-LINE indicator light is on, verify that the switch on the SIN-8 is in the single (S) mode position.

Fault Protection

The MAX 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 MAX is packaged in a specially designed heat sink 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 drivers 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. The Red LED should normally be off.

Faults

A fault condition always involves the Red LED:

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

* Latched; requires power cycle reset to clear.

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

Software Set-Up

If you are using a

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

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

- Step 1* Once you have selected and - if needed - installed the communication software, make sure that the active COM port matches the COM port to which the SIN-9/SIN-11 is connected.
- Step 2* Ensure that the port parameters are selected as follows: bits/s: 9600, data bits 8, parity: none, stop bits:1, flow control: hardware.

Sign-On

- Step 3* Ensure that the MAX-410/420 is powered up. For the next steps, if using the AMS Cockpit software, you need to make sure that you are in the Dumb Terminal mode and not in the Party Line Mode. Dumb Terminal mode is the default upon start-up. See the Section Communication Modes for more details.
- Step 4* Strike the SPACE BAR key. The controller should sign on with a message including “Advanced Micro Systems, Inc.” as well a version number (Vx.xx). If not, enter a (^C) (reset) and strike the SPACE BAR key again.

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

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.

Step 5 At this point it is recommended to set the Hold and Run current. These can be adjusted using the Y command. By default the MAX products are shipped with a 5% hold current setting and a 25% run current setting.

Set the defaults:

- A. Enter the “Y30 50” command (if you would like to adjust the hold current to 30% and the run current to 50% or use whatever value is appropriate for your application),
- B. Enter the “S” command (save default to NV memory).

Axis Naming

Each axis should be assigned a unique “name” for proper operation when used in Party-Line mode.

Step 6 Reset the controller (^C), then type a single, valid name character:

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

Step 8 Verify the Name by entering the “X” command (UPPER CASE) <CR>. The last item of the first line should contain the “Name” character.

Step 9 Enter the Save command (S) (UPPER CASE) <CR>. The axis Name is now stored in non-volatile memory.

The MAX is ready to operate in the current single axis mode or be switched over to Party Line mode. It is suggested that the operator use single mode first to become familiar with command input. The single axis mode can be used with any “dumb” terminal device and is not dependent on using the AMS software.

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

Examine Command

The Examine command (X) (UPPER CASE) <CR> causes the MAX to display a set of parameter values that were last stored in non-volatile memory. These parameters may be modified using the appropriate commands, then stored in non-volatile memory as the new “defaults”.

Default Parameter	Value
Y	5/25
E	50
K	10/10
H	vr
na	
I	400 (12800/32)
V	3004 (12047/4) (rl=15)
e	500 (ratio=10)
d	30
v	700/16
s	20
t	5
r	5 (lag=3)

Where:

Y= Hold/Run Current %
 E= Settling Time Delay
 K= Ramp Slope (ramp up/ramp down)
 H= Resolution Mode
 na= Axis Name
 I= Initial Velocity
 V= Slew Velocity

Encoder Mode Only:

e= Encoder Resolution
 d= Deadband
 v= Hunt Velocity
 s= Stall Factor
 t= Stall Test Delta
 r= Set Stall Retry Count

Only two lines are displayed if the encoder mode (a zero for the “e” command) was not setup.

Several other commands may be used to check the operation.

“A129”: Read ports (0).
 “w”: Read driver status (0).
 “j”: Read limits (0).

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

Example:

1. Issue the command: “R - 1000”<cr>. The motor should move.
2. Issue the “Z”<cr> command. The position (-1000) should be displayed.

Some Rules

1. The command line may be edited using back-space 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.

Programming

The above examples were samples of immediate commands. The following is a sample of the sequences that are stored in non-volatile memory. Note that when programming the sequence is immediately written to non-volatile memory, without any additional action required to save it.

Example:

	<u>Enter</u>	<u>Remark</u>
	P0<CR>	Place in Program mode. Insert instructions at location 00.
<u>Address</u>		
0	00<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.

Now 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.

Verify the Program

The MAX 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		

Execute the Program

<u>Enter</u>	<u>Remark</u>
G0 1<CR>	The MAX starts executing the program starting at location zero. If the Trace option is on, the MAX will display each instruction, prior to execution.

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

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 diskette, which includes a sophisticated Editor and Compiler.

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 (1s and 0s) 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. Motion commands with 2 numbers require at least one space between numbers. Motion command characters are issued as UPPER CASE, while encoder command characters are lower case.

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

Party Line Startup (Using a SIN-11 Serial Adapter)

After assignment of a unique axis name to all axes that you are intending to connect to one port of a computer, connect all axes in a chain by connecting the J1 Serial Input of the second axis to J2 Serial Output of the first axis, and so on. Make sure to insert the TERM-2 termination plug that is delivered with the SIN-11 into the J2 of the last axis that remains unconnected.

Now power up the devices. If using

- AMS Cockpit, then enter Party Line Mode by clicking on the “Start Party” button on the right hand side of the screen. The system will now scan for all legal axis names and will list them on the screen after the completion of the scan. Also, the background of the text window will change from yellow to blue symbolizing that you are now operating in Party Line mode.
- An alternative terminal program such as HyperTerminal, issue the “&” symbol. The system will perform the axis scan and list the names of all axes found. Once the scan is complete it will issue a line feed.

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

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).

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).

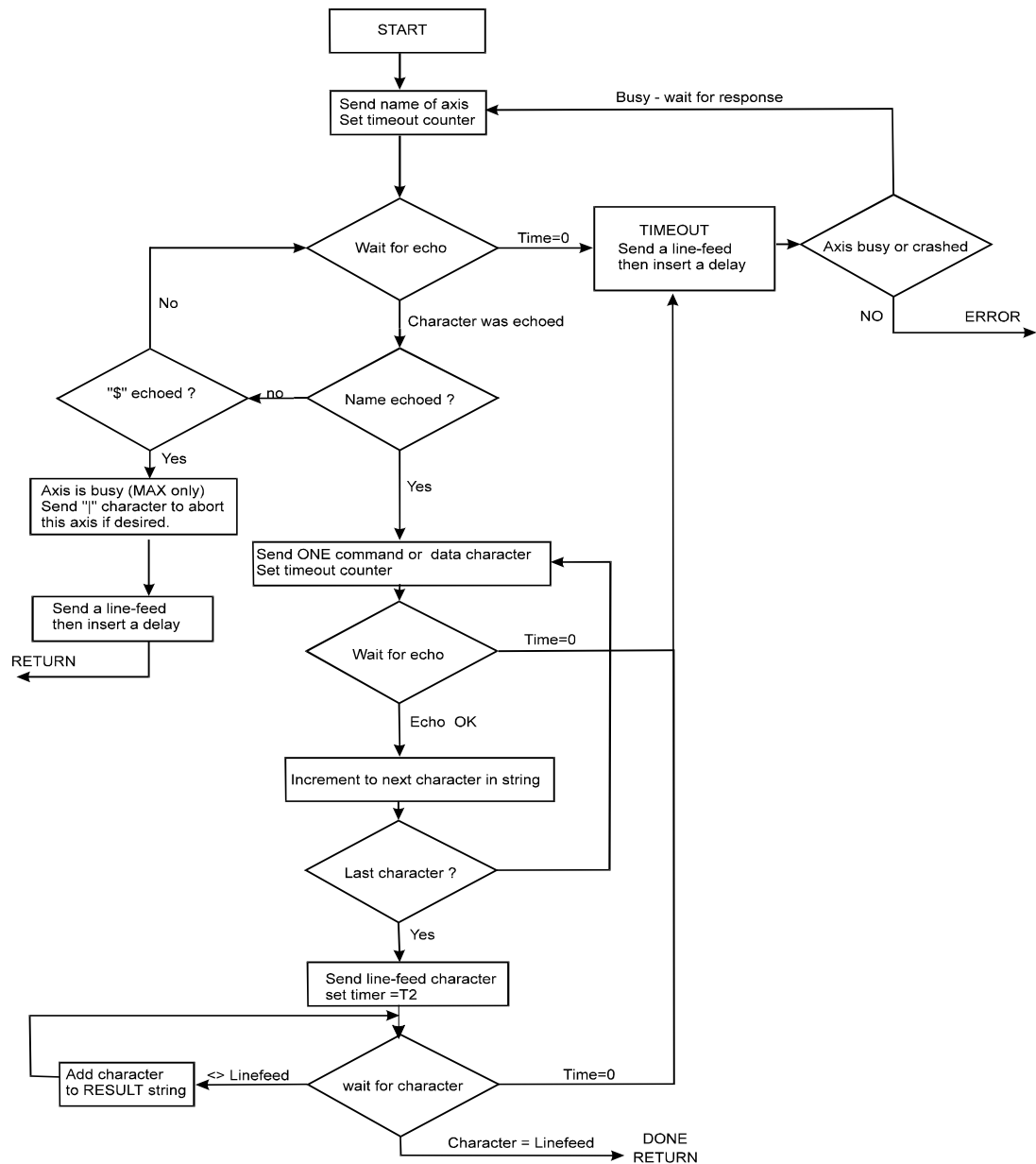
The echoed characters provide reliable handshaking and a means of receiving data from the Slave, i.e., a “Z” position register command.

Program Considerations

The MAX 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 character 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.

Communication Flow Chart**ASCII Baud Rates**

Baud rates for MAX communication is fixed at 9600 bauds per second. Consult AMS if an alternate communication rate is required.

Host/terminal settings:

Start Bits	Data Length	Stop Bits	Parity
1	8	1	None

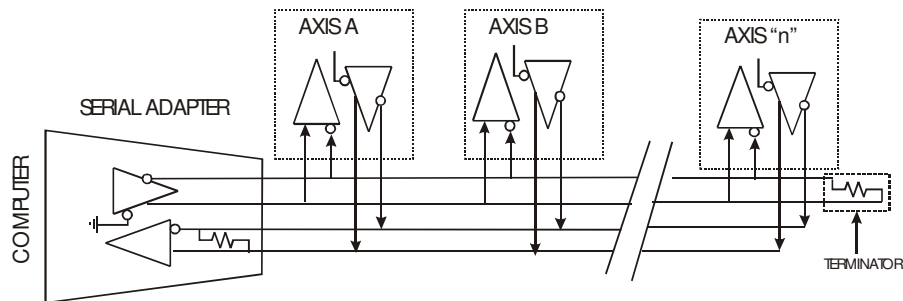
3) Communication Interface

RS-422 Hardware

AMS communication protocol is an RS-422 design that uses RS-485 rated circuits. This interconnect is comparable to a LAN configuration. The hybrid design merges the best of both EIA specifications and maintains compatibility with EIA RS-422 and features:

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

RS-422 Connect



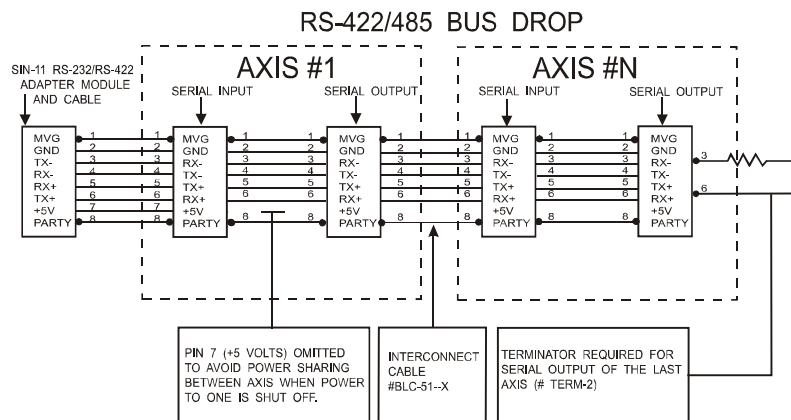
Communication hardware requires three components:

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

Other Party Line Signals

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

1. **GND** (pin 2) is common for all devices (controller). All power supply commons are connected to prevent high common mode voltages. Please note that the power common is generally connected to the case return.
2. **+5 Volts** (pin 7) is available to power the serial adapter from the first controller.
3. **Party Select** (pin 8) is used for other products that require this input.



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

Adapters

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

SIN-9 Passive Adapter

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



SIN-9, Serial Adapter

SIN-11 Intelligent Serial Adapter (Recommended)

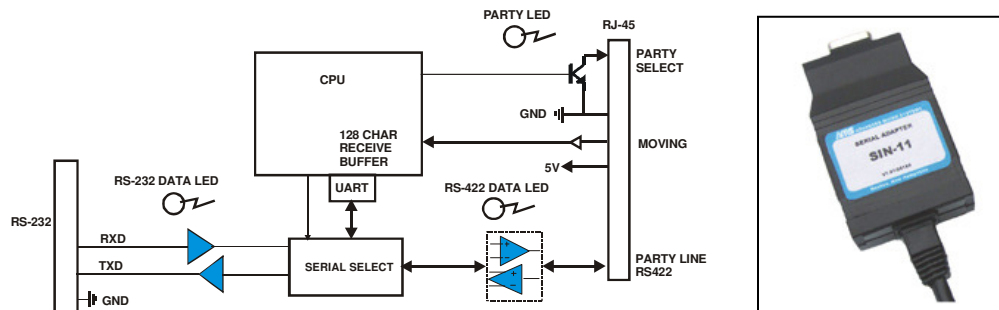
The SIN-11 is an intelligent serial line converter that simplifies application software development and improves overall performance. Communication between connected controllers and the SIN-11 is 9600.

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

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

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

Because the SIN-11 eliminates the need for special echoed character software it can be used in Windows applications where either the machine or software is slow and/or the operating system prevents direct programming of input or output instructions.



SIN-11, Intelligent Serial Line Converter

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

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

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

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

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

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

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

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

Communication Modes

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

1. “Dumb” Communications Mode

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

Serial adapters used: SIN-9 or SIN-11

At start-up:

1. Hit the SPACE BAR key to sign on.

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

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

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

2. Party Line Mode

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

Between 1 and 32 axis are configured as “slaves.” Unlike the “Dumb” mode, a proper character by character echoed protocol is necessary for proper operation. The SIN-11 adapter simplifies this protocol.

Serial adapter used: SIN-11

At start-up:

1. Issue an “&” command to enter Party Line mode.
2. The host computer interrogates and records axis name(s).

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

3. Daisy Chain Mode (not recommended for more than 1 axis)

This older protocol is similar to the party line mode but RS-232 protocol is used. Because it involves special wiring of RXD to TXD signals, it should only be used with a single axis design. When multiple axes are implemented they are less reliable, communication speeds are slower and troubleshooting is difficult.

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

Serial adapter used: SIN-9

At start-up:

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

Communications Software

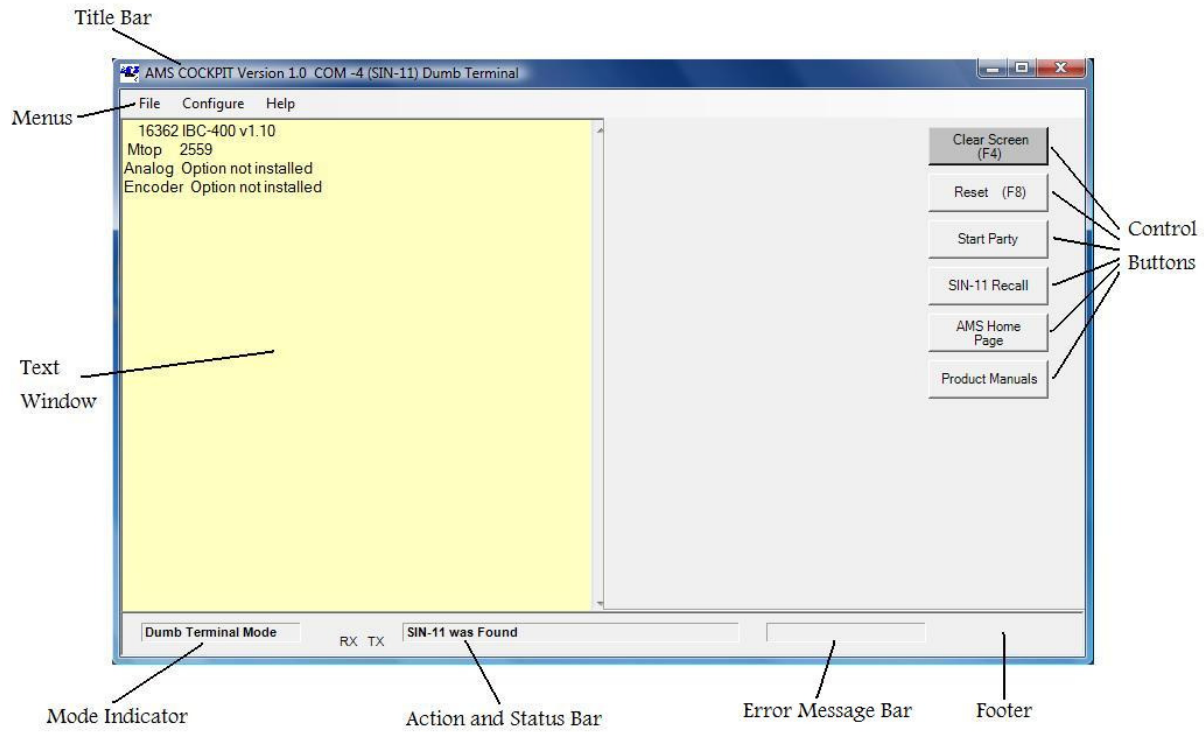
AMS offers the “AMS Cockpit” software that can be downloaded for free from the AMS website to assist customers in the implementation of their projects. It is compatible with Windows Operating systems. In addition to enabling communication, it includes some customized functionality, such as the downloading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

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

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

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

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



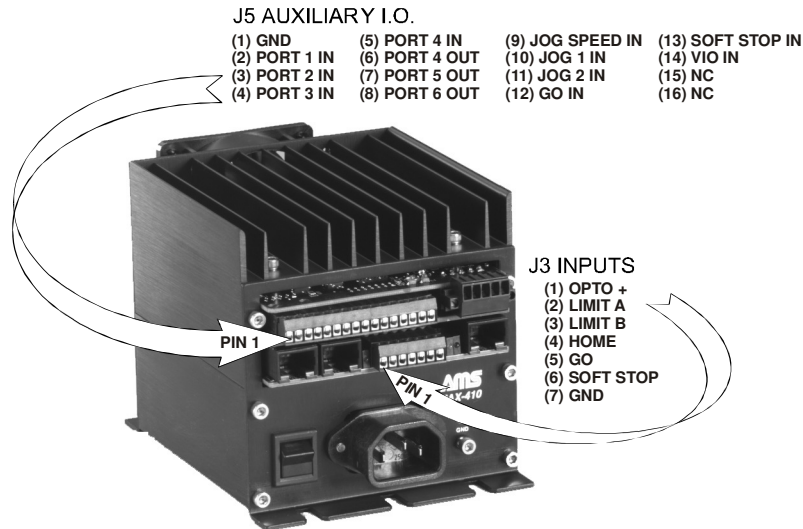
Main Screen of AMS Cockpit Software

4) Auxiliary IO

J3 and J5 Connections

J3 I.O. Interface

The MAX uses two connectors for the I.O. interface; J3 and J5. The J3 connector contains the “essential” motor related signals. All J3 inputs are optically isolated with a 5 volt internal LED source. The opto+ (pin 1) input permits other optical supply voltages to be used, providing additional current limiting resistors are used on the inputs. An internal pull-up resistor to 5 volts (opto+) is provided to force inputs normally high. To be activated they need to be pulled low.

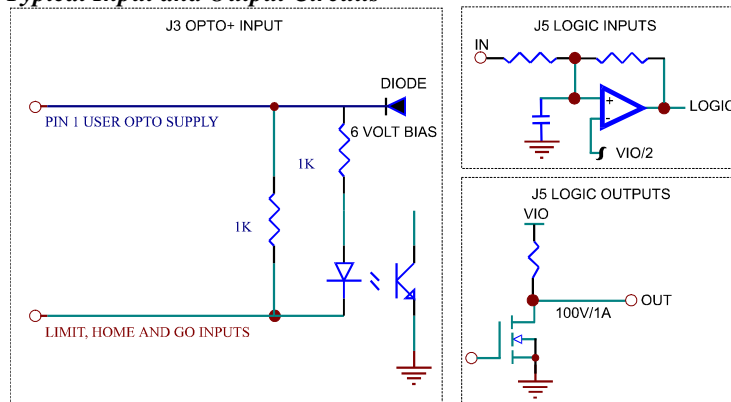


J5 Auxiliary I.O.

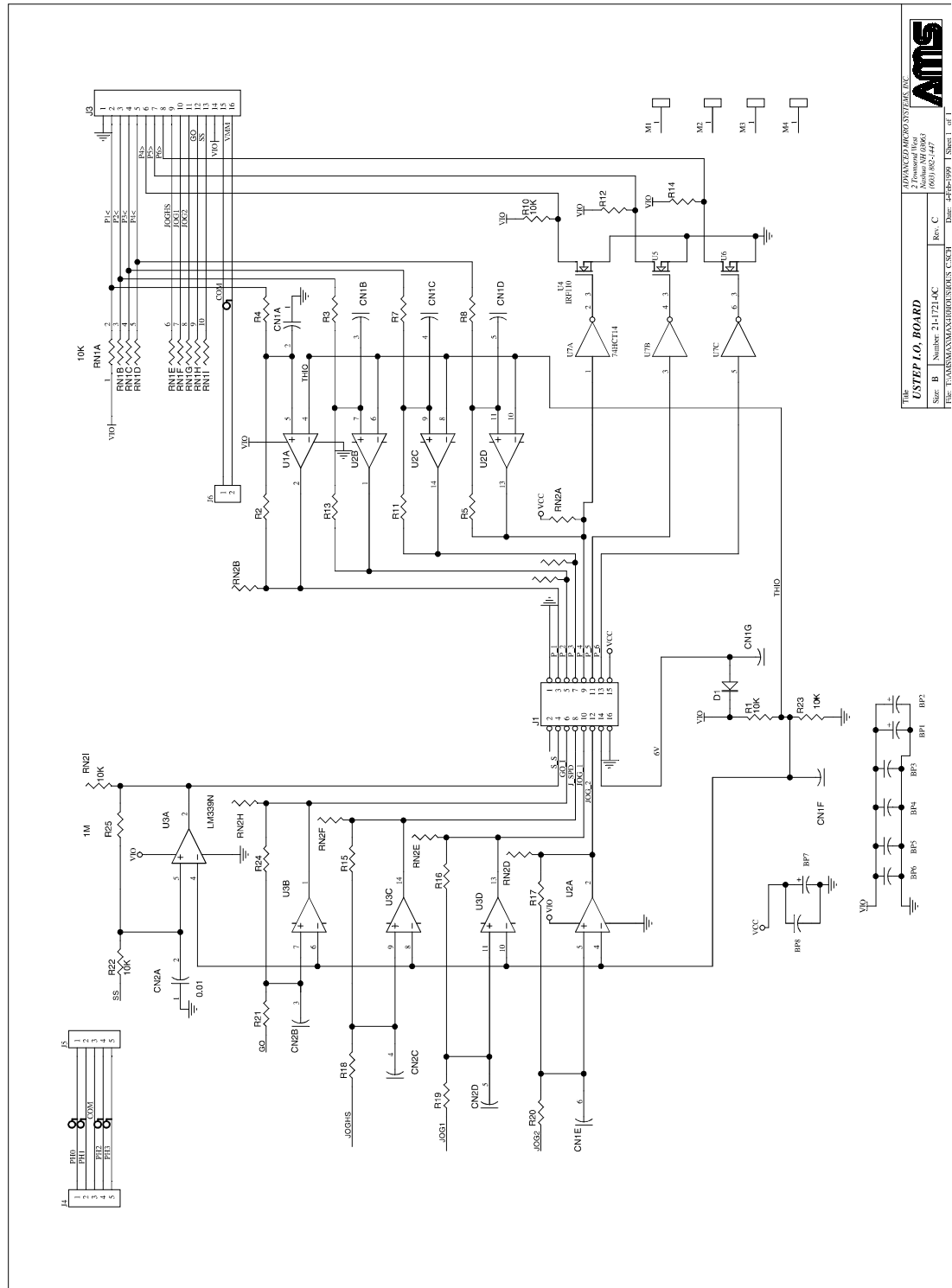
This connector contains nine inputs and three outputs. All inputs have RC noise filters, followed by voltage comparators with hysteresis, for high noise immunity. The input threshold voltage is set to 50% of VIO, the input circuit supply voltage. VIO is supplied from the 6 volt bias supply via an isolation diode and is approximately 5.4 volts. An external supply of up to 30Vdc may be used if a different input voltage range is required.

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.

Typical Input and Output Circuits



Note: NEVER attempt to draw current from the OPTO+ pin.

J5 Auxiliary I.O. Schematic

5) Operating Parameters / Memory

Reset/Initialize

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

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

Parameter	Standard Defaults
Initial Velocity (I)	400 (steps per second)
Slew Velocity (V)	3004 (steps per second)
Divide Factor (D)	1
Ramp Slope (K)	10/10
Jog Speeds (B)	30, 200
Trip Point (T)	Off
Step Resolution (H)	1 (auto variable)
Auto Power Down	Yes
Hold/Run Current (Y)	5, 25
Limit Polarity	Low
Auto Position Readout (Z)	Off
Name (after reset)	Undefined

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

Input Command Example

Command	Single Axis Mode Operation
+1000 (Carriage Return)	Step 1000 steps in + direction
+ 1000 (Carriage Return)	Same as #1
H 0 (Carriage Return)	Set fixed step resolution
H0 (Carriage Return)	Same as #3
H (Carriage Return)	Same as #3(0 is used by default)
R -1000 (Carriage Return)	Move to position -1000

Commands such as Jump and Loop instructions are only valid when used in the Program then Execute mode.

The following can only be executed from programs stored in optional non-volatile memory:

J 0 5 (Carriage Return) Jump to location 0, 6 (n + 1) times.

J0 5 (Carriage Return) same as above.

Result Data

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

Party Line Commands

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

Instruction Execution

For each Motion command there are four cycles:

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

Other commands have three cycles:

1. Entry
2. Execution
3. Result

In the idle state the MAX 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 Carriage Return (^J or ^Enter for Party Line) terminates the Entry cycle and initiates execution.

Cycle 2. Execution

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

Cycle 3. Result

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

Cycle 4. Completion

The completion phase is required for any Action command cycle.

The following are Action commands:

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

During the Completion cycle (except for Go) any non-action command, such as read position, may be executed. The MAX 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.

Interrupt Commands

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

Abort

Any Action command may be terminated using the ESCape command.

Process	Resulting Action
Command line input	Clear input buffer
Program mode	Exit without inserting "End"
Action command	Terminate all motion Hard Stop
Program execution	Terminate execution Hard Stop

Note: All processes are aborted upon ESCape.

Watchdog Functions (Optional)

The watchdog functions help to improve reliability in harsh environments (reference “J” 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.

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: *In any mode jogging and command reception are mutually exclusive. A command can NOT be loaded while jogging and jogging can NOT be performed until the last command is complete. A command starts with the reception of the first command character.*

High Speed Considerations

The MAX 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 MAX 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 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 MAX 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 EEPROM. The MAX 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.

Command/Program Mode

In the Command mode, commands are normally executed as soon as they are entered. The use of non-volatile memory allows storage of a list of commands. These stored program(s) can be triggered at power-up for automatic or repetitive operations by 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. When in the Program mode, the entered commands (now called instructions) are directed into the non-volatile memory. After leaving Program mode, the stored program(s) may be subsequently executed by entering the “G” Go command.

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

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

Once in the Program mode the current memory location is displayed on the terminal and instructions may be entered. As each instruction is entered, the location is displayed. All instructions have the same format as in the Command mode. Terminating the Program mode is done by entering a “P.” This will cause the end of program flag to be inserted and the controller will echo the pound (#) character. The MAX will return to the Command mode.

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.

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 then 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.

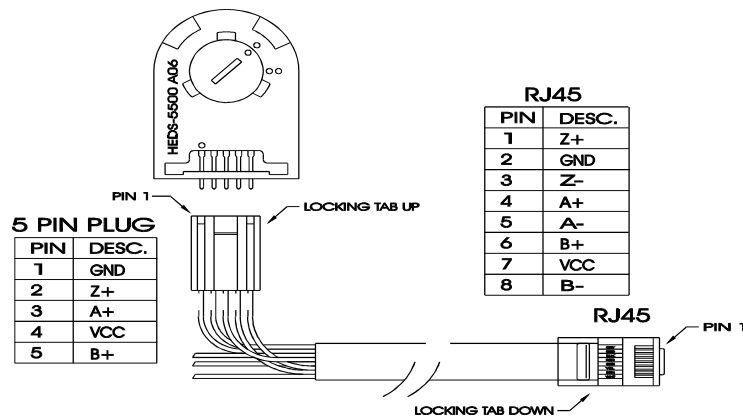
6) Encoder Feedback

Encoder Hardware

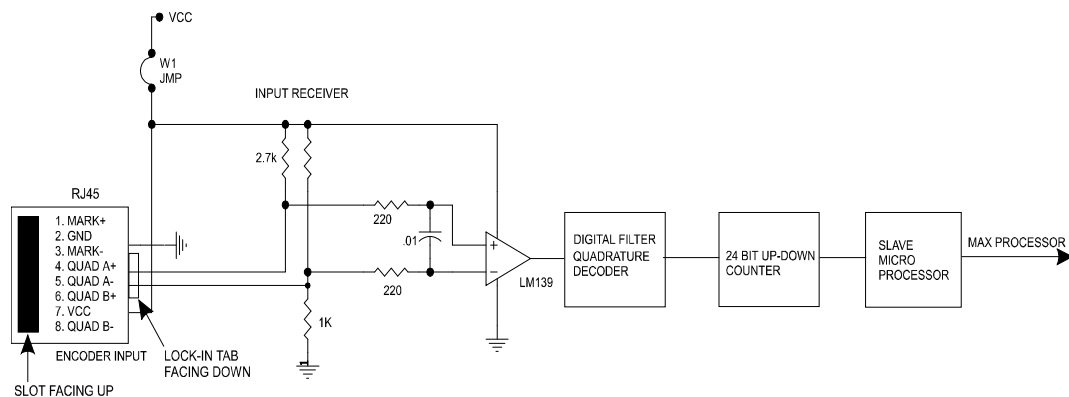
The MAX is available with encoder feedback. This permits use of an optical encoder to provide position feedback. The following elements are used to provide this function:

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 MAX CPU.

Wiring Connections



Encoder Schematic



Most applications are reliable using the single-ended inputs (MARK+, QUAD A+ and QUAD B+). The standard 5volt 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.

Testing Operation

- 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 MAX 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 result 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., $(s = (\text{encoder count/steps per rev.}) \times \text{fudge factor})$.
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 $50 \times 20/255 = 3$ 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

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).

7) Commands

Program Command Description

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

Where:

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

<i>Command</i> ESC	<i>Function</i> Terminate Operation		<i>Type</i> Immediate	<i>NV Bytes</i>
	<i>Mnemonic</i> (Name) ESC	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i> 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> Soft Stop		<i>Type</i> Immediate, Program	<i>NV Bytes</i> 2
	<i>Mnemonic</i> (Name) @	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i> 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.

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

(^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.

<i>Command</i> A	<i>Function</i> Read/Write to Ports		<i>Type</i> Immediate, Program		<i>NV Bytes</i> 2, 2
	<i>Mnemonic</i> (Name) A	<i>Data 1</i> 0-128	<i>Data 2</i>	<i>Result</i> 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.

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

(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 executing a motion command.
3. When not running a program.

Jogging is inhibited during actual command line entry.

<i>Command</i> C	<i>Function</i> Reinitialize Non-volatile Memory		<i>Type</i> Immediate		<i>NV Bytes</i>
	<i>Mnemonic</i> (Name) C	<i>Data 1</i> Page 0-9	<i>Data 2</i>	<i>Result</i> 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.

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. 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.

Command	Function	Type	NV Bytes
C	Reset Driver	Immediate	
	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.

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

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

(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.

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

(F) Find Home

The special Find Home algorithm is intended to eliminate mechanical hysteresis typically found in many switches and encoders. Home is always approached from the same direction based on the initial logic state of the Home switch and the value (0 or 1) assigned to the "d" direction byte.

1. The Find Home step velocity, using a normally open Home switch (actuation from logic high to low) is programmable over the entire slew velocity available; 18 to 23,000 SPS. Once the Home switch is encountered the system inertia typically overshoots the exact switch transition point so that the controller changes the direction signal and shifts the step speed down to the (I) initial parameter velocity. This direction reversal and speed reduction continues until the exact Home switch actuation point is reached and the Homing function is complete.
2. The Find Home step velocity, using a normally closed Home switch (actuation from logic low to high) will always be the (I) initial velocity parameter setting. Once the Home switch is actuated, all motion ceases and the Homing function is complete.

The following table illustrates the possible combinations of switch motion:

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

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

(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.

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

(H) Resolution Mode

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

Mode 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.

Mode 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.

Mode 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.

Command !	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 (stiction), and motor or system resonance’s. 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 criteria 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.

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

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

*Actual values are determined by the hardware configuration.

Command 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.

Command 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

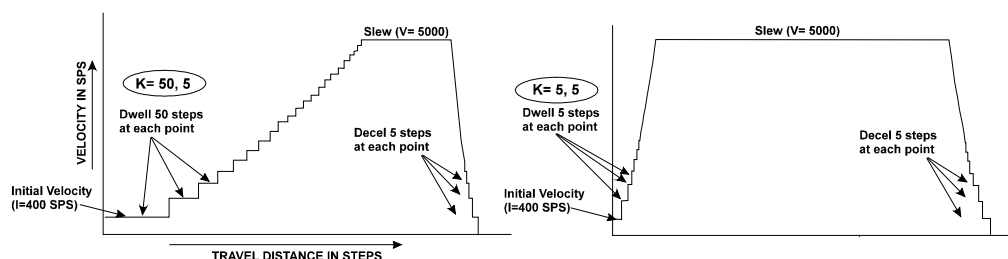
(K) Ramp Slope

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.

Command	Function		Type	NV Bytes
k	Special Trip		Program	5
	Mnemonic	Data 1	Data 2	Result
	(Name) k	Position $\pm 8,388,607$	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 power of 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:

Address		Description
0	"O 0"	The position counter is reset to zero
4	"k 200 8"	The initialize command "k" is first used to initialize the real time sequence.
		Assume that the command "k 200 8" is executed at the beginning of the program. The following actions take place:
		1. Port 4 is set on per data2 – see the "A" command.
		2. The first trip position is set per Data1 (200).
		3. A special trip program counter (PC) is set to 128.

9	" +6000 "	Now the +6000 index command is started. When the position matches 200, the command located at 128 is checked for either a "i" or "k."
128	"k 200 0"	The "k" changes the trip position to 400 and turns all output ports off (high logic voltage) This is executed while at the exact 400 position. The program counter is advanced to 133.
133	"k 600 16"	This is executed like the previous 128, new trip=600, ports 4 and are turned on and the program counter set to 138.
138	"i 200 8"	The restart command. 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

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

(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.

Command 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.

Command M	Function Move at Constant Velocity		Type Immediate, Program	NV Bytes 3
	Mnemonic (Name) M	Data 1 SPS $\pm 22,000$	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).

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.

N	Function Force Step Position		Type Program		NV Bytes 5
	Mnemonic (Name) N	Data 1 Steps $\pm 8,388,607.99$		Data 2	Result

(N) Force Step Position

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

O	Function Reset Position Counter to Zero		Type Program		NV Bytes 1
	Mnemonic (Name) O	Data 1		Data 2	Result

(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).

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 thorough 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 <CR>):

Response	Command	Comment
	P 0	Enter program mode
0 (echoed)	+5000	Index 5000 steps
5	- 5000	Index -5000 steps
10	P 0	End program, insert program marker (One program has been entered and stored in non-volatile memory. You may now insert another routine).
	P 100	Start program at memory location 100
100 (echoed)	O	Set origin to zero
101	R 2000	Move to position 2000
106	R - 500	Move to position -500
111	J 101 9	Repeat sequence 10 times
	P 0	End program
	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).

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

(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.

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

(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 $\pm 8,388,607.99$	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:

Non-encoder, Variable Resolution Mode, Range $\pm 8,388,607.99$ 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.

Non-encoder, Fixed Resolution Mode, Range $\pm 8,388,607$ 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.

Encoder, Variable Resolution Mode, Range $\pm 8,388,607$ 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.

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

Command 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:

Motion Command	Initializing Values (Reset via the "C8" command)
B Jog Speeds	3/20 (30/200 steps per second)
D Divide	0 (full step)
E Settling Delay	50 (500 Ms.)
H Resolution Mode	1 (variable)
I Initial Velocity	400
K Ramp Slopes	10/10
T Trip Point	Off
V Slew Speed	3004
Y Currents	20/75

Encoder Command	Initializing Values (Reset via the "C8" command)
D Dead Zone	0
E Encoder Setup	Off
H Hunt Resolution	
R Stall Retries	
S Lag Factor	
T Stall Delta	
V Hunt Speed	

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 T	Function		Type	NV Bytes
	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 <CR>):

1. Write program to location 0 (zero)

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

2. Write program to location 100

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

3. Set Trip Point

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

4. Run program

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

Command U	Function Control Line Feed		Type Immediate, Program	NV Bytes 2
	Mnemonic (Name) y	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.

Command 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)

Command 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.

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).

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

(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; 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.

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

(X/x) Examine ParametersSingle 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.

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

(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.

Command 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

([) 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

(]) Read Limits/Hardware/Watchdog Timer

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

"] 0" 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

"] 1" 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.

“1 2” 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:

Non-encoder, Variable Resolution Mode, Range $\pm 8,388,607.99$ 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.

Non-encoder, Fixed Resolution, Range $\pm 8,388,607$ 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).

Encoder, Variable Resolution Mode, Range $\pm 8,388,607$ 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.

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

(-) 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 effected 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.

Command 	Function		Type	NV Bytes
	Terminate Program		Immediate	2
	Mnemonic (Name)	Data 1	Data 2	Result 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.

Command ^N	Function		Type	NV Bytes
	Name Controller		None	N/A
	Mnemonic ^N	Data 1	Data 2	Result 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 MAX is ready to enter party line mode.

<i>Command</i> <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.

8) Encoder Commands

<i>Command</i> a	<i>Function</i> Read Stall Count		<i>Type</i> Immediate	<i>NV Bytes</i> 1
	<i>Mnemonic</i> (Name) a	<i>Data 1</i>	<i>Data 2</i>	<i>Result</i> 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

Command d	Function		Type	NV Bytes
	Set Deadband and Enable		Immediate, Program, Default	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.

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

(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/rev (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.

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

(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.

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

(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

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

(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).

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

(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.

m 0

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.

m 1

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.

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

(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.

Command 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.

Command 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.

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

(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:

ratio= 10 (2000 count encoder/200 step motor)

s= 20

t= 5

Then lag= $(10 * 20 * 5) / 256 = 3$

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.

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

(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 $10 \times 10 = 100$ encoder steps. The actual sampling should allow for some backlash, as defined by the stall factor.

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

(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.

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

(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.

9) Specifications

Electrical

	<u>MAX-410</u>	<u>MAX-420</u>
Power Supply.....	40Vdc @90VA	40Vdc @180VA
Output Current (Peak).....	4 Amps	
Chopping Frequency.....	28kHz	
Input Voltage.....	100 to 125 VAC, 60 Hz or 200 to 250 VAC, 50Hz	
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)	
AC Fuse		
MAX-410.....	1.0A/120VAC - .75A/240VAC (2AG/SB)	
MAX-420.....	3.15A/120VAC - 1.6A/240VAC (IEC127-2/SB)	

Signal Specifications

Serial I.O. (J1, J2)	Min	Typ	Max	Units
RS485 Inputs	-10		+15	Vdc (Note 2)
RS485 Outputs		5		Vdc (Note 2)
RS232 Inputs			±30	Vdc (Note 2)
RS232 Outputs	±5	±8		Vdc (Note 2)

I. O. (J3)	Min	Typ	Max	Units
I.O. Supply (OPTO+)	5		12	Vdc (Note 1)
OPTO+ Inputs	-0.7		OPTO+	Vdc

Aux. I.O. (J5)	Min	Typ	Max	Units
I.O. Supply (VIO)	5		28	Vdc (Note 1)
Inputs (Ports 1, 2, 3, 4)				
Input Voltage	-0.7		VIO	Vdc
Input Current			4	mA
Outputs (Ports 4, 5, 6)				
Output Voltage			36	Vdc
Output Current			0.5	Amp (cont.)

Encoder (J7)	Min	Typ	Max	Units
Supply Voltage		5		Vdc
Supply Current			50	mA
Inputs	-0.7		5.7	Vdc
Input Currents		5	10	mA

Notes:

- (1) VIO and OPTO+ internally diode OR'ed to provide 5Vdc to input circuitry.
 (2) RS232 or RS485 option.

Environmental

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

Physical

Weight

MAX-410.....4.37lbs. (1.98kg)

MAX-420.....9.37lbs. (4.25kg)

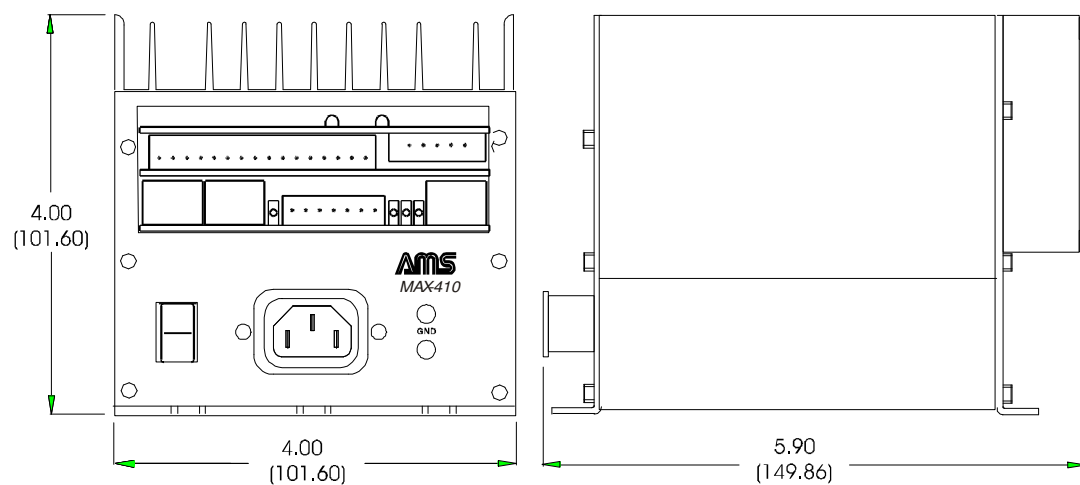
Size: inches (mm)

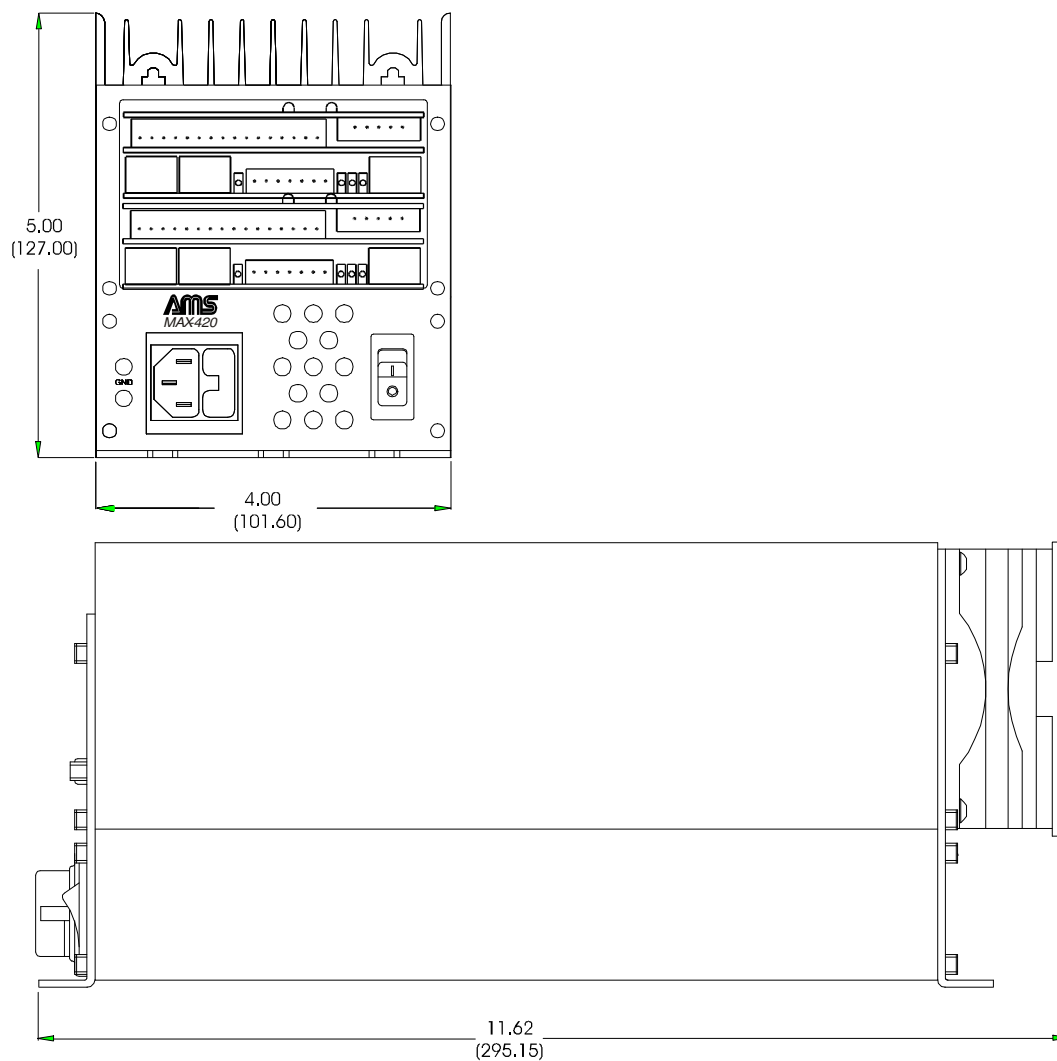
MAX-410.....4.0 (101.6) x 4.0 (101.6) x 5.9 (149.9)

MAX-420.....4.0 (101.6) x 5.0 (127.0) x 11.6 (295.2)

Dimensional Outline

MAX-410



MAX-420

10) Addendum

Command Summary

MNEMONIC / COMMAND		DATA 1	RANGE 1	DATA 2	RANGE 2	NV	D	I	P
MOTION 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 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:

- small “k”
- small “i”
- small “u” (replaces small “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.

MAX-410 and MAX-420

A faster microprocessor is implemented in the MAX-410 and MAX-420 product line but the watchdog timer and brown out functions are eliminated. Customers requiring these functions may order them as an option. The new version is identified by the suffix of “-X2” during sign on.

About Step Motor Current

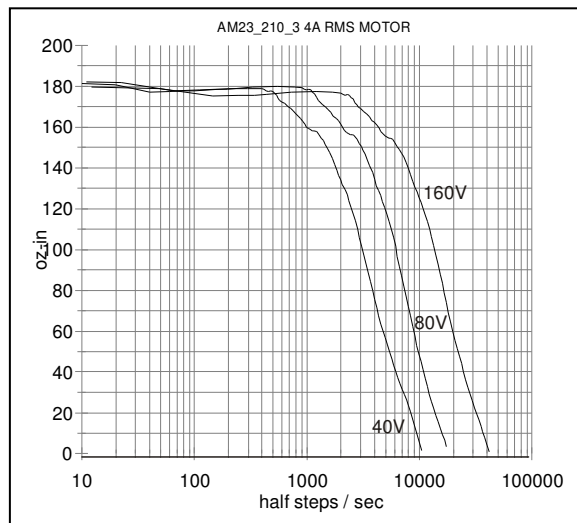
There is much confusion regarding the operation of step motors. Depending on your application, the step motor offers several advantages over servo motor designs, including lower cost and simplicity. The step (or stepper, or stepping) motor is a digital “synchronous” motor with a pre-designed number of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

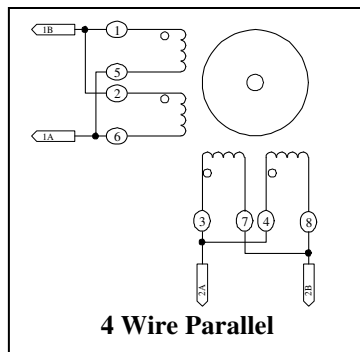
Current = 2 amps
Resistance = 1.0 ohm
Voltage = 2.0 volts
Inductance = 4.4 mH

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:

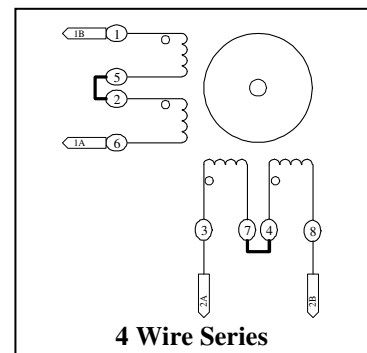
Parallel Resistance = 0.5 ohms
Parallel Inductance = 2.2 mH
Current = 4 amps (2 volts)
Watts per phase = $8 \times 2 \text{ phases} = 16 \text{ watts total}$

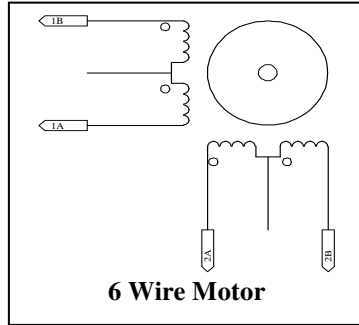
B. 4 Wire Series

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 = $8 \times 2 \text{ phases} = 16 \text{ watts total}$

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)

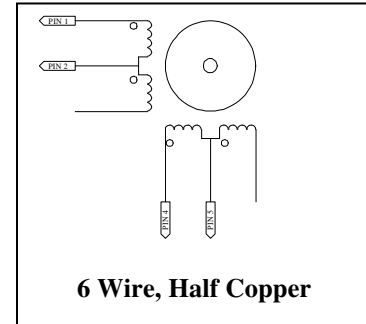
In practice the two coil ends are connected, while no connection is made to the center tap.

Half Copper or 50% Winding

The maximum speed can be increased by using $\frac{1}{2}$ 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

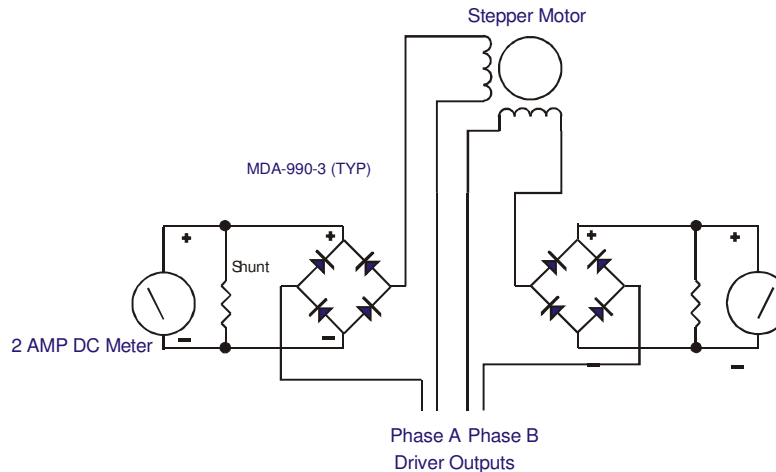
# Leads	Winding Connection	Per Phase Bipolar Current	
		Full Step (Total)	Microstep RMS (Peak)
4 wire	Parallel	4.0A (8.0A)	4.0A (5.66A)
4 wire	Series	2A (4.0A)	2.0A (2.8A)
6 wire	Unipolar	4A per phase	NA
6 wire	Series	2.8A (5.6A)	2.8A (4A)
6 wire	50% Copper	4A (8A)	4A (1.7A)

- 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: RMS Current for Dual Potentiometer Models

The two motor windings are adjusted separately in these microstep designs. There is one Sine (SIN) and one Cosine (COS) control. The preferred method is to enable both outputs to produce equal currents (RMS), and then adjust both potentiometers to equate the values.

1. Adjust both potentiometers to minimum (CCW). After applying power, the “E 3” command presets the digital phase data to equal values corresponding to the RMS value. One potentiometer (either one) is then increased to the desired RMS current.
2. A comparing LED is provided that changes state when both current settings are equal. Once one phase current is set, the other potentiometer can be “tweaked” until the LED changes to the opposite, on or off, condition. Both meters will affirm equal currents.

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.

HEAT: The Primary Enemy of Motor Damage

Advanced Micro Systems driver designs limit winding current to an adjustable value. Higher speeds are achieved from higher voltage DC supplies. In general, the only cause of motor damage is from excessive heating. Most step motors can withstand 100 degrees C.

A chopper drive regulates the motor current. Generally the “run” current is set at zero speed. If a hold mode is available the current will “set-back” when the motor is not moving. The voltage supplying the motor should be three or more times the rated motor voltage. If the supply were equal to the motor voltage, chopping would not function and performance would be very poor.

With higher voltages the regulation limits power and, hence heating. As the motor is rotated faster the chopper uses the available voltage to overcome a “back EMF” effect that takes place, thereby retaining more shaft torque.

More voltage applied to the motor equals more watts into the motor, resulting in additional heating. Motor heating can accumulate at continuous high stepping speeds, load torque and supply voltages of over 40 volts. Methods of reducing heating include motor heat sinking and short duty cycle (with a reduced holding current).