USER'S MANUAL

INTELLIGENT MOTOR CONTROLLERS

For VME bus

MAXv FAMILY

PRO-DEX, INC.

TWIN OAKS BUSINESS CENTER
1800 NW 169th PLACE, BUILDING C100
BEAVERTON, OR 97006
PHONE 503-629-8081
FAX 503-629-0688
EMAIL: SalesOR@pro-dex.com

WEB SITE http://www.pro-dex.com

3301-2700000

Rev. A

TABLE OF CONTENTS

This page intentionally left blank

1. GENERAL DESCRIPTION

1.1 INTRODUCTION

The Pro-Dex, Inc. Oregon Micro Systems MAXv motion controllers form a family of high performance VME bus-based products and are in compliance with the universal 6U VME Bus Specification ISO/IEC 15776 (2001 E). The MAXv motion controller can manage up to 8 axes of open-loop stepper, closed-loop stepper or servo systems, in any combination at the user's option, as incremental encoder feedback can be provided on each axis. The OMS MAXv controller synchronizes all independent or coordinated motion of up to 8 axes, while incorporating other critical signals, such as hard or soft limits, home, and other digital and/or analog I/O signals, to provide the motion solutions to perform virtually any task. With high level functionality, such as additional two encoder inputs, circular and linear interpolation, multi-tasking, custom profiling, backlash compensation, etc., the MAXv can satisfy the needs of most any motion control application. See Appendix C "Ordering Information" for specific MAXv family motion controller models and accessories. See block diagram of the MAXv.

The MAXv communicates as a "slave only" device and functions as a motion co-processor to the VME host. It utilizes patented proprietary technology to control the trajectory profile, acceleration, velocity, deceleration, and direction of selected axes. In response to commands from the host computer, the MAXv controller will calculate the optimum velocity profile to reach the desired destination in the minimum time, while conforming to the programmed acceleration and velocity parameters. In addition, the MAXv can provide motion control information such as axis and encoder positions as well as the state of over travel limit inputs, home switch inputs, and done flags. The MAXv motion controllers utilize an OMS custom VME and PowerPC Bridge, enabling a very efficient and rapid data transferring between the PowerPC and the VME host.

The stepper control of the MAXv produces an average 50% duty cycle square wave step pulse at velocities of 0 to 4,000,000 pulses per second and an acceleration of 0 to 8,000,000 pulses per second per second. The servo control utilizes a 16-bit DAC and outputs either bi-polar +/- 10V or unipolar 0 to +10V. The encoder feedback control can be used as closed-loop feedback for the servo PID, position maintenance for the stepper axes or strictly as a position reference for any axis. The encoder input supports single-ended quadrature TTL signals at a rate of up to 8 MHz and counts at a times 4 resolution. This means a 1000 line encoder will produce 4000 counts per revolution in the MAXv controller. The MAXv motion controller has 6 general purpose analog inputs that utilize a 16-bit ADC, with a DC range of +/- 10 VDC. Two additional +/- 10 V DACs that are not assigned to any axis are also available. Complete specifications for MAXv can be found in Appendix C.

The MAXv command set employs two or three ASCII character commands. These commands can be combined into character strings, using virtually any programming language. These ASCII command strings can be sent to the MAXv motion controller over the VME bus.

For detailed descriptions of the ASCII commands, including examples see Chapter 5: Command Structure.

1.2 SYSTEM OVERVIEW

The MAXv is a standard length size 6U VME module (6.299 x 9.187 Inches) that can be installed in a VME cage. The MAXv communication interface is accessed through the VME Bus via the J1/J2 connectors and is compliant with the VME Bus Specifications ISO/IEC 15776 (2001 E)

The MAXv utilizes an optimally configured Power PC RISC based 32-bit micro-controller and FPGA technology for extensive logic integration and flexibility.

The MAXv motion controller has three high density front panel SCSI connectors. The IOvMAX is the companion accessory break out module that makes for easy connections.

The IOvMAX supports a straight-through connector to connector cable configuration from the MAXv controller's three front panel connectors (two 68-pin connectors (SCSI3 type) and one 50-pin connector (SCSI2 type)). All signals from these connectors are routed through a PCB to a 180-pin screw-terminal block.

The IOvMAX also includes a connector that supports the 100-pin connector of the OMS VME58 (J29). This is intended to help reduce the cabling efforts of current users of the OMS VME58 when they begin using the MAXv controller.

The MAXv controller factory default is for 8 open-loop stepper axes. If you will require other types of axes, such as servos or closed-loop steppers, you will need to change these with the software commands that assign specific axis to be servo, open-loop steppers or closed-loop stepper axis (See ?PS, PSE, PSM and PSO commands). If you want these changes to be implemented with the next power up, then you must save them with the APP (Archive Parameter Power-Up command, page 5-57.)

The axes are defined as X, Y, Z, T, U, V, R & S as well as 0, 1, 2, 3, 4, 5, 6, and 7, respectively. The X-axis and axis-0 are one in the same, as well as the Y-axis and axis-1, Z-axis and axis-2, and so on. The two extra channels of encoder feedback are referred to as 8 and 9 and are not assigned directly to any axis of control.

The MAXv bus interface uses VME memory technology to provide a fast communication channel for the commands from the host PC as well as feedback of motion parameters, such as encoder positions. Commands may be written to this RAM by the host, thus eliminating the bottlenecks of I/O and port based communications. Critical motion parameters such as position and velocity are available, allowing the host to interrogate these parameters in real time while the motion is in progress. All of the data can be captured within the same update cycle.

Interrupt control and other data are available through blocks of dedicated memory and used similarly to registers. These registers report status on controller flags, over travel limit, done flag, and encoder slip for each axis. Details of the shared memory mapping, memory registers, and usage are found in Chapter 3.

All status bits are capable of generating an interrupt so that the host can interact using either polled or interrupt mode.

More details on the functionality of the MAXv controller are included in the following chapters.

2. GETTING STARTED

2.1 INSTALLATION

The Pro-Dex, Inc., Oregon Micro Systems MAXv motion controllers form a family of high performance VME bus-based products and are in compliance with the “Standard” universal 6U VME Bus Specification ISO/IEC 15776 (2001 E). The MAXv will occupy one full width slot in the VME card cage.

Please read through the following sections before attempting to install the MAXv motion controller, as some safety issues need to be considered prior to powering up the system. Although the MAXv is a low power device, there should be ventilation, including forced air, around the circuit board. The MAXv will draw all of its power from the VME bus, so no external power supply is needed.

The MAXv supports 16-bit, 24-bit, 32-bit address mode and five address modifier mode selections. The address mode (J12), board address (J13) and IRQ interrupt level (J12) are selectable via the hardware selection jumpers. See also Figure 2-1 VME Board Address and Operation Jumper Diagram and note Factory Defaults.

NOTE: Factory Default: Address Mode = A16, 16-bit addressing
IRQ = 101 (IRQ5)
Address Modifier = 101001 (0x29)
Base Address = 0xF000

2.2 TO PREPARE FOR INSTALLATION

Note: If you plan on changing any of the factory default jumper settings on J12 and J13 this should be done before installing the MAXv in the chassis. (See Figure 2-2 for MAXv component locations.)

NOTE: Only J12, J13, J7, and J8 are user selectable

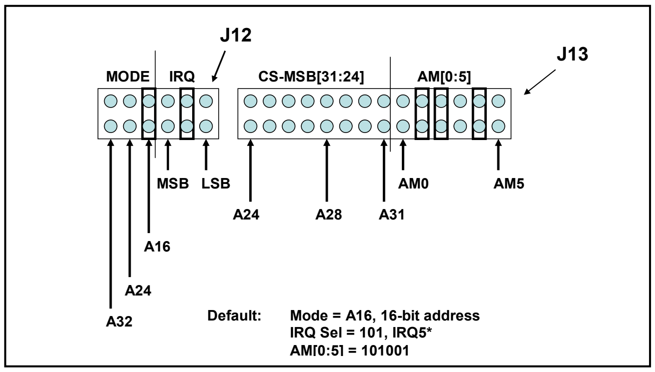


FIGURE 2-2, VME BOARD ADDRESS AND OPERATION OPTION JUMPER SWITCH DIAGRAM

J12 contains the IRQ and the address mode selection jumpers. The IRQ interrupt level range is 0x010-0x111 (IRQ2-7) and the default setting is 0x101 (IRQ5). The address mode selection supports the 16-bit, 24-bit, or 32-bit address space operation. Note that an open jumper indicates a “1” bit and a closed jumper is a “0”.

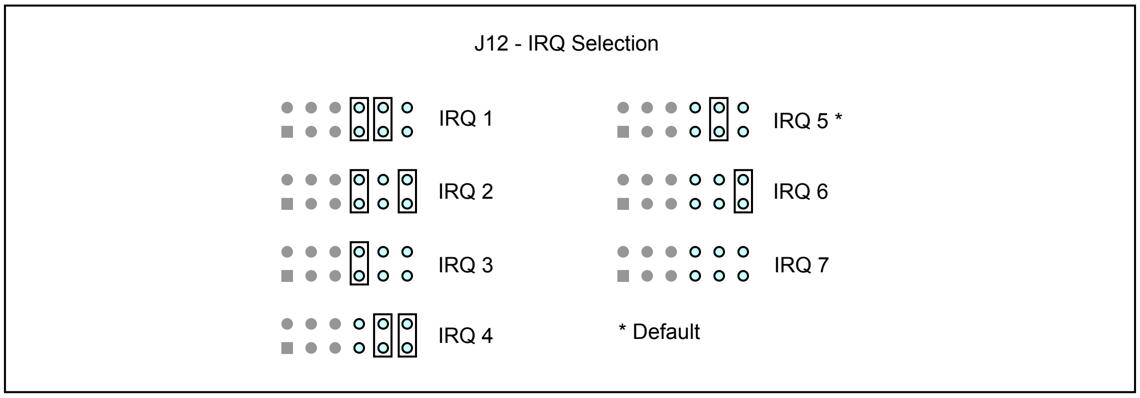


Figure 2-3 J12, IRQ SELECTION

J13 contains the hardware address modifier and the board’s main address selection jumpers. Note that not all combinations of address modifiers are valid. Please refer to the VMEbus specification and/or the user manual for your processor card.

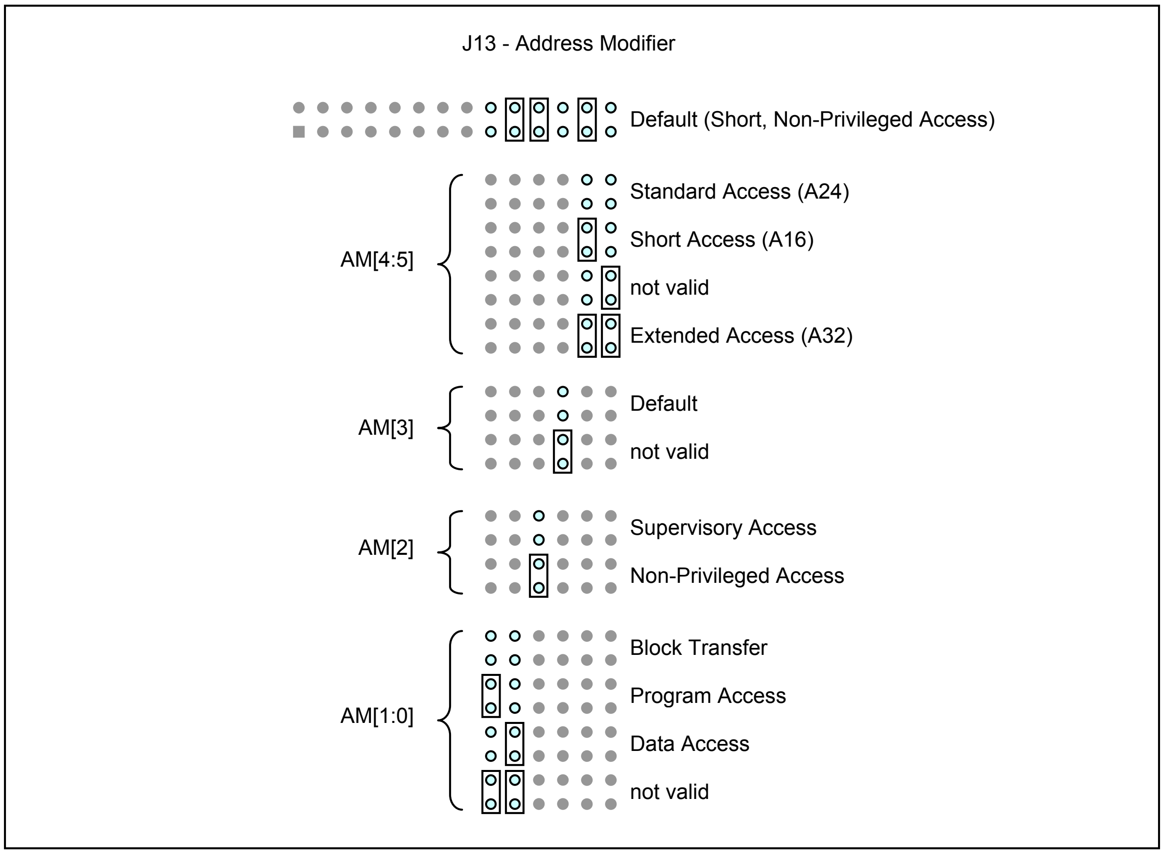
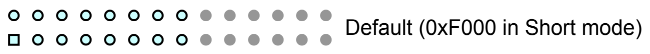


FIGURE 2-4 J13, ADDRESS MODIFIER

J13 Base Address Selection



		Addressing Mode	
	Extended	Standard	*Short*
J13.15-16	A31	A23	*A15*
J13.13-14	A30	A22	*A14*
J13.11-12	A29	A21	*A13*
J13.9-10	A28	A20	*A12*
J13.7-8	*A27*	*A19*	*(ignored)*
J13.5-6	*A26*	*A18*	*(ignored)*
J13.3-4	*A25*	*A17*	*(ignored)*
J13.1-2	*A24*	*A16*	*(ignored)*

FIGURE 2-5 J13, BASE ADDRESS SELECTION

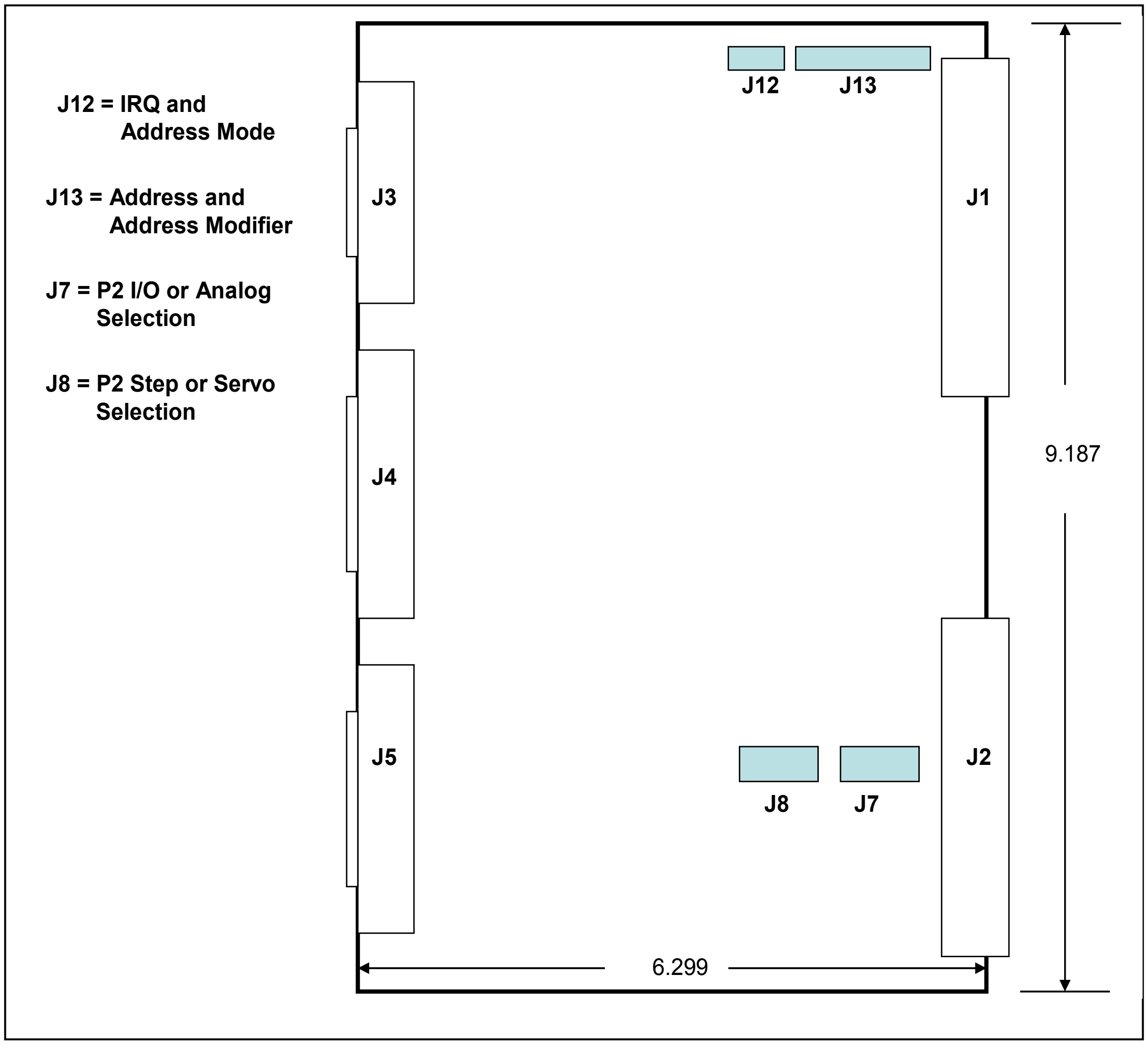

J12 = IRQ and Address Mode
J13 = Address and Address Modifier
J7 = P2 I/O or Analog Selection
J8 = P2 Step or Servo Selection
J12
J13
J3
J1
J4
9.187
J5
J2
J8
J7
6.299

Figure 2-6 MAXvPCB Diagram

NOTE: The J2 Backplane connector interface signals are selected on J8 as shown in Figure 2-7.

J8 routes signals to the J2 connector if they need to be controlled from the VME.

All analog and digital signals are accessible via the three front panel connectors (J3, J4 and J5).Most signal are also accessible via the VME 160-pin back plane connector. Further routing of signals to the back J2 connection is done with J8. (See also Figure 2-3.)

The signals for the STEP/SERVO are available on the front panel connectors (J4) and (J5), and hence on the IOvMAX at all times, however, those signals may be routed to the back J2 connector by setting the appropriate jumpers of J7.

J8 contains the STEP/SERVO output selection jumpers, selecting either the STEP or SERVO output. J7 contains the I/O/ANALOG selection jumpers for selecting the digital I/O or the analog I/O. J7, pin 8-13 selects either the analog input (ADC 0-6) or the general purpose I/O 8-13. J7, pin 14-15 selects either the analog output (DAC8-9) or the general purpose I/O14-15.

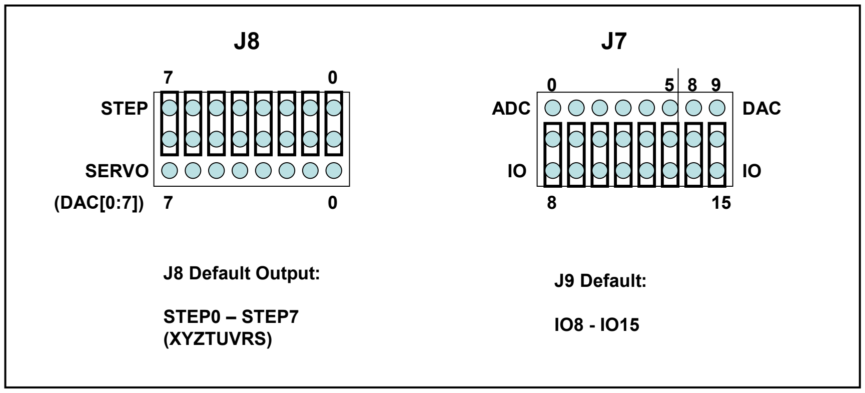


FIGURE 2-7 STEP/SERVO AND IO/ANALOG JUMPER SWITCH FOR THE BACKPLANE P2 INTERFACE DIAGRAM

2.3 HARDWARE INSTALLATION

Configure the MAXv board, as required, by setting appropriate jumpers or using factory defaults. Align the MAXv with the VME slot of the computer and insert the MAXv fully into the slot, seating the board ejectors.
Double check the board to ensure it is properly seated in the connector.

Caution

Establish communication with the controller board **before** wiring external components to the board (i.e. drivers and motors).

DO NOT make wiring connections to the controller board with power applied to the board.

Caution

ESD Warning: The MAXv, as well as most computers, are sensitive to Electro Static Discharge (ESD) and may be damaged if proper precautions are not taken to avoid ESD. Use properly grounded ESD mats, wrist-straps and other ESD techniques to prevent damage to the controller and/or computer.

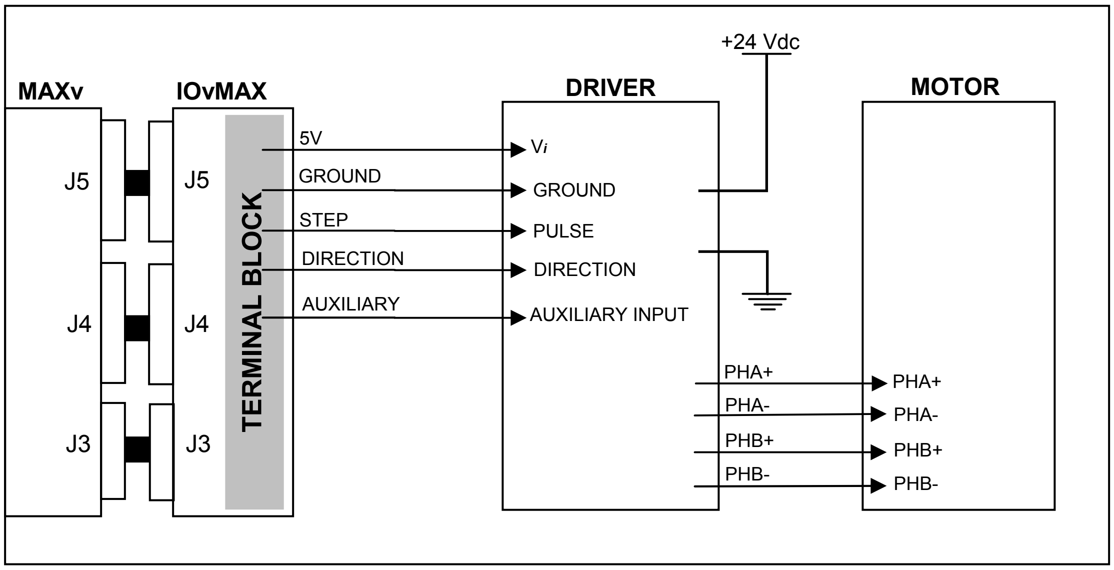


Figure 2-8
Example of Wiring Diagram of MAXv Controller Connected to a Stepper Driver / Motor

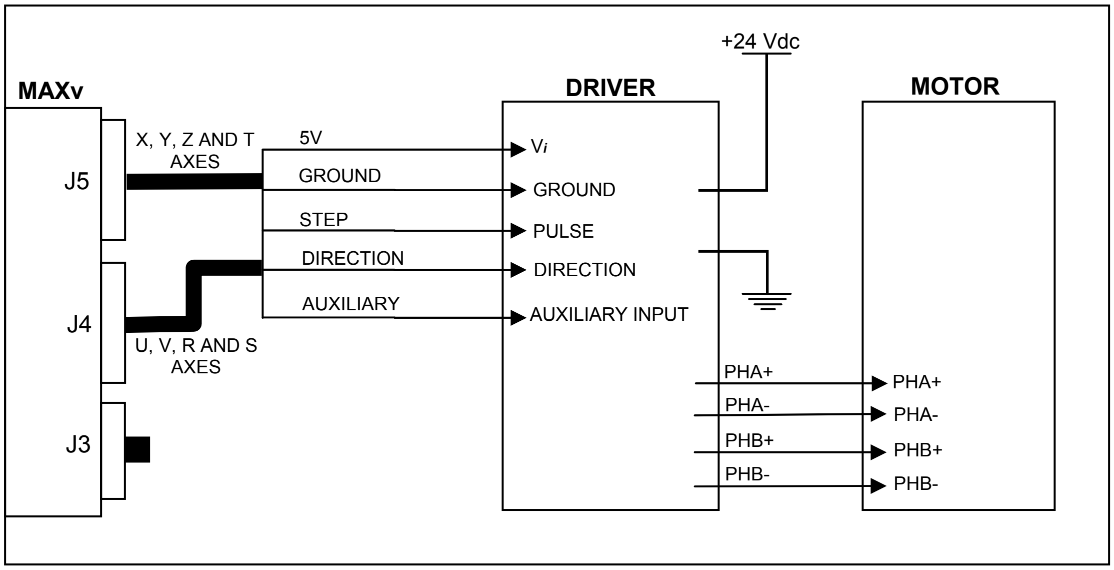


Figure 2-9
Example of Wiring Diagram of MAXv Controller via the IOvMAX Interface Module

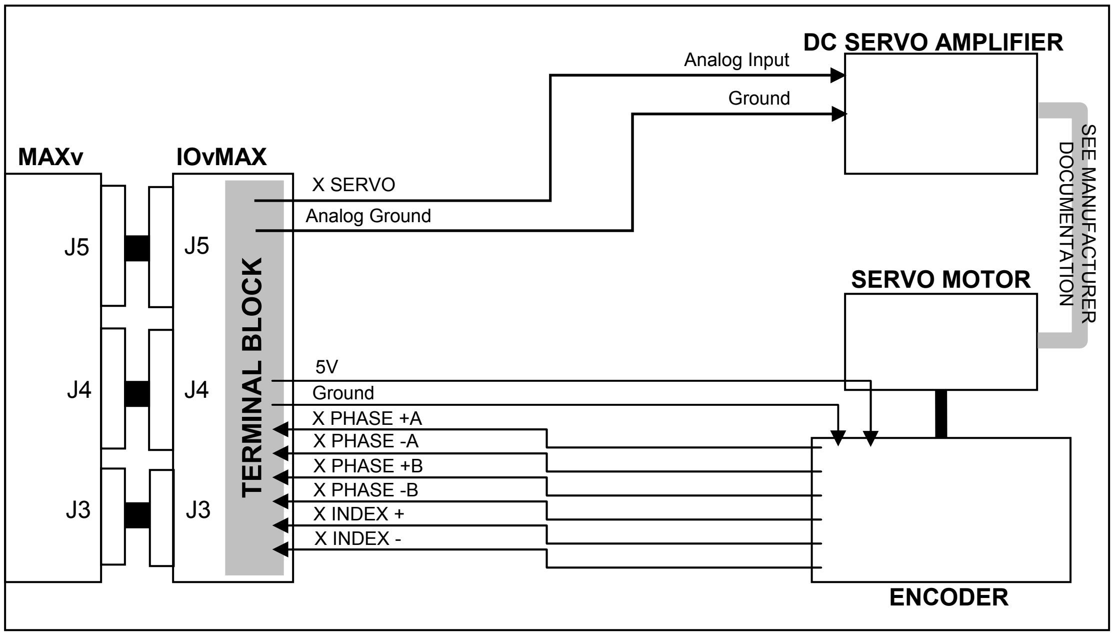


Figure 2-10
Example of Wiring Diagram of MAXv Controller via the IOvMAX Interface Module to Servo Motor

2.4 IOvMAX BREAKOUT MODULE

The IOvMAX breakout module is an accessory for the MAXv family. It provides an easy way to set all the control and I/O signals and provides a screw terminal connection for each signal. A block diagram is shown below.

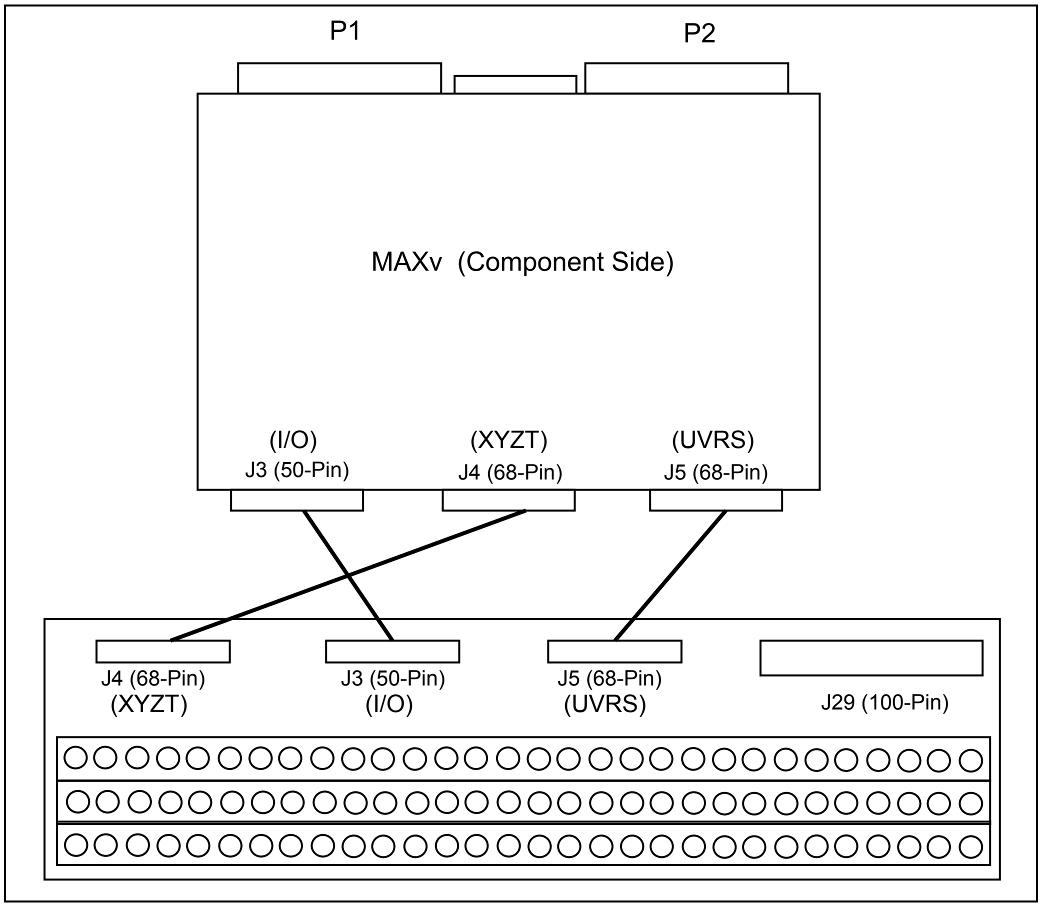


Figure 2-11 IOvMAX Break-Out to MAXv

The IOvMAX provides 180 screw terminals, one for each signal from the IOvMAX to the MAXv controller. The 100-pin connector on the IOvMAX is pin compatible with the VME58 controller. Details for the IOvMAX break out module are shown in Chapter 4 (page 4-13).

2.5 CONNECT AND CHECKOUT THE SERVO SYSTEM

Servo systems tend not to respond gracefully to connection errors. You can reduce the chance of making connection errors by following a step-by-step procedure:

Caution

The servo motor may jump or spin at a very high velocity during connection and configuration. The motor should be restrained by some means before beginning this procedure. Keep hands and clothing clear of the motor and any mechanical assemblies while performing this procedure.

It is recommended that the motor shaft not be connected to the physical system until you are sure you have control over the motor.

2.5.1 CONNECT AND CONFIGURE THE MOTOR/AMPLIFIER

1. Connect and configure your amplifier per the manufacturer's instructions for "Torque" or "Open-Loop" mode.
2. With the motor and amplifier power turned off, connect the MAXv to the amplifier.
3. Balance your motor:
    a. Using a voltage meter, verify that the command signal from the MAXv is less than 500mV. If it is not, send the command "KO0;" to the MAXv and recheck the voltage. If the voltage is still too high, contact Pro-Dex's Technical Support department for guidance.
    b. Turn on power to the amplifier and then to the motor.
    c. Adjust the balance setting of your amplifier (if equipped) until the motor stops moving.
    d. If the motor continues to revolve or your amplifier has no balance adjustment:
        i. Send the command "KO100;" to the MAXv.
        ii. If the motor spins faster, reduce the command parameter and resend the command, e.g. "KO50;"
        iii. If the motor spins slower but does not stop, increase the command parameter and resend the command, e.g. "KO150;"
        iv. Continue adjusting and resending the KO command until the motor comes to rest. Write down the final KO value for later reference as your "zero" setting.
4. Maximize your system's usage of the MAXv's DAC:
    a. Connect the servo encoder to the MAXv.
    b. Set the signal/command gain of your amplifier to its minimum setting.
    c. Send the "KO3277;" command to the MAXv and observe the velocity of the motor. The output of MAXv will be near 1VDC.

   d. If the motor does not move at all, your amplifier does not work well at a low velocity. In this case, adjust the signal/command gain of the amplifier to approximately 20% of maximum or until the motor begins to move.

   e. Using a frequency meter, measure the pulse rate of Phase +A of the encoder. The frequency measured is ¼ of the actual pulse rate.

   f. Adjust the signal/command gain of the amplifier until the pulse rate of Phase +A (4 is approximately 10% of your desired peak operational velocity. If the pulse rate is already greater than 10% of peak, your amplifier is not designed for low velocity motion and you will likely have some difficulty tuning your motors).

   g. Send the "KO-3277;" command to the MAXv and recheck the velocity. You may need to readjust your amplifier. If so, do not reduce the signal/command gain – only increase the setting as needed. Increasing the gain will not impair the forward peak velocity but reduction will.

   h. Send the KO command with the "zero" value noted at the end of step 3d (iv) to the MAXv.

5. Verify the direction of your servo encoder:

   a. Send the "LP0; KO2000;" command to the MAXv.

   b. Send the "RE" command to the MAXv and observe the response.

   c. If the response is positive, no further action need be taken; go to step 6.

   d. If the response is negative, your encoder must be reversed.

      i. If your encoder produces a differential signal, swap Phase + B with Phase - B and repeat from step (a.) above.

      ii. If your encoder produces a single-ended (or TTL) signal, swap Phase A with Phase B and repeat from step (a.) above.

   e. If the RE response is still negative, contact Pro-Dex Technical Support for assistance.

6. Repeat from step 1 for the other servo axes.

7. Remember to set KO for each axis at every power-up unless you store the values in flash (see Section 5: Command Structure)

NOTE: Most encoder problems are caused by lack of power or incorrect connections. If the encoder position changes by only 1 count, this is an indication that one of the phases is not connected.

Caution

Do not proceed until you perform all the steps in this procedure, ensure that the outputs of the MAXv are as described, and ensure that the encoder is operating correctly.

2.6 TUNE THE SYSTEM

2.6.1 INTRODUCTION

The following is an introduction to tuning a servo motor and the basics of the process of doing it. Tuning a servo system is the process of balancing three primary gain values Proportional, Integral, and Derivative in order to achieve optimum system performance.

In a closed-loop system, an error signal is derived from the command position and actual position, amplified, and then supplied to the motor to correct any error. Clearly, if a system is to compensate for infinitely small errors, the gain of the amplifier needs to be infinite. Real world amplifiers do not possess infinite gain; therefore, there is some minimal error which cannot be corrected.

The three primary gain values used in servo systems are **P** (proportional), **I** (integral) and **D** (derivative). The "P" term is used as a straight gain factor to get the system response "in the ballpark." The "I" term defines how quickly the system will respond to change. The "D" term is a dampening term. This term defines how quickly the system settles at its desired position without oscillating.

The effects of these parameters can be seen when looking at the system's response to a step change at the input. The shape of the step response falls into one of three categories: under damped, critically damped, or over damped. Over damped systems are slow to reach their final value and produce little or no oscillation. Critically damped systems reach final value quickly, without overshoot. Under damped systems reach final value quickly, but have various degrees of "ringing", or oscillation, that decay to zero over time. Ideally, a system should be critically damped, allowing for the fastest response time with the least amount of oscillation.

2.6.2 TUNING ASSISTANT

OMS' Tuning Assistant (only available on PCs using Windows) utility is provided to assist the user in finding the right combination of parameters. This utility plots the motor's response. The user can analyze this data to arrive at the right servo parameters for their servo system. The Tuning Assistant utility is available on the Software CD-ROM.

2.6.3 MANUAL TUNING

In most all motion control applications the optimum tuning of the servo system is achieved through a manual tuning process. Auto-tuning algorithms typically can only get the system parameters close and require manual steps to fine tune the parameters. An empirical trial and error approach will be discussed first.

There are some system parameters that need to be determined before attempting to tune a motor. The encoder resolution, counts per revolution, is one element to be determined. Another is the system's maximum velocity. Note that a motor should never exceed 90% of the motor's maximum rated RPM. If the system requirement is for a velocity higher than 90% of the motor's top RPM, then another motor with higher RPM capability should be used.

The system's maximum acceleration is determined several different ways. The best method is to determine the system time constant, which includes "hitting" or "bumping" the motor under system load and measure the time from 0 RPM to maximum RPM and divide this value by 5. The maximum acceleration is either 2.5 times this value, or is based on the system requirements for handling the load as defined in the operating specifications of the system. This value is always lower than the calculated value and if this acceleration value is not high enough then a different motor/amplifier with more power or bandwidth should be utilized.

The MAXv can control either current mode or voltage mode amplifiers. The servo update rate of the MAXv is user selectable: 976.6μs, 488.3μs, 244.1μs, 122.1μs. High "Following Error" can be compensated for using the feedforward coefficients explained later in this section. There are some general formulas that have been developed to determine acceptable following error for both current and velocity mode systems:

Current mode "Following Error" for KP = (3°/360°) × (counts per revolution)
Voltage mode "Following Error" for KP = (90°/360°) × (counts per revolution)

It is obvious that the voltage mode allows for much greater "Following Errors" than the current mode. This value is the "Following Error" when the motor is at peak velocity and will be used when determining the proportional gain (KP).

The "Following Error" for the integral term (KI) or long-term gain value will follow the guidelines below:

Current mode "Following Error" for KI = 0 counts
Voltage mode "Following Error" for KI = 80° of 360° (expressed in motor counts)

1. While still in open-loop mode, with 'hold' off (HF), use the KO command to zero the motor. This variable is used to provide a constant output that will compensate for torque offset from the load. So, when the system should be stationary, the necessary voltage will be sent to the amplifier to cause the motor to maintain position. With the correct KO value, the motor should successfully maintain a zero position.

   KO is the offset coefficient used while in closed-loop or open-loop mode, hold on (HN). You should have determined the correct value the KO variable before beginning to tune the PID filter.

The values for KO range from –32767 to 32767.

2. Set the known values for velocity, acceleration and the move distance for a trapezoidal profile with at least a 20% flat spot at peak velocity. Formula:

   Profile distance = ((peak velocity) ^2/ (2×acceleration)) ×2.4

   Example: ((50,000) ^2/ (2×500,000)) ×2.4 = 6,000

3. Execute the move by sending the move commands to the MAXv.

Example:

```
MR6000;
GO;
```

4. Adjust the KP term while repeating step 3 until the "following error" at the flat spot of the profile is acceptable. If the motor becomes unstable prior to obtaining the optimum KP term, then increase the KD term until the motor stabilizes.

Example:

```
LP0;
KP3;
HN;
MR6000;
GO;
LP0;
KP10;
HN;
MR6000;
GO;
LP0;
KP25;
HN;
MR6000;
GO;
LP0;
KD100;
HN;
LP0;
KP35;
HN;
MR6000;
GO;
LP0;
KD125;
HN;
```

The values in the above example are totally arbitrary and may vary drastically with different systems. The LP0 command is used to set the position error to 0.

The values for KP range from 0.00 to 32767.00.

5. Once the KP term has been obtained, continue executing the motion while raising the KI term until the long-term "following" error is acceptable. This error can be measured at the two knees of the motion profile. Increasing the KI term, increases the response time of your system. The motion profile should have a steeper slope as KI increases. (See FIGURE 2-12 and FIGURE 2-13.)

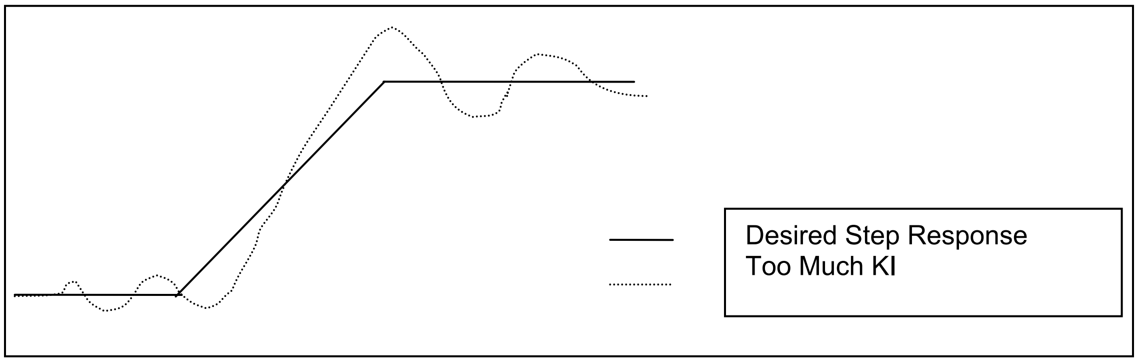


Figure 2-12

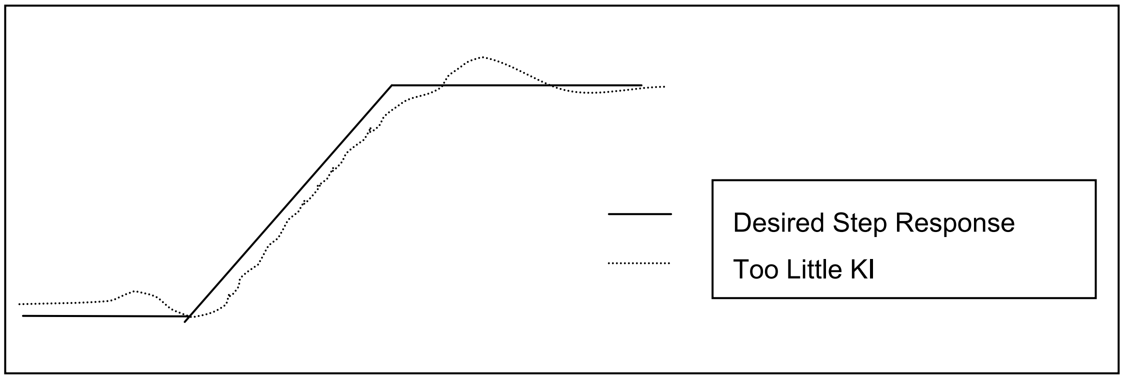


Figure 2-13

However, as KI increases the system can also become unstable. When the instability becomes unacceptable, increase the KD parameter. This will increase the dampening on the system's motion profile (therefore reducing oscillation or "ringing".) Continue adjusting the KI and KD terms until the proper response time is obtained.

The values for KI range from 0.00 to 32767.00.

6. If you are getting too much "ringing" in the motion profile, then increase KD to help dampen the system's response. If, instead, the system is over-damped and is reaching the final velocity too slowly, then reduce the KD parameter. Optimally, the system's motion profile should show the motor reaching the desired velocity as quickly as possible without overshoot and oscillation ("ringing").

The values for KD range from 0.00 to 32767.00.

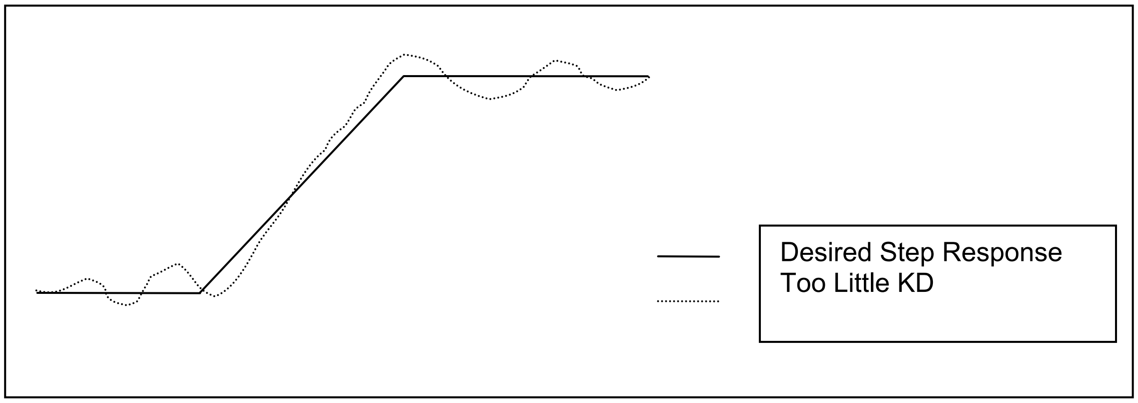


Figure 2-14

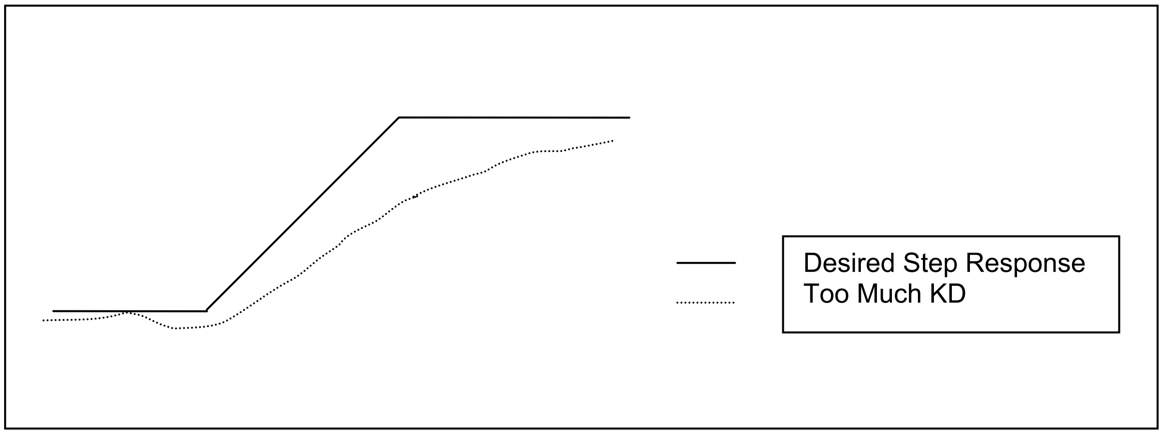


Figure 2-15

7. KP, KI, and KD are the primary parameters of concern when tuning a servo system. Once the optimum values for these variables have been determined, then you can adjust some of the secondary parameters that will help fine tune your system's performance. These other variables are described in the subsequent steps.

8. The KV variable is used when tuning velocity controlled servos (voltage mode servo amplifiers.) This is the velocity feedforward coefficient. KV determines how closely the system follows the desired constant velocity portion of the motion profile. By increasing this term, the "Following Error" of the system's response can be minimized. However, too large of a value may result in unstable behavior after command velocity changes.

The values for KV range from 0.00 to 32767.00.

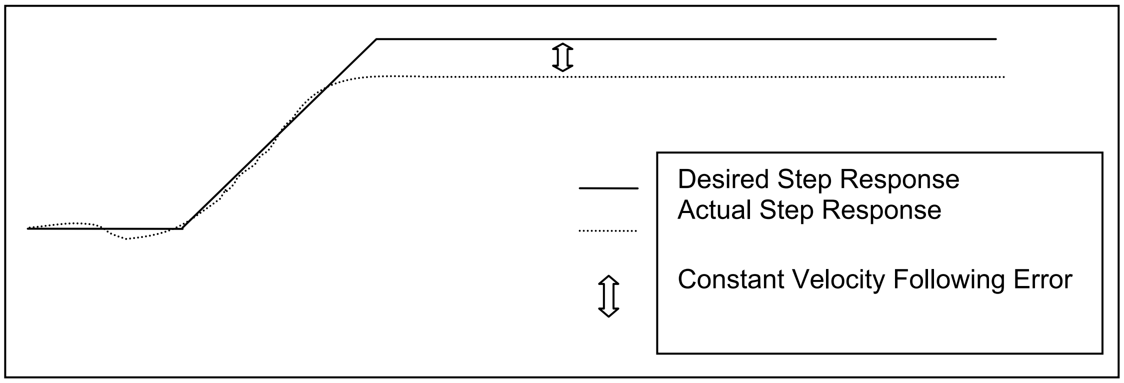


Figure 2-16

9. The KA variable is used when tuning torque controlled servos (current mode servo amplifiers.) This is the acceleration feedforward coefficient. Systems with high inertial loads may require additional torque during acceleration or deceleration to achieve optimum performance. KA determines how closely the system follows the desired acceleration and deceleration portions of the motion profile. Increasing this term reduces the “following error” occurring during acceleration and deceleration of the system. Although, if KA is too large, instability may occur.

The values for KA range from 0.00 to 34767.00.

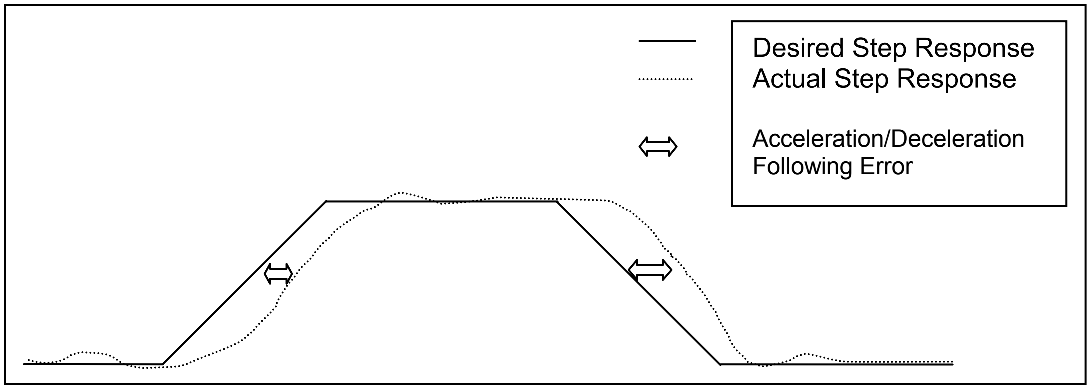


Figure 2-17

The block diagram below describes the feedback loop that is taking place in the servo system:

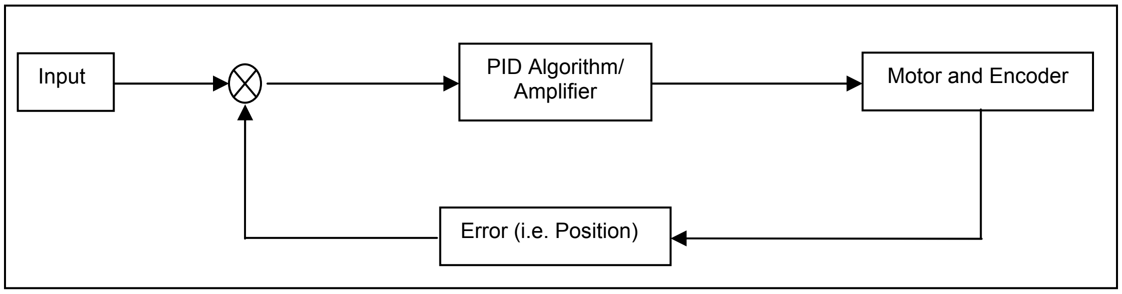


Figure 2-18

10. You may want to save the values for KD, KI, KP, etc., for future reference. These values can be saved in the board's flash memory, so they can be accessed easily on reset or power-up. The command APP will store your current parameter assignments, such as KD, KI, KP, etc., into flash memory. These saved parameters will then be used as the power up default set of values. Refer to page 5-57 for more detailed information regarding how to use the commands to save and load parameter sets from flash memory.

11. To verify that your motor is tuned properly after you have completed the first 10 steps perform the following test to test the holding torque: Send LP0; HN commands and check the shaft of the motor to make sure it is stiff. If there is play in the motor shaft when you turn it then you may have to re-adjust your PID filter.

12. Once you are satisfied with the static holding torque you could check for position error. Send the command "AC100000; VL5000; MR64000; GO;". With a 2000 line encoder this move would be equivalent to 8 revolutions of the motor. After the move is complete check the position error by sending the RE and RP commands for the specific axis you are moving. Compare the difference in the two responses. If they are the same then you are on the right track, if the error is greater than 327670 or the maximum allowable position error, the controller will disable the PID so that you don't have a runaway motor. In this case major changes to the PID parameters may be required. For minor differences in the encoder and the position reading you can fine-tune your PID filter according to the earlier steps. (See ES command for more options.)

NOTE: Reference Chapter 5: Command Structure for additional information on servo tuning commands and parameters.

2.7 SETTING THE USER DEFAULT CONFIGURATION

There are several parameters that can be defined by the user as default. These parameter values can supersede the factory default values and be stored in flash memory for power-up configuration. Most of these parameters consist of axis specific values; i.e. velocity, acceleration, limit switch, logic sense, etc.

The MAXv comes from the factory with default values for all parameters. For instance, the default value for the velocity of all axes is 200,000 counts per second. (A count is equivalent to a step pulse or one count of an encoder.) In a typical application, when the system is powered up, the main host computer would initialize all of the peripherals, such as the MAXv, sending each of the axes the peak velocity. When the User Definable Default Parameter value is defined, then the velocities of the defined axes will be set accordingly. This feature can greatly simplify the software and initialization process.

Once the values for all of the associated parameters are defined; i.e. velocity, acceleration, PID values, etc. then the "APP" Archive Parameters command is executed to place the values into flash memory. From this point forward these defined values will be used after reset or power-up. The individual parameters can be over-written at anytime by using the associated command; i.e. VL#, AC#, etc. To restore the factory defaults the command "RDF" Restore Factory defaults is executed. To restore the User Defined Default Parameters the command "RDP" Restore Defaults is executed. Refer to Section 5 Command Structure for more information on these commands and how they can be archived with the APP command.

The following is a list of parameters that can be defined as part of the User Definable Power-Up Default Parameters.

Description	Factory Default	Commands
Over travel limit (soft limit or hard limit)	Hard limit	SF, SL
Over travel limit (enabled or disabled)	Enabled	LF, LN
Over travel limit polarity (active high or active low)	Active low	LH, LL
Software based over travel for each axis	Disabled	TL
Step direction bit polarity	Low	DBI, DBN
Acceleration value for each axis	2,000,000	AC
Trajectory profile for each axis (linear, parabolic, S-curve, custom)	Linear	CN, LA, PF, PN, PR, SC, SR
Velocity Peak	200,000	VL
Velocity Base	0	VB
User Unit values for each axis:	Off	UU
Auxiliary output settle time for each axis	0	SE
Automatic auxiliary control axis by axis	Off	AF, AN, PA
Encoder ratio for each axis	1:1	ER
Encoder slip tolerance for each axis (used for stepper motors)	0	ES
Home Active State	Low	HH, HL
Position Maintenance Dead-Band, Hold Gain and Hold Velocity. (Used for stepper systems)	0,0,0	HD, HG, HV
Servo axis unipolar/bipolar output	Bipolar	BI, UN
Servo PID values:		
Acceleration feedforward	0	KA
Derivative gain coefficient	160	KD
Integral gain coefficient	1.00	KI
Servo DAC zero offset	0	KO
Proportional gain coefficient	10	KP
Velocity feedforward	0	KV
Axis type	Open-loop stepper	?PS, PSE, PSM, PSO
I/O bit level at power up can be high or low	High	BR
Encoder Home Pattern	101=Index high, B low, A high	EH
I/O Direction	8 inputs and 8 outputs	BD, IO
Custom and S-curve acceleration ramps	N/A	AJ
Update rate	1024	#UR

2.8 POWER SUPPLY REQUIREMENTS

The MAXv motion controller card plugs into the VME Bus. The MAXv is designed to fit into a standard full size card VME slot, and draws 1.2 amps from the +5V and 3.3V power supplies of the VME bus. For servo models only, +12V at 0.1 amp and -12V at 0.1 amps are also taken from the bus.

3. COMMUNICATION INTERFACE

3.1 INTRODUCTION

The VME64 Bus specification (ISO/IEC 15TT6 1002 (E) allows for a number of different options. The MAXv can support three modes A16, A24, and A32 with the default being 16-bit Non-Privileged mode.

3.2 VME INTERFACE

The VME interface is via the standard P1 and P2 interface using the 160-pin VME bus connectors.

3.3 VME COMMUNICATION INTERFACES

As shown in the simplified data flow diagram below communication between the MAXv controller and the application is via the VME shared memory and the hardware registers in the FPGA. Further details on specific items 1-12 are shown in the Data Dictionary, following the Data Flow Diagram, below.

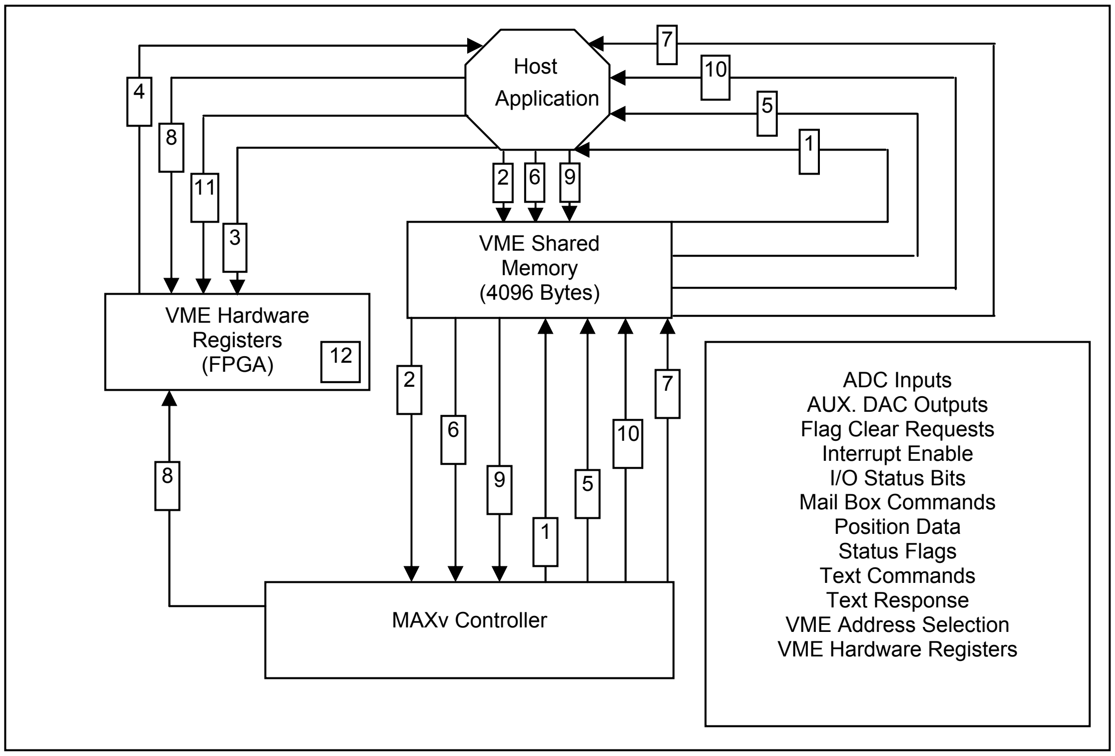


Figure 3-1 Data Flow Diagram

3.4 APPLICATION INTERFACE DATA DICTIONARY

Please refer to the Data Flow Diagram (Figure 3.1) for additional information and clarification.

DATA DICTIONARY

1. ADC Inputs = {6 Analog to Digital converter values read each motor update cycle and stored in shared memory}

2. Aux. DAC Outputs = {Auxiliary Digital to Analog output requests placed in the DAC0 or DAC1 shared memory register locations.}

3. Flag Clear Requests = {Bits written to Status Word 1 to clear selected status flags and dismiss the latched interrupt.
   OR
   Bits written to Status Word 2 to clear selected status flags and dismiss the latched interrupt.}

   Interrupt Enables = {Interrupt enable bits written to the Status Word 1 Interrupt enable register.
   OR
   Interrupt enable bits written to the Status Word 2 Interrupt enable register}

4. I/O Bits Status = {The state of the 16 general purpose I/O bits read and stored in shared memory each motor update cycle.
   OR
   The state of the Limit Sensor Inputs read and stored in shared memory each motor update cycle.
   OR
   The state of the Home Sensor Inputs read and stored in shared memory each motor update cycle.}

5. Mail Box CMDs =Command codes {CONTROLLER_ID_QUERY (1) or KILL_ALL_MOTION (2)
   OR
   RESET_CONTROLLER (3) or SOFTBOOT_CONTROLLER (4)} placed in the direct command mail box.

6. Position Data = {Axis motor positions and axis encoder positions updated and stored in shared memory each motor update cycle.
   OR
   Axis motor positions and axis encoder positions copied to shared memory on Request via the Position Request Mail box.
   OR
   Auxiliary Encoder 8 position updated and stored in shared memory each motor update cycle.
   OR
   Auxiliary Encoder 9 position updated and stored in shared memory each motor update cycle.
   OR
   Multi-axis Motion Profile Data transferred to shared memory via a mail box request.
   OR
   Position Capture Table Data transferred to shared memory via a mail box request.}

7. Status Flags = {Status Word 1 Flags = (8 axis done flags + 8 axis limit flags + 8 axis encoder slip flags + command error flag + response available flag + requested data available flag}

   OR

   Status Word 2 Flags = {Axis home flags}

8. Text CMDs = {Null terminated ASCII controller command strings placed in the VME shared memory ASCII command buffer.}

9. Text Responses = {Null terminated ASCII text response strings placed in the ASCII response buffer, by the controller, in response to query commands such as "RP".}

10. VME Address Selection = {Vector value (1 - 255) written to the VME IACK ID register.

    OR

    Address modifier value written to the VME Address Modifier register.}

11. VME Hardware Registers = The set of MAXv controller registers implemented in an FPGA.

    The registers include:

    Status Word 1 flag register = Done flags, Limit flags, Slip flags, CMD error, Response available & Requested data available.

    Status Word 1 interrupt enable = Bit by bit interrupt enable for Status Word 1 flags.

    Status Word 2 flag register = Axis home flags

    Status Word 2 interrupt enable =Bit by bit interrupt enable for Status Word 2 flags.

    VME IACK ID vector register = Vector Id value used to identify the Controller when an interrupt occurs.

    Controller Configuration Register= Controller Configuration Switch Status

    VME Address Modifier Register = VME bus address modifier selection Value

    FIFO Status and Control register = Not currently used.

    FIFO Data Register = Not currently used.

3.5 COMPARISON OF PREVIOUS OMS ARCHITECTURE

OMS motion controllers such as the VME58 family previously used hardware registers for status, slip, done and over travel limits. The MAXv uses the Power PC's Message unit, in combination with reserved storage regions in the common memory area, to accomplish these functions.

3.6 MAXv

To aid in the understanding of the interface, several flow charts have been provided. These include the elements to be handled for the initialization of the MAXv, a sample Interrupt Service Routine (ISR), a sample of sending a command to the controller, such as a Send String, and a sample of a SendAndGetString. The SendAndGetString would be a pattern for the WY (Who Are You, see page 5-241) command, used to identify the MAXv, its serial number, and revision levels of the firmware.

3.7 MAXv CONTROLLER INITIALIZATION

The following flow chart shows the activities required to initialize the MAXv controller.

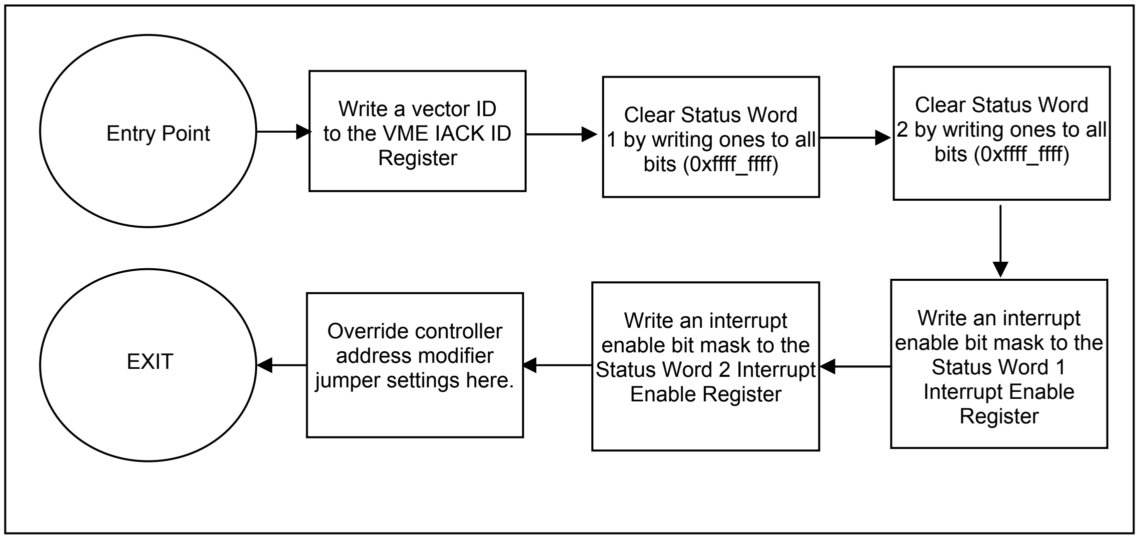

Entry Point
Write a vector ID to the VME IACK ID Register
Clear Status Word 1 by writing ones to all bits (0xffff_ffff)
Clear Status Word 2 by writing ones to all bits (0xffff_ffff)
Write an interrupt enable bit mask to the Status Word 1 Interrupt Enable Register
Write an interrupt enable bit mask to the Status Word 2 Interrupt Enable Register
Override controller address modifier jumper settings here.
EXIT

Figure 3-2 Controller Initialization

3.7.1 SAMPLE OF AN INTERRUPT SERVICE ROUTINE

Diagram A

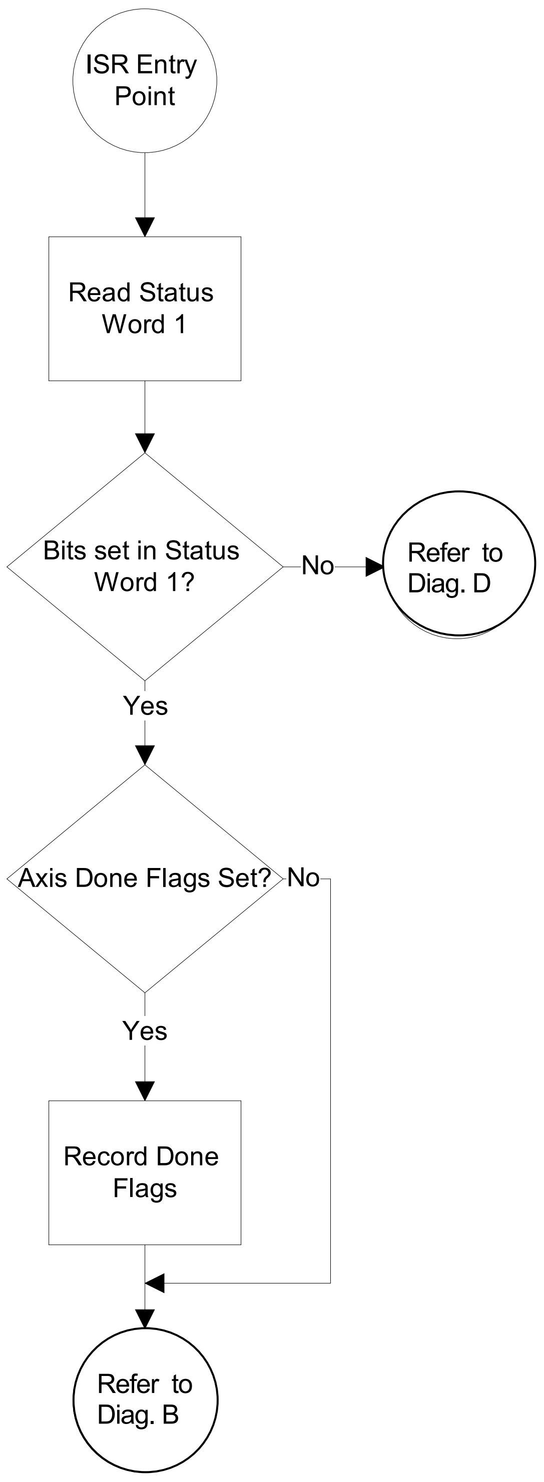


Diagram B

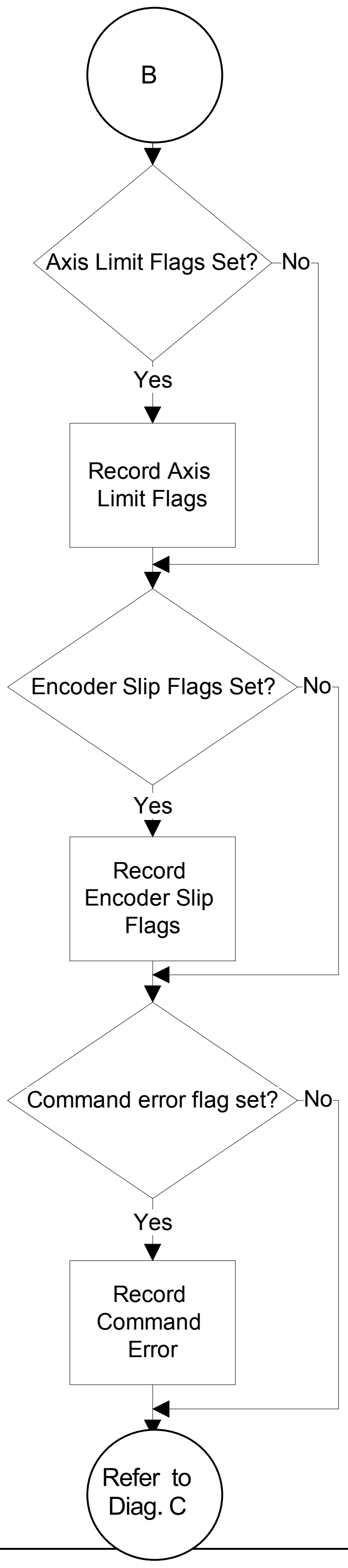


Diagram C

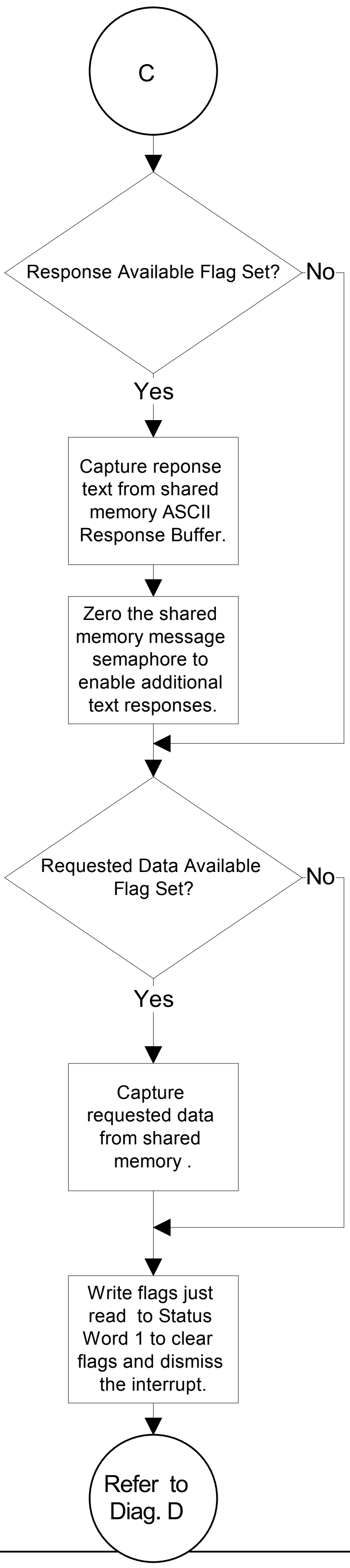
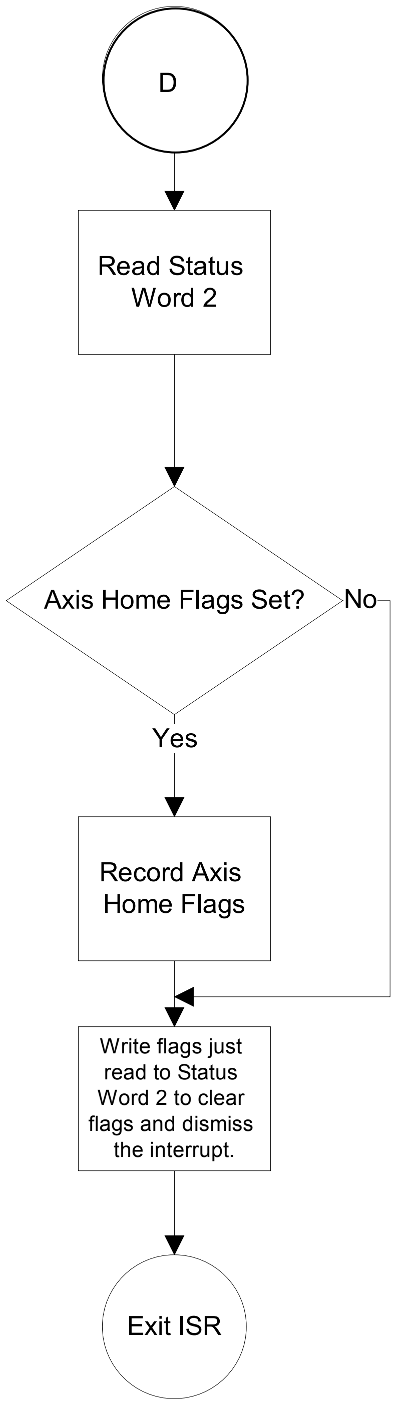
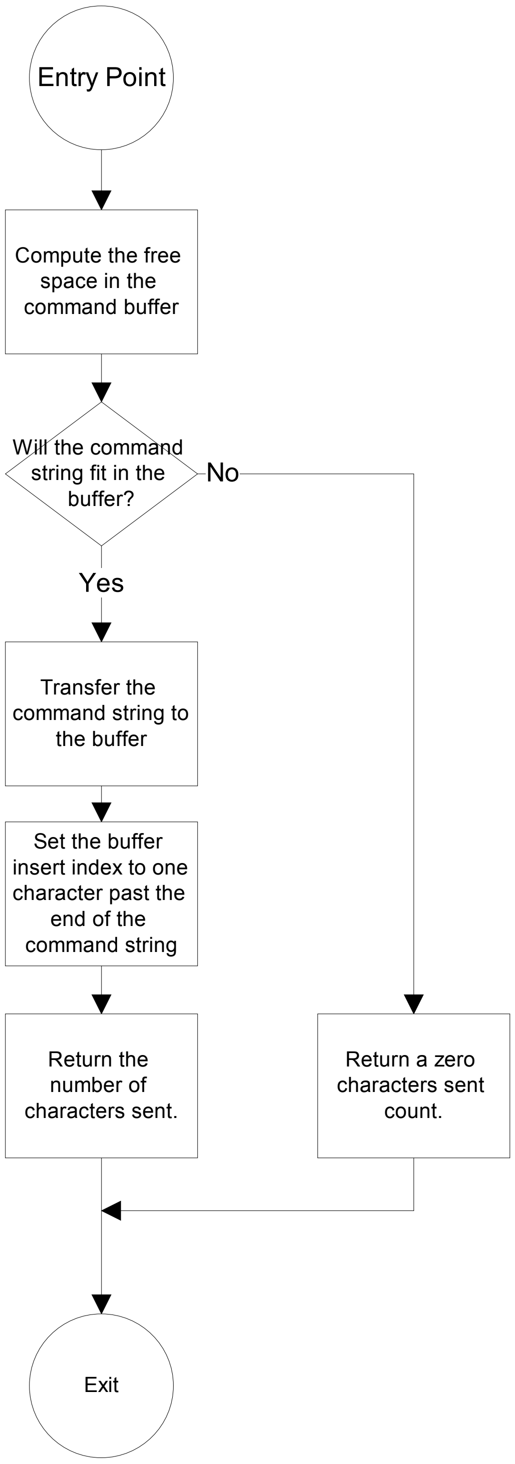

C
Response Available Flag Set?
No
Yes
Capture reponse text from shared memory ASCII Response Buffer.
Zero the shared memory message semaphore to enable additional text responses.
Requested Data Available Flag Set?
No
Yes
Capture requested data from shared memory .
Write flags just read to Status Word 1 to clear flags and dismiss the interrupt.
Refer to Diag. D

Diagram D


D
Read Status Word 2
Axis Home Flags Set?
No
Yes
Record Axis Home Flags
Write flags just read to Status Word 2 to clear flags and dismiss the interrupt.
Exit ISR

3.7.2 SAMPLE OF SEND STRING


Entry Point
Compute the free space in the command buffer
Will the command string fit in the buffer?
No
Yes
Transfer the command string to the buffer
Set the buffer insert index to one character past the end of the command string
Return the number of characters sent.
Return a zero characters sent count.
Exit

3.7.3 SAMPLE OF A SENDANDGETSTRING

Diagram A

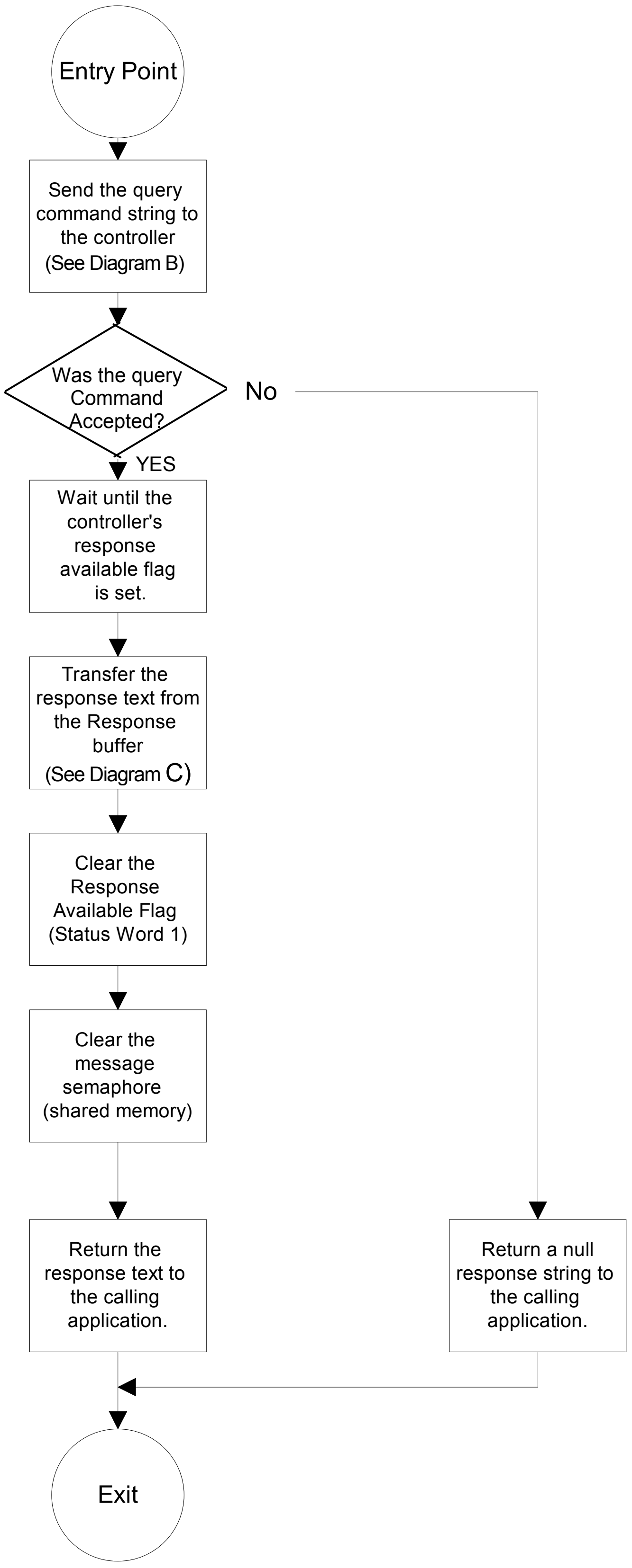

Entry Point
Send the query command string to the controller (See Diagram B)
Was the query Command Accepted?
No
YES
Wait until the controller's response available flag is set.
Transfer the response text from the Response buffer (See Diagram C)
Clear the Response Available Flag (Status Word 1)
Clear the message semaphore (shared memory)
Return the response text to the calling application.
Return a null response string to the calling application.
Exit

Diagram B

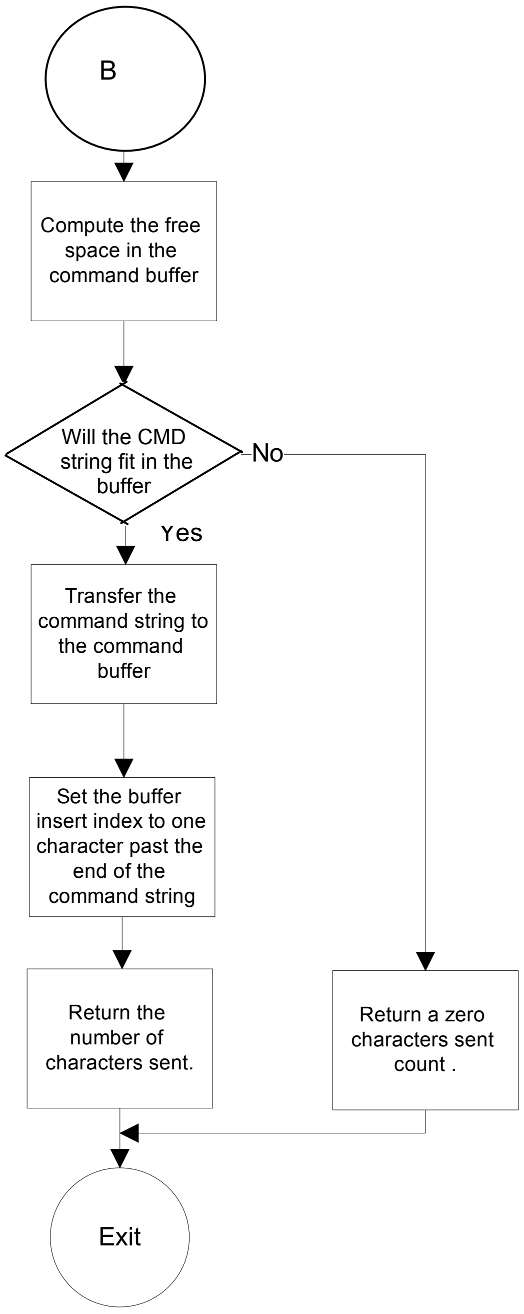
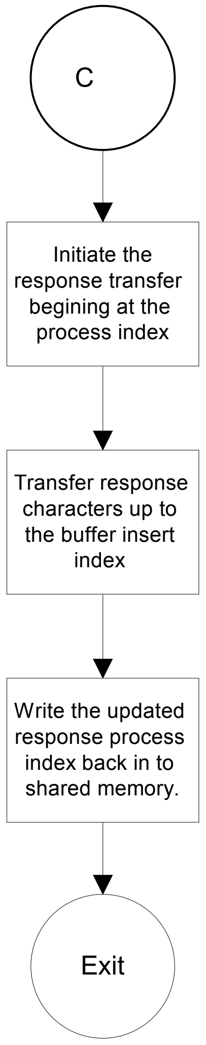

B
Compute the free space in the command buffer
Will the CMD string fit in the buffer
No
Yes
Transfer the command string to the command buffer
Set the buffer insert index to one character past the end of the command string
Return the number of characters sent.
Return a zero characters sent count .
Exit

Diagram C


C
Initiate the response transfer begining at the process index
Transfer response characters up to the buffer insert index
Write the updated response process index back in to shared memory.
Exit

NOTE: The WY (Who Are You, see page 5-241) is a command where the controller is asked to respond with its serial number, model number, and firmware revision level. As it establishes that the communication between the MAXv and the host is functioning properly, it should be the first send and get string developed.

3.8 MAXv - VME ADDRESS SPACE MEMORY / REGISTER MAP

The MAXv family of motion controllers, as do the VME58 controllers, utilizes 0x1000 (4096) bytes in the VME Controller/Host shared memory address space. On the MAXv, a set of jumpers is used to determine the base address for host accesses to this address space. The MAXv controller uses a base address of 0xff00_3000 to access this address space.

TABLE 3-1 MAXv - VME SHARED ADDRESS SPACE MAPPING

Byte Offset	Byte Offset (Hex)	Byte length	Description
The following 8 words contain axis motor positions. It is updated each motor update cycle.			
0	0x0	4	X axis motor position
4	0x4	4	Y axis motor position
8	0x8	4	Z axis motor position
12	0xc	4	T axis motor position
16	0x10	4	U axis motor position
20	0x14	4	V axis motor position
24	0x18	4	R axis motor position
28	0x1c	4	S axis motor position
The following words contain the axis encoder positions. It is updated each update cycle.			
32	0x20	4	X axis encoder position
36	0x24	4	Y axis encoder position
40	0x28	4	Z axis encoder position
44	0x2c	4	T axis encoder position
48	0x30	4	U axis encoder position
52	0x34	4	V axis encoder position
56	0x38	4	R axis encoder position
60	0x3c	4	S axis encoder position
The following word contains the axis limit status bits. It is updated each update cycle.			
64	0x40	4	Limit Switch status word
The following word contains the axis home sensor status bits. It is updated each update cycle.			
68	0x44	4	Home Switch status word
The following word contains the controller firmware status flags. It is updated as events occur.			
72	0x48	4	Firmware State flags
The following word is a direct command mechanism that bypasses the text command buffer.			
76	0x4c	4	Direct Command Mail Box
The following 17 words contain a memory region used to capture coherent snapshots of axis position.			
80	0x50	4	Position Request Mail Box
84	0x54	4	X axis motor position
88	0x58	4	Y axis motor position
92	0x5c	4	Z axis motor position
96	0x60	4	T axis motor position
100	0x64	4	U axis motor position
104	0x68	4	V axis motor position
108	0x6c	4	R axis motor position
112	0x70	4	S axis motor position
Byte Offset	Byte Offset (Hex)	Byte length	Description
116	0x74	4	X axis encoder position
120	0x78	4	Y axis encoder position
124	0x7c	4	Z axis encoder position
128	0x80	4	T axis encoder position
132	0x84	4	U axis encoder position

TABLE 3-1 MAXv – VME SHARED ADDRESS SPACE MAPPING (con't)

136	0x88	4	V axis encoder position
140	0x8c	4	R axis encoder position
144	0x90	4	S axis encoder position
The following word is used to coordinate the sending of text responses from the controller to the host.			
148	0x94	4	Message semaphore
152	0x98	4	Reserved
The following word contains the state of the 16 general purpose I/O bits, updated each update cycle.			
156	0x9c	4	General Purpose I/O bits status
160	0xa0	76	Reserved
236	0xec	4	Reserved
The following memory region contains various data transfer buffers.			
240	0xf0	4	ASCII Command Buffer insert index
244	0xf4	4	ASCII Command Buffer process index
248	0xf8	4	ASCII Response Buffer insert index
252	0xfc	4	ASCII Response Buffer process index
256	0x100	1024	ASCII Command Ring Buffer
1280	0x500	1024	ASCII Response Ring Buffer
2304	0x900	1024	Utility transfer buffer
3328	0xd00	704	Reserved
The remaining 64 bytes are reserved for the controller's hardware register set:			
4032	0xfc0	4	Status Word 1 flag register
4034	0xfc4	4	Status Word 1 interrupt enable register
4036	0xfc8	4	Status Word 2 flag register
4040	0xfcc	4	Status Word 2 interrupt enable register
4044	0xfd0	4	VME IACK ID Vector
4048	0xfd4	4	Controller configuration switch register
4052	0xfd8	4	Address modifier register
4060	0xfdc	28	Reserved
4088	0xff8	4	FIFO Status & Control
4092	0xffc	4	FIFO Data Register

TABLE 3-2 MAXv - Limit Switch Status Word (Word Access Offset 0x40)

Bit	Function
Byte access offset 0x43	
00	X axis positive limit sensor
01	Y axis positive limit sensor
02	Z axis positive limit sensor
03	T axis positive limit sensor
04	U axis positive limit sensor
05	V axis positive limit sensor
06	R axis positive limit sensor
07	S axis positive limit sensor
Byte access offset 0x42	
08	X axis negative limit sensor
09	Y axis negative limit sensor
10	Z axis negative limit sensor
11	T axis negative limit sensor
12	U axis negative limit sensor
13	V axis negative limit sensor
14	R axis negative limit sensor
15	S axis negative limit sensor
Byte access offset 0x41	
16	Not used
17	Not used
18	Not used
19	Not used
20	Not used
21	Not used
22	Not used
23	Not used
Byte access offset 0x40	
24	Not used
25	Not used
26	Not used
27	Not used
28	Not used
29	Not used
30	Not used
31	Not used

TABLE 3-3 HOME SWITCH STATUS WORD (WORD ACCESS OFFSET 0X44)

Bit	Function
Byte access offset 0x47	
00	X axis home sensor
01	Y axis home sensor
02	Z axis home sensor
03	T axis home sensor
04	U axis home sensor
05	V axis home sensor
06	R axis home sensor
07	S axis home sensor
Byte access offset 0x46	
08	Not used
09	Not used
10	Not used
11	Not used
12	Not used
13	Not used
14	Not used
15	Not used
Byte access offset 0x45	
16	Not used
17	Not used
18	Not used
19	Not used
20	Not used
21	Not used
22	Not used
23	Not used
Byte access offset 0x44	
24	Not used
25	Not used
26	Not used
27	Not used
28	Not used
29	Not used
30	Not used
31	Not used

TABLE 3-4 MAXv CONTROLLER FIRMWARE STATUS FLAGS (WORD ACCESS OFFSET 0x48)

Bit	Function
00	Controller application code not downloaded to RAM.
01	Controller application code is initializing.
02	Controller application code is running.
03	Not used
04	Not used
05	Not used
06	Not used
07	Not used
08	Application stored in flash memory has a check sum error.
09	A programming error occurred while storing the application code in flash memory.
10	Not used
11	Not used
12	A checksum error was detected in the power up default parameter archive.
13	A programming error occurred while storing parameters in the power up default parameter archive.
14	A checksum error was detected in the alternate parameter archive.
15	A programming error occurred while storing parameters in the alternate parameter archive.
16	The power up default parameter set has been loaded into working memory.
17	The alternate parameter set has been loaded into working memory.
18	The factory default parameter set has been loaded into working memory.
19	Not used
20	Not used
21	Not used
22	Not used
23	Not used
24	Not used
25	Not used
26	Not used
27	Not used
28	Not used
29	Not used
30	Not used
31	Not used

NOTE: If the firmware state register contains 0xFFFF_FFFF then the controller has not completed power up initialization.

3.8.1 MAXv CONTROLLER STATUS

Once set by the controller, the host must clear a status bit by writing a 1 to that bit. Individual bits in the controller in Status Word 1 can be configured to interrupt the host when the controller sets them to a 1. Bit interrupts are enabled by setting the corresponding bit in the Status Word 1 Interrupt Enable register. Note: if a bit interrupt is enabled and the controller sets that bit, then the interrupt will not be dismissed until the host clears that bit.

TABLE 3-5 Controller Status Word 1 Flag Register (Word Access Offset 0xfc0)

Bit	Function
Byte access offset 0xfc3	
00	X axis done bit
01	Y axis done bit
02	Z axis done bit
03	T axis done bit
04	U axis done bit
05	V axis done bit
06	R axis done bit
07	S axis done bit
Byte access offset 0xfc2	
08	X axis over-travel detected
09	Y axis over-travel detected
10	Z axis over-travel detected
11	T axis over-travel detected
12	U axis over-travel detected
13	V axis over-travel detected
14	R axis over-travel detected
15	S axis over-travel detected
Byte access offset 0xfc1	
16	X axis encoder slip detected.
17	Y axis encoder slip detected.
18	Z axis encoder slip detected.
19	T axis encoder slip detected.
20	U axis encoder slip detected.
21	V axis encoder slip detected.
22	R axis encoder slip detected.
23	S axis encoder slip detected.
Byte access offset 0xfc0	
24	Command error
25	Text response is available.
26	Requested data is available.
27	Not used
28	Not used
29	Not used
30	Not used
31	Not used

3.8.2 MAXv CONTROLLER STATUS WORD 1 INTERRUPT ENABLES

Setting a bit to a 1 enables the interrupt for the corresponding bit in controller status word 1.

TABLE 3-6 Controller Status Word 1 Interrupt Enables (Word Access Offset 0xfc4)

Bit	Function
Byte access offset 0xfc7	
00	X axis done bit interrupt enable.
01	Y axis done bit interrupt enable.
02	Z axis done bit interrupt enable.
03	T axis done bit interrupt enable.
04	U axis done bit interrupt enable.
05	V axis done bit interrupt enable.
06	R axis done bit interrupt enable.
07	S axis done bit interrupt enable.
Byte access offset 0xfc6	
08	X axis over-travel detected interrupt enable.
09	Y axis over-travel detected interrupt enable.
10	Z axis over-travel detected interrupt enable.
11	T axis over-travel detected interrupt enable.
12	U axis over-travel detected interrupt enable.
13	V axis over-travel detected interrupt enable.
14	R axis over-travel detected interrupt enable.
15	S axis over-travel detected interrupt enable.
Byte access offset 0xfc5	
16	X axis slip detected interrupt enable.
17	Y axis slip detected interrupt enable.
18	Z axis slip detected interrupt enable.
19	T axis slip detected interrupt enable.
20	U axis slip detected interrupt enable.
21	V axis slip detected interrupt enable.
22	R axis slip detected interrupt enable.
23	S axis slip detected interrupt enable.
Byte access offset 0xfc4	
24	Command error interrupt enable.
25	Text response is available interrupt enable.
26	Requested data is available interrupt enable.
27	Not used
28	Not used
29	Not used
30	Not used
31	Not used

3.8.3 MAXv CONTROLLER STATUS WORD 2

Individual bits in the controller in Status Word 2 can be configured to interrupt the host when the controller sets them to a 1. Note if a bit interrupt is enabled and the controller sets that bit then the interrupt will not be dismissed until the host clears that bit.

TABLE 3-7 MAXv Controller Status Word 2 (0xfc8)

Bit	Function
Byte access offset 0xfcb	
00	X axis home detected.
01	Y axis home detected.
02	Z axis home detected.
03	T axis home detected.
04	U axis home detected.
05	V axis home detected.
06	R axis home detected.
07	S axis home detected.
Byte access offset 0xfca	
08	Not used
09	Not used
10	Not used
11	Not used
12	Not used
13	Not used
14	Not used
15	Not used
Byte access offset 0xfc9	
16	Not used
17	Not used
18	Not used
19	Not used
20	Not used
21	Not used
22	Not used
23	Not used
Byte access offset 0xfc8	
24	Not used
25	Not used
26	Not used
27	Not used
28	Not used
29	Not used
30	Not used
31	Not used

3.8.4 CONTROLLER STATUS WORD 2 INTERRUPT ENABLE

Setting a bit to a 1 enables the interrupt for the corresponding bit in controller status word 1.

TABLE 3-8 MAXv Controller Status Word 2 (0xfcc)

Bit	Function
Byte access offset 0xfcf	
00	X axis home detected interrupt enable.
01	Y axis home detected interrupt enable.
02	Z axis home detected interrupt enable.
03	T axis home detected interrupt enable.
04	U axis home detected interrupt enable.
05	V axis home detected interrupt enable.
06	R axis home detected interrupt enable.
07	S axis home detected interrupt enable.
Byte access offset 0xfce	
08	Not used
09	Not used
10	Not used
11	Not used
12	Not used
13	Not used
14	Not used
15	Not used
Byte access offset 0xfcd	
16	Not used
17	Not used
18	Not used
19	Not used
20	Not used
21	Not used
22	Not used
23	Not used
Byte access offset 0xfcc	
24	Not used
25	Not used
26	Not used
27	Not used
28	Not used
29	Not used
30	Not used
31	Not used

3.9 COMPARISON TO VME58

The following is a list of the Shared Memory allocation for the VME58 family of controllers, and is shown here to make it easier to find the differences in the assignments when developing a new driver for the MAXv environment.

TABLE 3-9 VME58 - VME Shared Address Space Memory Mapping

Byte Offset (10)	Byte Offset (Hex)	Byte length	Description	Mem. Function
The Dual Port RAM Memory Region begins here				
0	0x0	2	Command char Put Index	Read Only
2	0x2	2	Response char Get Index	Read Only
4	0x4	512	Command Char Buffer	Read Only
516	0x204	508	Reserved	Read Only
1024	0x400	4	X axis encoder position	Read Only
1028	0x404	4	X axis command position	Read Only
1032	0x408	4	X axis command velocity	Read Only
1036	0x40c	4	X axis acceleration	Read Only
1040	0x410	4	X axis velocity limit	Read Only
1044	0x414	4	X axis base velocity limit	Read Only
1048	0x418	4	X axis proportional gain	Read Only
1052	0x41c	4	X axis differential gain	Read Only
1056	0x420	4	X axis integral gain	Read Only
1060	0x424	4	X axis accel. feed fwd.	Read Only
1064	0x428	4	X axis vel. feed fwd.	Read Only
1068	0x42c	4	X axis DC offset	Read Only
1072	0x430	80	Reserved	Read Only
1152	0x480	4	Y axis encoder position	Read Only
1156	0x484	4	Y axis command position	Read Only
1160	0x488	4	Y axis command velocity	Read Only
1164	0x48c	4	Y axis acceleration	Read Only
1168	0x490	4	Y axis velocity limit	Read Only
1172	0x494	4	Y axis base velocity limit	Read Only
1176	0x498	4	Y axis proportional gain	Read Only
1180	0x49c	4	Y axis differential gain	Read Only
1184	0x4a0	4	Y axis integral gain	Read Only
1188	0x4a4	4	Y axis accel. feed fwd.	Read Only
1192	0x4a8	4	Y axis vel. feed fwd.	Read Only
1196	0x4ac	4	Y axis DC offset	Read Only
1200	0x4b0	80	Reserved	Read Only
1280	0x500	4	Z axis encoder position	Read Only
1284	0x504	4	Z axis command position	Read Only
1288	0x508	4	Z axis command velocity	Read Only
1292	0x50c	4	Z axis acceleration	Read Only
1296	0x510	4	Z axis velocity limit	Read Only
1300	0x514	4	Z axis base velocity limit	Read Only
1304	0x518	4	Z axis proportional gain	Read Only

TABLE 3-9 VME58 - VME Shared Address Space Memory Mapping (Continued)

Byte Offset (10)	Byte Offset (Hex)	Byte length	Description	Mem. Function
1308	0x51c	4	Z axis differential gain	Read Only
1312	0x520	4	Z axis integral gain	Read Only
1316	0x524	4	Z axis accel. feed fwd.	Read Only
1320	0x528	4	Z axis vel. feed fwd.	Read Only
1324	0x52c	4	Z axis DC offset	Read Only
1328	0x530	80	Reserved	Read Only
1408	0x580	4	T axis encoder position	Read Only
1412	0x584	4	T axis command position	Read Only
1416	0x588	4	T axis command velocity	Read Only
1420	0x58c	4	T axis acceleration	Read Only
1424	0x590	4	T axis velocity limit	Read Only
1428	0x594	4	T axis base velocity limit	Read Only
1432	0x598	4	T axis proportional gain	Read Only
1436	0x59c	4	T axis differential gain	Read Only
1440	0x5a0	4	T axis integral gain	Read Only
1444	0x5a4	4	T axis accel. feed fwd.	Read Only
1448	0x5a8	4	T axis vel. feed fwd.	Read Only
1452	0x5ac	4	T axis DC offset	Read Only
1456	0x5b0	80	Reserved	Read Only
1536	0x600	4	U axis encoder position	Read Only
1540	0x604	4	U axis command position	Read Only
1544	0x608	4	U axis command velocity	Read Only
1548	0x60c	4	U axis acceleration	Read Only
1552	0x610	4	U axis velocity limit	Read Only
1556	0x614	4	U axis base velocity limit	Read Only
1560	0x618	4	U axis proportional gain	Read Only
1564	0x61c	4	U axis differential gain	Read Only
1568	0x620	4	U axis integral gain	Read Only
1572	0x624	4	U axis accel. feed fwd.	Read Only
1576	0x628	4	U axis vel. feed fwd.	Read Only
1580	0x62c	4	U axis DC offset	Read Only
1584	0x630	80	Reserved	Read Only
1664	0x680	4	V axis encoder position	Read Only
1668	0x684	4	V axis command position	Read Only
1672	0x688	4	V axis command velocity	Read Only
1676	0x68c	4	V axis acceleration	Read Only
1680	0x690	4	V axis velocity limit	Read Only
1684	0x694	4	V axis base velocity limit	Read Only
1688	0x698	4	V axis proportional gain	Read Only
1692	0x69c	4	V axis differential gain	Read Only
1696	0x6a0	4	V axis integral gain	Read Only
1700	0x6a4	4	V axis accel. feed fwd.	Read Only
1704	0x6a8	4	V axis vel. feed fwd.	Read Only
1708	0x6ac	4	V axis DC offset	Read Only
1712	0x6b0	80	Reserved	Read Only
1792	0x700	4	R axis encoder position	Read Only
1796	0x704	4	R axis command position	Read Only
1800	0x708	4	R axis command velocity	Read Only

TABLE 3-9 VME58 - VME Shared Address Space Memory Mapping (Continued)

1804	0x70c	4	R axis acceleration	Read Only
1808	0x710	4	R axis velocity limit	Read Only
1812	0x714	4	R axis base velocity limit	Read Only
1816	0x718	4	R axis proportional gain	Read Only
Byte Offset (10)	Byte Offset (Hex)	Byte length	Description	Mem. Function
1820	0x71c	4	R axis differential gain	Read Only
1824	0x720	4	R axis integral gain	Read Only
1828	0x724	4	R axis accel. feed fwd.	Read Only
1832	0x728	4	R axis vel. feed fwd.	Read Only
1836	0x72c	4	R axis DC offset	Read Only
1840	0x730	80	Reserved	Read Only
1920	0x780	4	S axis encoder position	Read Only
1924	0x784	4	S axis command position	Read Only
1928	0x788	4	S axis command velocity	Read Only
1932	0x78c	4	S axis acceleration	Read Only
1936	0x790	4	S axis velocity limit	Read Only
1940	0x794	4	S axis base velocity limit	Read Only
1944	0x798	4	S axis proportional gain	Read Only
1948	0x79c	4	S axis differential gain	Read Only
1952	0x7a0	4	S axis integral gain	Read Only
1956	0x7a4	4	S axis accel. feed fwd.	Read Only
1960	0x7a8	4	S axis vel. feed fwd.	Read Only
1964	0x7ac	4	S axis DC offset	Read Only
1968	0x7b0	80	Reserved	Read Only
2048	0x800	2	Response Put Index	Read/Write
2050	0x802	2	Command Get Index	Read/Write
2052	0x804	512	Response Buffer	Read/Write
2564	0xa04	1500	Reserved	Read/Write
The Hardware Register Set begins here				
4064	0xfe0	1	Not used	
4065	0xfe1	1	Control Register	Read/Write
4066	0xfe2	1	Not used	
4067	0xfe3	1	Status Register	Read Only
4068	0xfe4	1	Not used	
4069	0xfe5	1	User I/O bits (0-7)	Read Only
4070	0xfe6	1	Not used	
4071	0xfe7	1	Encoder Slip Flags	Read Only
4072	0xfe8	1	Not used	
4073	0xfe9	1	Done Flags	Read Only
4074	0xfea	1	Not used	
4075	0xfeb	1	User I/O bits (8-13)	Read Only
4076	0xfec	1	Not Used	
4077	0xfed	1	Limit Switch Status	Read Only
4078	0xfee	1	Not Used	
4079	0xfef	1	Home Switch Status	Read Only
4080	0xff0	1	Not used	
4081	0xff1	1	Interrupt Vector	Read/Write
4082	0xff2	14	Not Used	

TABLE 3-10 VME58 - Control Register (0xfe1)

Bit Number	Bit Function
7	Data Area Update Request
6	Unused
5	Encoder Slip Interrupt Enable
4	Limit Register Interrupt Enable
3	Done Register Interrupt Enable
2	Interrupt Request to VME58
1	I/O bits 0 and 1 Interrupt Enable
0	Interrupt Request Enable

TABLE 3-11 VME58 - Status Register (0xfe3)

Bit Number	Bit Function
7	Command Error
6	Initialized (Power up complete)
5	Encoder Slip
4	Over-travel Encountered
3	Done
2	Interrupt Request to VME58 status
1	Interrupt Request from VME58 status
0	Interrupt Request status

TABLE 3-12 VME58 -Axis Done Flags Register (0xfe9)

Bit Number	Bit Function
7	X Axis Done
6	Y Axis Done
5	Z Axis Done
4	T Axis Done
3	U Axis Done
2	V Axis Done
1	R Axis Done
0	S Axis Done

TABLE 3-13 VME58 - Encoder Slip Register (0xfe7)

Bit Number	Bit Function
7	X Axis Slip
6	Y Axis Slip
5	Z Axis Slip
4	T Axis Slip
3	U Axis Slip
2	V Axis Slip
1	R Axis Slip
0	S Axis Slip

TABLE 3-14 VME58 - Limit Switch Status Register (0xfed)

Bit Number	Bit Function
7	X Axis Limit
6	Y Axis Limit
5	Z Axis Limit
4	T Axis Limit
3	U Axis Limit
2	V Axis Limit
1	R Axis Limit
0	S Axis Limit

TABLE 3-15 VME58 - Home Switch Status Register (0xfef)

Bit Number	Bit Function
7	X Home
6	Y Home
5	Z Home
4	T Home
3	U Home
2	V Home
1	R Home
0	S Home

TABLE 3-16 VME58 - User I/O Bits (0-7) Register (0xfe5)

Bit Number	Bit Function
7	I/O Bit 0
6	I/O Bit 1
5	I/O Bit 2
4	I/O Bit 3
3	I/O Bit 4
2	I/O Bit 5
1	I/O Bit 6
0	I/O Bit 7

TABLE 3-17 VME58 - User I/O Bits (8-13) Register (0xfeb)

Bit Number	Bit Function
7	I/O Bit 8
6	I/O Bit 9
5	I/O Bit 10
4	I/O Bit 11
3	I/O Bit 12
2	I/O Bit 13
1	Not used
0	Not used

TABLE 3-18 VME58 - Interrupt Vector Register (0xff1)

Bit Number	Bit Function
7	Least Significant Bit of vector number
6	
5	
4	
3	
2	
1	
0	Most Significant Bit of vector number

4. CONTROL SIGNAL INTERFACE

4.1 INTRODUCTION

The MAXv family of motion controllers is available in configurations from one to eight axes and manages any combination of servo and step motor systems. The MAXv connects to the VME bus through the VME P1 and VME P2 connectors, through rows A, B, & C, as did the OMS VME58 previously. The MAXv default configuration is an open-loop stepper controller for the number of axes purchased.

The connectors; two 68-pin SCSI3 and one 50-pin SCSI2, are located on the front panel of the MAXv for easy connection to the IOvMAX breakout board (accessory). The IOvMAX also has a 100-pin connector that is pin compatible with the J29 connector of the VME58 family.

The MAXv controller fully meets the ISO/IEE/5776(2001)E specification and plugs directly into a VME 6U slot of a VME rack.

4.2 GENERAL PURPOSE I/O, LIMIT AND HOME INPUTS & ANALOG INPUTS

To facilitate system implementation, limit and home inputs are provided for each axis. Limits may be activated by mechanical switches using contact closures or other suitable active switches, such as a Hall Effect switch or an opto-isolator that connects to ground.

If the motor travels beyond its allowable limits and trips the switch, the limit switch closure removes the excitation from the affected axis. (Servo motor systems should be designed for safety, i.e. to have electrical braking to stop them). The limit switch active signal state can be selected True High or True Low with the LH or LL command on an axis by axis basis. The behavior of the limit functionality can be controlled with the System Status and Control Commands (see page 5-19).

4.3 CONTROL OUTPUT

The MAXv is configured at the factory to control open-loop stepper motors for the number of axes purchased. Upon installation, each axis can be configured for servo motors, stepper motors with or without encoder feedback or any combination thereof. The servo output may be either unipolar analog (0/+10V) or bipolar analog (-10/+10 V). (See the UN and BI commands in Section 5 Command Structure (on page 5-226 and on page 5-66, respectively.))

4.3.1 I/O INTERFACE CONNECTIONS

The MAXv has four types of I/O interface connections; Open Drain I/O, Single ended active I/O, Differential I/O, and Analog I/O.

Open Drain I/O:

Auxiliary, direction, and step are examples of Open Drain I/O. These signals are transmitted as twisted pair signals. The desirable load value is 75-150 Ohms.

TABLE 4-1 STEP / AUX, DIR, OPEN DRAIN DC CHARACTERISTICS

Symbol	Min	Type	Max	Unit
I sink			120	mA
R_{on} (48mA)		5	7	Ω
Vcc			7	V

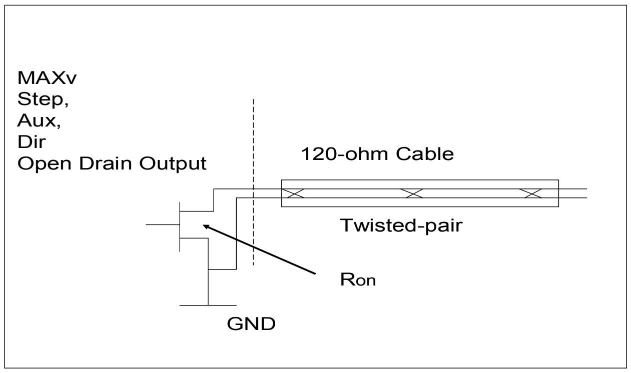


Figure 4-1 Open Drain I/O Interface Diagram

TABLE 4-2 RECOMMENDED NON-ISOLATED INTERFACE COMPONENT VALUE

Symbol	Description	Min	Type	Max
Vcc	Pull-up Voltage	3.0V		7.0V
RL	Pull-up Resistor		120Ω	
Cable	Cable Impedance		120Ω	

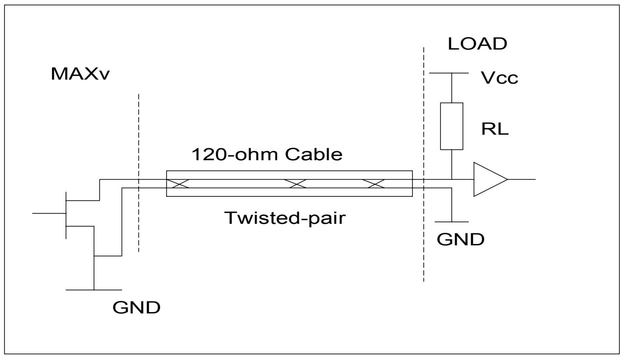


Figure 4-2 Non-Isolated Interface

TABLE 4-3 RECOMMENDED ISOLATED INTERFACE COMPONENT VALUE

Symbol	Description	Min	Type	Max
C1	5V filter capacitance		0.1uF	
R1	Opto series resistor		270Ω	
Cable	Cable impedance		120Ω	

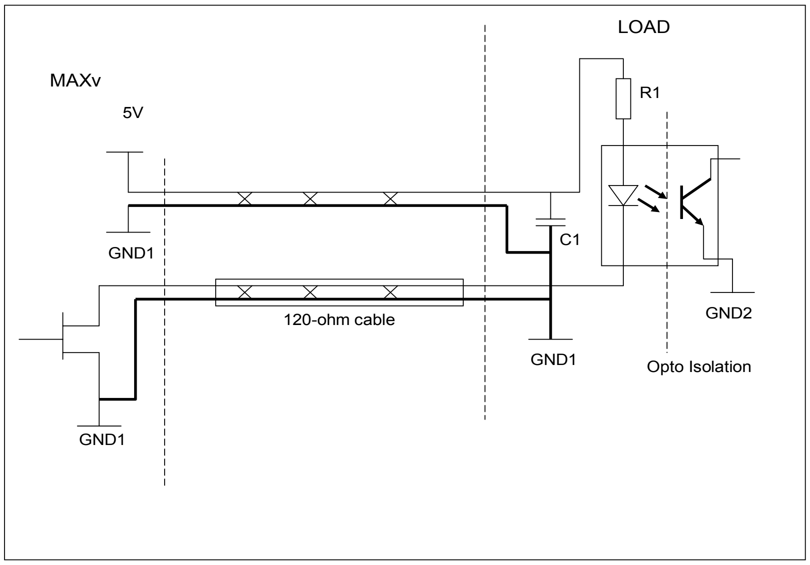


Figure 4-3 Recommend Isolated Interface

Table 4-4 Encoder / Index Input DC Characteristics

Symbol	Parameter	Min	Type	Max	Unit
Vdiff	Diff input threshold			200	mV
V_{hy}	Diff input hysteresis		35		mV
R_{in}	Input impedance	5K	8K		Ω
V_{cm}	Common mode input	-12		+12	V
Vbias	Open input bias voltage		1.5		V

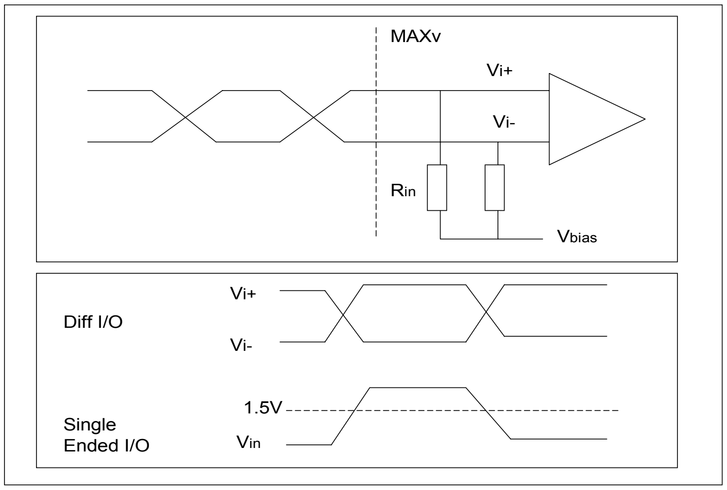


FIGURE 4-4 ENCODER AND INDEX DIFFERENTIAL INPUT DIAGRAM

Note: Single ended encoders with leads longer than 2 feet should be biased at 1.5 volts.

TABLE 4-5 RECOMMENDED ENCODER / INDEX INTERFACE COMPONENT VALUE

Symbol	Description	Min	Type	Max
R_S	Series termination resistor		100Ω	
Vbias	Single ended bias voltage		1.5 VDC	
V_{oh}	Driver high level voltage	3.0V		12V

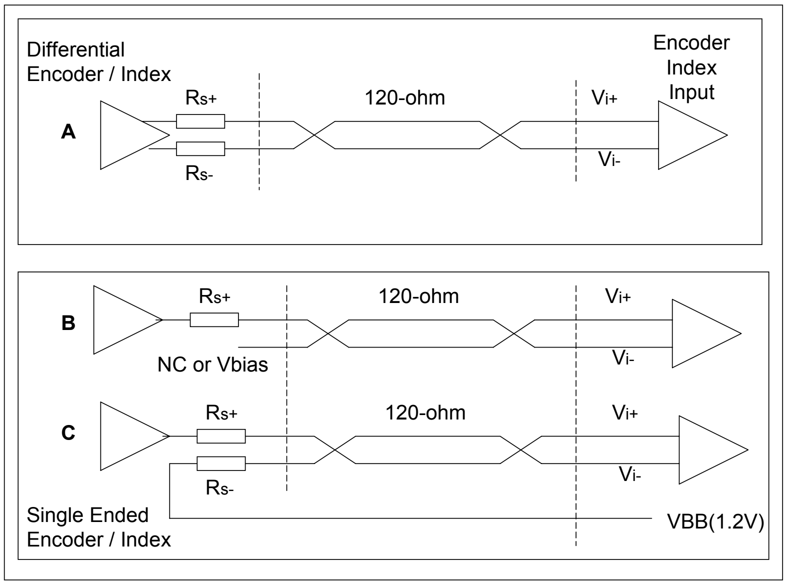


Figure 4-5 Encoder And Index Interface

Table 4-6 Servo Output DC Characteristics

Symbol	Parameter	Min	Type	Max	Unit
V_s12V+	12V Supply	11.6	12.0	15.0	V
V_s12V-	-12V Supply	-15.0	-12.0	-11.6	V
IL	Output current	5			mA
RL	Load impedance for ± 10V output	2K			Ω

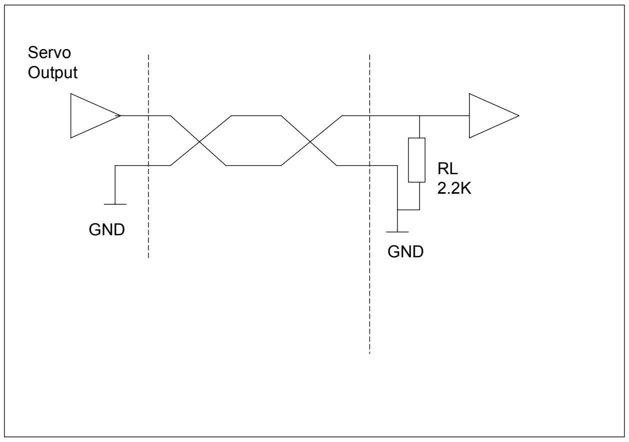


Figure 4-6 Servo Interface Diagram

Table 4-7 Home / Limit Input DC Characteristics

Symbol	Min	Type	Max	Unit
V_{ih}	3.0		15	V
V_{il}	-15		0.4	V
Zi		2.3K		Ω
V_{th}	1.3		2.7	V

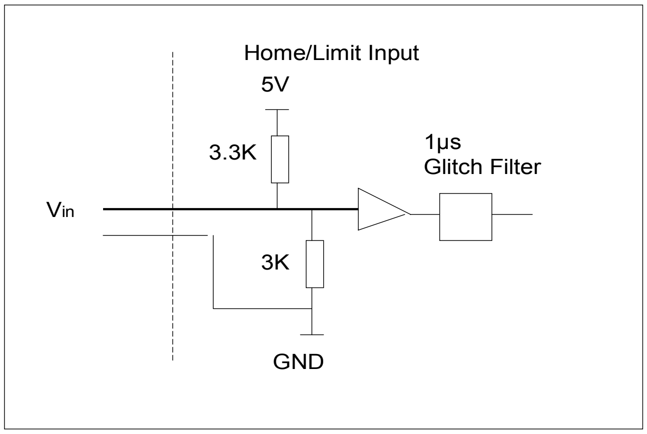


Figure 4-7 Home and Limit Input Diagram

TABLE 4-8 RECOMMENDED RELAY SWITCH INTERFACE COMPONENT VALUE

Symbol	Description	Min	Type	Max
Vcc	Relay voltage	3.0V		15V
R1	Pulse damping resistor		120Ω	220Ω
C1	Pulse damping capacitance		0.01μF	

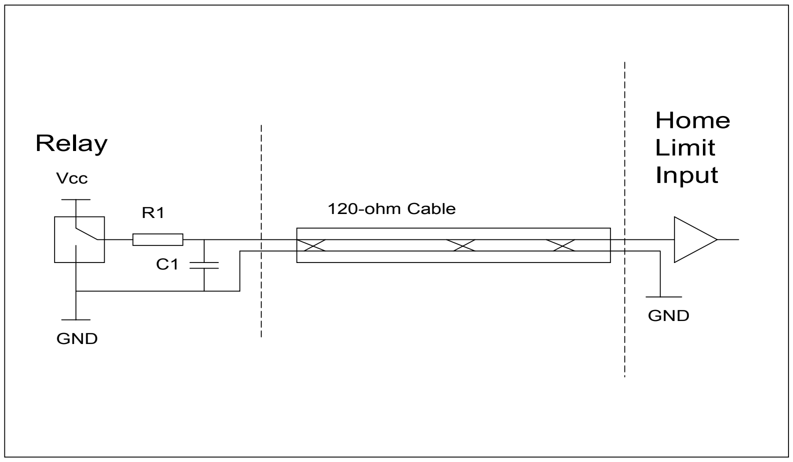


Figure 4-8 Relay Switch Interface Diagram

TABLE 4-9 RECOMMENDED ACTIVE DRIVER INTERFACE COMPONENT VALUE

Symbol	Description	Min	Type	Max
V_{ih}	Input high level voltage	3.0V		15V
V_{il}	Input low level voltage	-15V		0.4V
R1	Series termination resistor		100Ω	

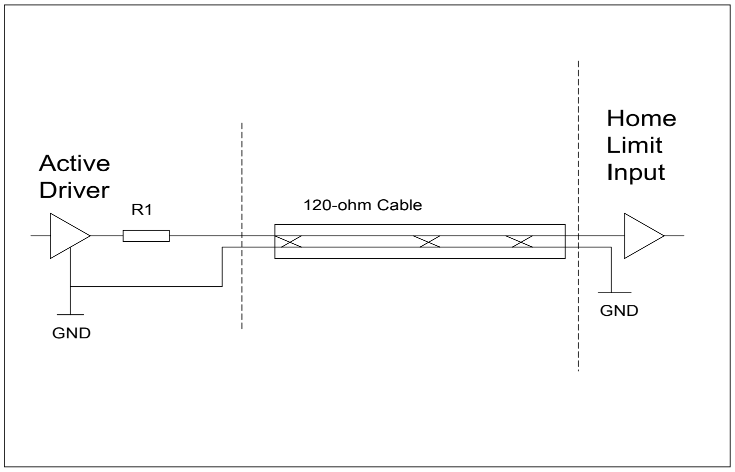


Figure 4-9 Active Drive Interface Diagram

4.4 ENCODER FEEDBACK

Incremental encoder feedback is provided for all servo axes and is optional for the stepper axes. The MAXv encoder feedback accepts quadrature pulse inputs from high resolution encoders at rates up to 16 MHz (after quadrature detection.) When used with stepper motors, the encoder monitors the actual position through the encoder pulse train. On servo axes, it continuously provides input to calculate the position error, adjust through the PID filter, and change the output accordingly. On the stepper axes, it can monitor the error and correct the position after the move is finished. The encoder input can also be used as an independent feedback source or in the encoder tracking mode to mimic an activity. All closed loop stepper axes are capable of slip or stall detection and encoder tracking with electronic gearing. These options are selectable by the user through software commands.

4.5 ENCODER SELECTION AND COMPATIBILITY

The MAXv is compatible with virtually any incremental encoder which provides quadrature outputs. Times four quadrature detection is used to increase resolution. This means that an encoder rated for 1000 counts (or lines) per revolution will result in 4000 pulses of quadrature encoded input for each motor shaft revolution. The inputs are compatible with encoders that have single ended or differential TTL outputs. The MAXv inputs have built in hysteresis to minimize the effects of noise pickup. The MAXv has differential line receivers to accommodate encoders with differential line driver outputs.

For short distances when single ended encoders are used, the unused negative inputs; i.e. Phase A-, Phase B-, etc. must be left open. If distances are longer then 2 feet, they must be biased with 1.5 VDC (see also Figure 4-4).

4.6 HOME PROCEDURES

Two logical input functionalities are provided to synchronize the physical hardware with the MAXv controller; i.e. put the controlled motor in the home position.

The home switch input is a 3-15 VDC level input signal to the MAXv. The general purpose home pull-up load is 2.2k Ohms and its “ON” impedance is about 10 Ohms.

The MAXv home switch input can be used to physically home a mechanical stage. When this functionality is used, the axis position counter will be reset to a selected value when the switch is activated. At this point, the MAXv can either ramp the axis to a stop or stop the axis immediately. To control of the direction of travel, the logic active state and the response to the active switch are controlled through commands. The duration of the home signal must be “UP” for at least two controller update cycles, to assure reliable detection.

The other homing method on the MAXv uses the home switch and the encoder signals to home a motor. When using the Home Encoder (HE) mode, the homing logic is used with these input signals. The home position consists of the logical AND of the encoder index pulse, the home switch input, and a single quadrant from the encoder logic. The home enable pulse must be true for less than one revolution of the encoder thus allowing only one home for the complete travel of the stage. The HH, HL and EH commands can be used to create different patterns for the home logic. The default home logic expressed in Boolean terms is:

Home = Phase +A * Phase -B * Index * Home Switch (Default)

It is necessary that the above quadrant occur within the index pulse as provided by the encoder for this logic to function properly. It may be necessary with some encoders to shift the phase of this quadrant by inverting one or both of the phases. Inverting one phase or swapping Phase A for Phase B will also reverse the direction. The encoder counter (read by a RE command) must increase for positive moves or the system will oscillate due to positive feedback. For other options, please contact OMS Technical Support. (See also Fig. 4-10, below).

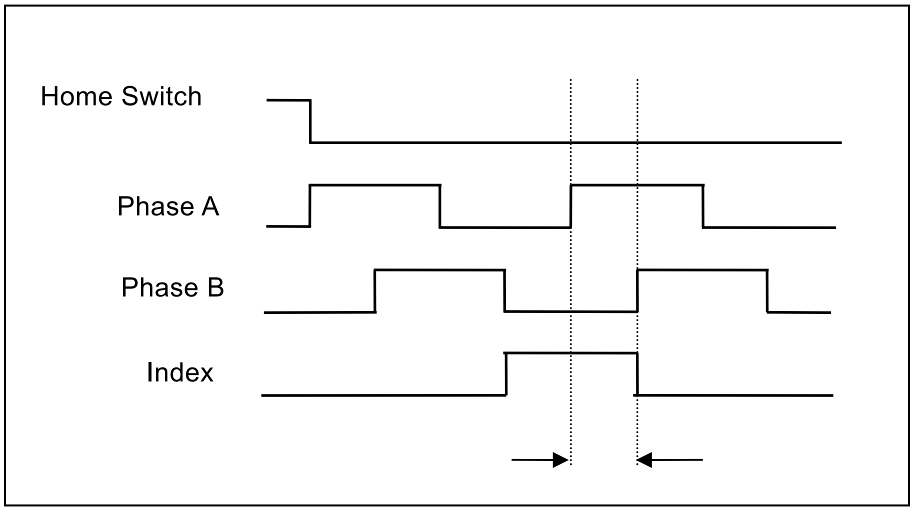


Figure 4-10 Encoder Homing State Detection

4.7 UNASSIGNED ENCODERS

The MAXv has two encoders that are not assigned to any axis, but can be used as an independent way to monitor and control complex motion profiles. These are designated encoders 8 and 9.

4.8 FRONT PANEL CONNECTORS

Table 4-10 Pin Out For Front Panel Connectors

50-Pin General Purpose I/O (J3)			
Pin	FUNCTION	Pin	FUNCTION
1	ADC0	26	ADC1
2	AGND	27	AGND
3	ADC2	28	ADC3
4	AGND	29	AGND
5	ADC4	30	ADC5
6	AGND	31	AGND
7	DAC8	32	DAC9
8	AGND	33	AGND
9	IO0/ Phase A8+	34	IO8
10	Phase A8-	35	GND
11	IO1/ Phase B8+	36	IO9
12	Phase B8-	37	GND
13	IO2/ Index 8+	38	IO10
14	Index 8-	39	GND
15	IO3	40	IO11
16	GND	41	GND
17	IO4/ Phase A9+	42	IO12
18	Phase A9-	43	GND
19	IO5 Phase B9+	44	IO13
20	Phase B9-	45	GND
21	IO6/ Index 9+	46	IO14
22	Index 9-	47	GND
23	IO7	48	IO15
24	GND	49	GND
25	5VDC	50	12VDC

68-Pin 4 AXIS LIST (J4)			
Pin	FUNCTION	Pin	FUNCTION
1	X Phase A+	35	X Index +
2	X Phase A -	36	X Index -
3	X Phase B +	37	X STEP
4	X Phase B -	38	GND
5	Y SERVO	39	X SERVO
6	AGND	40	AGND
7	Y Home	41	X Home
8	Y Dir	42	X Dir
9	Y Aux	43	X Aux
10	GND	44	GND
11	Y Pos Limit	45	X Pos Limit
12	Y Neg Limit	46	X Neg Limit
13	Y Phase A +	47	Y Index +
14	Y Phase A -	48	Y Index -
15	Y Phase B +	49	Y STEP
16	Y Phase B -	50	GND
17	+5V	51	V-BIAS
18	GND	52	GND
19	Z Phase A +	53	Z Index +
20	Z Phase A -	54	Z Index -
21	Z Phase B +	55	Z STEP
22	Z Phase B -	56	GND
23	T SERVO	57	Z SERVO
24	AGND	58	AGND
25	T Home	59	Z Home
26	T Dir	60	Z Dir
27	T Aux	61	Z Aux
28	GND	62	GND
29	T Pos Limit	63	Z Pos Limit
30	T Neg Limit	64	Z Neg Limit
31	T Phase A +	65	T Index +
32	T Phase A -	66	T Index -
33	T Phase B +	67	T STEP
34	T Phase B -	68	GND

68-Pin 4 AXIS LIST (J5)			
Pin	FUNCTION	Pin	FUNCTION
1	U Phase A+	35	U Index+
2	U Phase A-	36	U Index-
3	U Phase B+	37	U STEP
4	U Phase B-	38	GND
5	V SERVO	39	U SERVO
6	AGND	40	AGND
7	V Home	41	U Home
8	V Dir	42	U Dir
9	V Aux	43	U Aux
10	GND	44	GND
11	V Pos Limit	45	U Pos Limit
12	V Neg Limit	46	U Neg Limit
13	V Phase A +	47	V Index +
14	V Phase A -	48	V Index -
15	V Phase B +	49	V STEP
16	V Phase B -	50	GND
17	+5V	51	V-BIAS
18	GND	52	GND
19	R Phase A +	53	R Index +
20	R Phase A -	54	R Index -
21	R Phase B +	55	R STEP
22	R Phase B -	56	GND
23	S SERVO	57	R SERVO
24	AGND	58	AGND
25	S Home	59	R Home
26	S Dir	60	R Dir
27	S Aux	61	R Aux
28	GND	62	GND
29	S Pos Limit	63	R Pos Limit
30	S Neg Limit	64	R Neg Limit
31	S Phase A +	65	S Index +
32	S Phase A -	66	S Index -
33	S Phase B +	67	S STEP
34	S Phase B -	68	GND

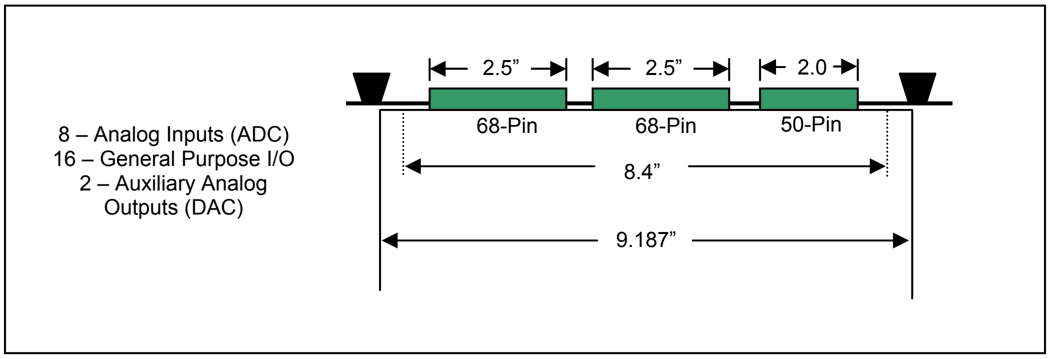


Figure 4-11 MAXv Front Panel Connector Pin Assignment

4.8.1 MAXv TO IOvMAX HOOK DIAGRAM

The hook up diagram is shown below. Note that the nomenclature for the axis can be alpha (x,y,z…) or numerical (0,1,2…)

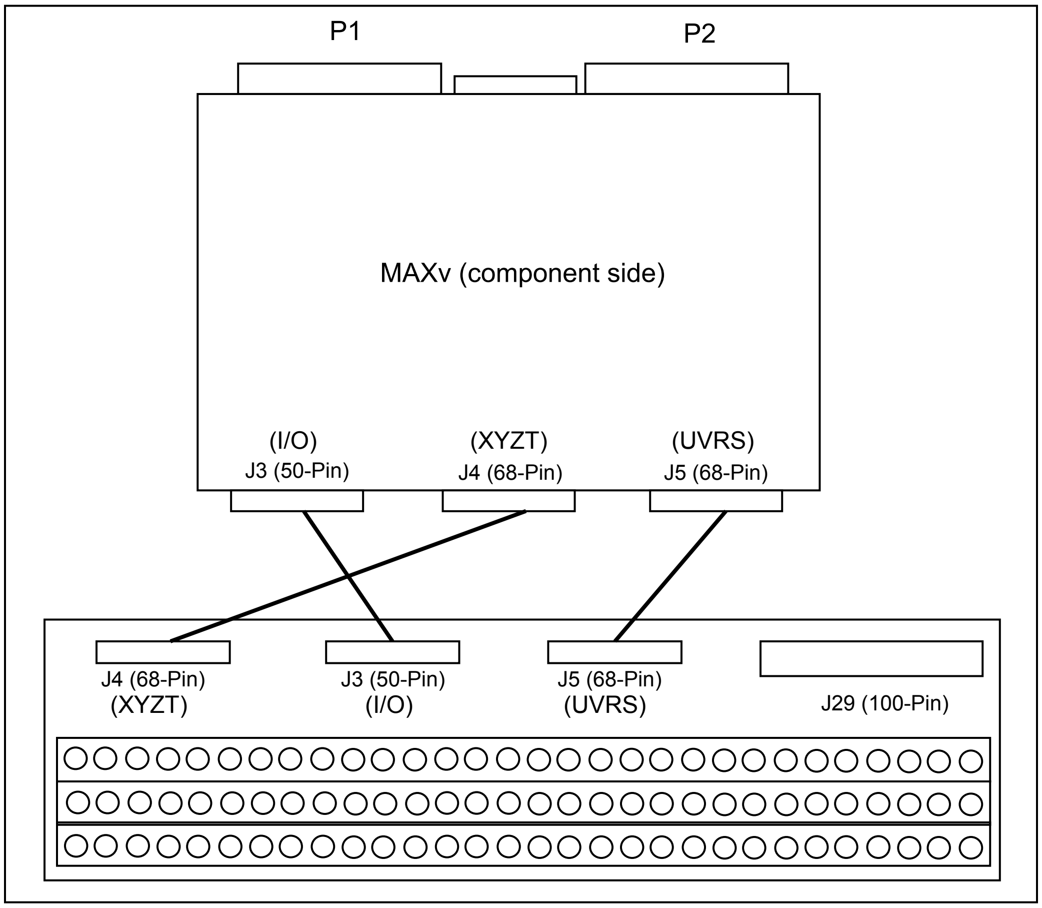


Figure 4-12 IOvMAX Break-Out to MAXv

NOTE:	**X Axis = 0**	**Y Axis = 1**	**Z Axis = 2**
	T Axis = 3	**U Axis = 4**	**V Axis = 5**
	R Axis = 6	**S Axis = 7**	
	Aux Encoder ‘A’ = 8 Aux Encoder ‘B’ = 9		

TABLE 4-11 P2 CONNECTOR PINOUT AT BACKPLANE
(ROWS A, B, & C ARE VME58 COMPATIBLE)

MAXv Pin Assignment (P2 Connector)					
Pin	Row Z	Row A	Row B	Row C	Row D
1	X Phase B -	X Phase B +	+5V	X Index +	X Index -
2	GND	X Step	GND	X Phase A +	X Phase A -
3	ADC0/IO8	X Pos LMT	RSVD	X Dir	X Aux
4	GND	X Neg LMT	A24	X Home	I/O0
5	Y Phase B -	Y Phase B +	A25	Y Index +	Y Index -
6	GND	Y Step	A26	Y Phase A +	Y Phase A -
7	ADC1/IO9	Y Pos LMT	A27	Y Dir	Y Aux
8	GND	Y Neg LMT	A28	Y Home	I/O1
9	Z Phase B -	Z Phase B	A29	Z Index +	Z Index -
10	GND	Z Step	A30	Z Phase A +	Z Phase A -
11	ADC2/IO10	Z Pos LMT	A31	Z Dir	Z Aux
12	GND	Z Neg LMT	GND	Z Home	I/O2
13	T Phase B -	T Phase B	+5V	T Index +	T Index -
14	GND	T Step	D16	T Phase A +	T Phase A -
15	ADC3/IO11	T Pos LMT	D17	T Dir	T Aux
16	GND	T Neg LMT	D18	T Home	I/O3
17	U Phase B -	U Phase B	D19	U Index +	U Index -
18	GND	U Step	D20	U Phase A +	U Phase A -
19	ADC4/IO12	U Pos LMT	D21	U Dir	U Aux
20	GND	U Neg LMT	D22	U Home	I/O4
21	V Phase B -	V Phase B	D23	V Index +	V Index -
22	GND	V Step	GND	V Phase A +	V Phase A -
23	ADC5/IO13	V Pos LMT	D24	V Dir	V Aux
24	GND	V Neg LMT	D25	V Home	I/O5
25	R Phase B -	R Phase B	D26	R Index +	R Index -
26	GND	R Step	D27	R Phase A +	R Phase A -
27	DAC8/IO14	R Pos LMT	D28	R Dir	R Aux
28	GND	R Neg LMT	D29	R Home	I/O6 / 5 AUX
29	S Phase B -	S Phase B	D30	S Index +	S Index -
30	GND	S Step	D31	S Phase A +	S Phase A -
31	DAC9/IO15	S Pos LMT	GND	S Dir	NC
32	GND	S Neg LMT	+5V	S Home	NC

4.9 IOvMAX ADAPTER MODULE

For ease of connection to the MAXv, OMS has developed the IOvMAX interface module. All three SCSI connectors on the IOvMAX connect directly to the appropriate 50-pin or 68-pin connectors on the MAXv (See FIGURE 4-12).

The IOvMAX has a 180-pin terminal block that provides an independent screw connection for each signal..

In addition the IOvMAX also has a 100-pin connector (J29) that is compatible with the OMS VME58 pin out. (See Figure 4-12 IOvMAX Breakout diagram).

TABLE 4-12 IOvMAX Terminal Block Pin-Out

Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name	Pin #	Signal Name
1	Bias (1.2V)	46	R Index A+	91	U Neg Limit	136	Z Servo
2	X Phase A+	47	R Index A-	92	GND	137	GND
3	X Phase A-	48	R Phase B+	93	U Ang	138	Z Step
4	X Index A+	49	R Phase B-	94	I/O 8	139	Z Dir
5	X Index A-	50	GND	95	GND	140	Z Aux
6	X Phase B+	51	S Phase A+	96	I/O 9	141	Z Home
7	X Phase B-	52	S Phase A-	97	V Pos Limit	142	5V
8	GND	53	S Index A+	98	V Neg Limit	143	T Servo
9	Y Phase A+	54	S Index A-	99	GND	144	GND
10	Y Phase A-	55	S Phase B+	100	V Ang	145	T Step
11	Y Index A+	56	S Phase B-	101	I/O 10	146	T Dir
12	Y Index A-	57	GND	102	GND	147	T Aux
13	Y Phase B+	58	pha_n8	103	I/O 11	148	T Home
14	Y Phase B-	59	GND	104	R Pos Limit	149	5V
15	GND	60	phb_n8	105	R Neg Limit	150	U Servo
16	Z Phase A+	61	GND	106	GND	151	GND
17	Z Phase A-	62	X Pos Limit	107	R Ang	152	U Step
18	Z Index A+	63	X Neg Limit	108	I/O 12	153	U Dir
19	Z Index A-	64	GND	109	GND	154	U Aux
20	Z Phase B+	65	X Ang	110	I/O 13	155	U Home
21	Z Phase B-	66	I/O 0	111	S Pos Limit	156	5V
22	GND	67	GND	112	S Neg Limit	157	V Servo
23	T Phase A+	68	I/O 1	113	GND	158	GND
24	T Phase A-	69	Y Pos Limit	114	S Ang	159	V Step
25	T Index A+	70	Y Neg Limit	115	I/O 14	160	V Dir
26	T Index A-	71	GND	116	GND	161	V Aux
27	T Phase B+	72	Y Ang	117	I/O 15	162	V Home
28	T Phase B-	73	I/O 2	118	indx_n8	163	5V
29	GND	74	GND	119	GND	164	R Servo
30	U Phase A+	75	I/O 3	120	indx_n9	165	GND
31	U Phase A-	76	Z Pos Limit	121	5v	166	R Step
32	U Index A+	77	Z Neg Limit	122	X Servo	167	R Dir
33	U Index A-	78	GND	123	GND	168	R Aux
34	U Phase B+	79	Z Ang	124	X Step	169	R Home
35	U Phase B-	80	I/O 4	125	X Dir	170	5V
36	GND	81	GND	126	X Aux	171	S Servo
37	V Phase A+	82	I/O 5	127	X Home	172	GND
38	V Phase A-	83	T Pos Limit	128	5V	173	S Step
39	V Index A+	84	T Neg Limit	129	Y Servo	174	S Dir
40	V Index A-	85	GND	130	GND	175	S Aux
41	V Phase B+	86	T Ang	131	Y Step	176	S Home
42	V Phase B-	87	I/O 6	132	Y Dir	177	5V
43	GND	88	GND	133	Y Aux	178	pha_n9
44	R Phase A+	89	I/O 7	134	Y Home	179	GND
45	R Phase A-	90	U Pos Limit	135	GND	180	phb_n9

NOTE: Aux Encoder 'A' = 8 and Aux Encoder 'B' = 9 See Table 4-10 for pin-outs of J3(I/O), J4(XYZT), and J5(UVRS) of the MAXv.

TABLE 4-13 VME58 Output Connector Pin List (J29) IOvMAX

FUNCTION	PINS		FUNCTION
User I/O 0	1	51	+5VDC
User I/O 2	2	52	User I/O 1
User I/O 4	3	53	User I/O 3
User I/O 6	4	54	User I/O 5
User I/O 8	5	55	User I/O 7
User I/O 10	6	56	User I/O 9
User I/O 12	7	57	User I/O 11
User I/O 13	8	58	Ground
Analog Ground	9	59	+5VDC
X Phase A	10	60	Ground
X Phase B	11	61	X Index
X Direction	12	62	X Axis Step Output
X Auxiliary Output	13	63	X Positive Limit
X Home	14	64	X Negative Limit
Y Phase A	15	65	Y Index
Y Phase B	16	66	Y Axis Servo Output
Y Direction	17	67	Y Positive Limit
Y Auxiliary Output	18	68	Y Negative Limit
Y Home	19	69	+5VDC
Analog Ground	20	70	Ground
Z Phase A	21	71	Z Index
Z Phase B	22	72	Z Axis Servo Output
Z Direction	23	73	Z Positive Limit
Z Auxiliary Output	24	74	Z Negative Limit
Z Home	25	75	T Index
T Phase A	26	76	T Axis Servo Output
T Phase B	27	77	T Auxiliary Output
T Direction	28	78	T Positive Limit
T Home	29	79	T Negative Limit
Analog Ground	30	80	+5VDC
U Phase A	31	81	Ground
U Phase B	32	82	U Index
U Direction	33	83	U Axis Step Output
U Auxiliary Output	34	84	U Positive Limit
U Home	35	85	U Negative Limit
V Phase A	36	86	V Index
V Phase B	37	87	V Axis Step Output
V Direction	38	88	V Positive Limit
V Auxiliary Output	39	89	V Negative Limit
V Home	40	90	+5VDC
Analog Ground	41	91	Ground
R Phase A	42	92	R Index
R Phase B	43	93	R Axis Step Output
R Direction	44	94	R Positive Limit
R Auxiliary Output	45	95	R Negative Limit
R Home	46	96	S Index
S Phase A	47	97	S Axis Step Output
S Phase B	48	98	S Auxiliary Output
S Direction	49	99	S Positive Limit
S Home	50	100	S Negative Limit

5. COMMAND STRUCTURE

5.1 INTRODUCTION

An extensive command structure is built into the MAXv family of intelligent motor controllers. The commands consist of two or three ASCII characters and may be in upper or lower case. Some of the commands expect a numerical operand to follow. These commands are identified with a “#” after the command. The operand must be terminated by a space, carriage return, or semi-colon to indicate the end of the operand list. No terminator is required on the other commands, but it is strongly recommended it be included to improve readability. The operand must immediately follow the command with no space or separation character. The “#” indicates a signed integer input parameter or a signed fixed point number of the format “##.#” when user units are enabled. With user units enabled, distance, velocity, and acceleration parameters may be entered in units such as inches, revolutions, etc.

Most commands are usable in both single-axis and multi-axis modes. Those that require a numeric parameter in single-axis mode require multiple numeric parameters of the same type in multi-axis modes. For example, the MR (Move Relative) command takes a distance as a numeric parameter and is formatted as “MR#;” in single-axis modes. Multi-axis modes have a parameter position for each axis and must be formatted as “MR#,#,#,#,#,#,#,#;” in an 8-axis system. (Note: Use of commas “,” between axes.) Any “#” parameter may be omitted for any axis which is not to be affected by the command and the command may be terminated prematurely with a semicolon. For example, to move only the Y and Z axes, enter the command as “MR,#,#;”.

Some commands that are usable in both single-axis and multi-axis modes do not take a parameter in single-axis mode. These commands require numeric parameters in multi-axis modes, and the parameters indicate whether of not to take action for each axis. If a parameter exists for an axis, then the command affects that axis and if the parameter does not exist for that axis, then the command has no affect on that axis. For example, the single-axis format of LN (Limits On) is simply "LN" without any parameters of any kind. The multi-axis format of LN is "LNb,b,b,b,b,b,b,b;" for 8-axis systems where 'b' represents the parameter for the corresponding axis. Like other multi-axis commands, a 'b' parameter may be omitted if that axis is to remain unchanged and command may be prematurely terminated with a semicolon. Each 'b' position, if used, can be any numeric value. For example, to enable the Y and Z axes limit switches and leave the X and T axes unchanged, send the command "LN,1,100;". The 1 and 100 parameters could be any numeric value whatsoever, and the effect of the command would be the same. For example, the following commands are equivalent:

"LN,1,1;"

"LN,0,0;"

"LN,50,99;"

Synchronized moves may be made by entering the AA or AM command. These commands perform a context switch which allows entering motion commands in the format MRx#,y#,z#,t#,u#,v#,r#,s#;. Numbers are entered for each axis which is to be commanded to move. An axis may be skipped by entering a comma with no parameter. The command may be prematurely terminated with a “;” i.e. a move requiring only the X and Y axes would use the command MRx#,y#; followed by the GO command. Each axis programmed to move will start together upon executing the GO command. The MAXv can be switched back to the unsynchronized mode by entering the desired single axis command such as AX.

The AM command is provided for complex applications where the host manages multiple motion processes by a multitasking operating system. This mode shares the same instructions as the AA mode, but allows starting a task while some other task involving one or more axes is active. For example, the X and Y axes could be doing linear interpolation while the Z axis is making an unrelated move simultaneously.

Constant velocity contouring provides another mode wherein the move parameters are predefined by entering AA (or AM) then CD#,#;. The MAXv will then calculate the move profile in advance and move at a constant velocity in the prescribed pattern. It can do linear interpolation on as many as 8 axes between the predefined points and it can do circular interpolation mixed with linear on any two axes.

5.2 QUEUES

The input characters are placed in a character buffer on input then removed and interpreted. The commands are then placed in separate command and argument queues for each axis. The command queues contain the commands, while the argument queues contain the operands for the commands. For example, in the string "AX; MR100; GO;" the X axis command queues would get "AX; MR; GO;" and the X axis argument queue would get "100". As they are executed the space is reclaimed allowing the host to pass commands ahead of the moves actually being processed. Most of the commands are placed in the appropriate command and argument queues for execution, while others are executed immediately allowing return of status information in a timely way rather than when encountered in the command stream. This information is provided in a table for each command which shows the queue requirements, if any, and indicates immediate in those cases where the command is not queued. The queue requirements shown in the tables are typical. Depending on the circumstances in which the command is issued, the actual queue requirement may vary slightly. The single axis cases are indicated by the mode reference indicating the appropriate axis. The synchronized mode is indicated by the mode identifier AA/AM. The contouring case is indicated by AA/CD for multiple axes in contour definition mode. The RQC and RQA commands may be used to determine the actual queue space available at any time. The queues operate independently allowing each axis to perform separate processes concurrently. The synchronized modes (AA) insert special wait opcodes which allow the axes to be synchronized in this mode. When the commands are nested within loops, the queue space is not reclaimed until after the loop has been executed the programmed number of times. For loops larger than the queue space, the loop may never be completed since it cannot reclaim the queue space and cannot accept the loop terminator. Therefore, loops are effectively limited in size by the size of the command queue. The current axis command queue size for MAXv is 2559, and the axis argument queue size for MAXv is 12799. Note that if either queue is overrun the MAXv will stall and the communication link will be lost. There is just one contour queue on MAXv to handle all axes commands while in contour mode. The size of the contour queue is 32763.

Some commands are valid only for stepper axes, others for stepper axes with encoder feedback, and still others for servo axes. Most are valid for all three types or some combination of types. A set of symbols to the right of each command identifies which motor types with which each command may be used. The symbols' meanings are as follows:

Stepper motor without an encoder (open loop)

Stepper motor with an encoder (closed loop)

Servo motor

If a command is usable with one of these motor types, the symbol will appear in black. If the command is not usable with a motor type, that motor symbol will be displayed in gray

This command is not usable with servo motors

Indicates an example.

The following commands are available in firmware revision 1.28 and above.

5.3 COMMAND SUMMARY

The following commands are included in the MAXv family of motor controllers. The '#' indicates a signed integer input parameter or a signed fixed point number of the format ##.# when user units are enabled. With User Units enabled, distances, velocity and acceleration parameters may be input in inches, revolutions, etc. Note that numeric parameters must be within the range of a 32-bit signed integer (2147483647 to -2147483648). For fixed point numeric parameters, the value without the decimal point must be within the range of a signed 32-bit integer. Entering parameter values outside the range of a signed 32-bit integer will cause a Command Error.

ALPHABETICAL COMMAND SUMMARY			
COMMAND	PAGE #	Q = QUERY C=CMD	COMMAND DESCRIPTION
AA	5-39	C	Any following commands are for the AA (All Axes) mode
?AB	5-40	Q	Report Auxiliary bit state
AC	5-41	C	Acceleration, set acceleration/deceleration register
?AC	5-42	Q	Report AC command
?AD	5-42	Q	Report Auxiliary bit state
ADH	5-43	C	Set Auxiliary default to high
ADL	5-44	C	Set Auxiliary default to low
AEL	5-45	C	Load Auxiliary Encoder Position
AER	5-46	Q	Report Auxiliary Encoder Position
AF	5-47	C	Auxiliary off
AI	5-48	C	Analog Input
AJ	5-49	C	Custom S-curve Definition
?AJ	5-52	Q	Report Custom S-curve Parameters
AM	5-53	C	Axes multitasking mode
AN	5-54	C	Auxiliary on
AO	5-55	C	Set analog input zero offset
APB	5-56	C	Archive current parameters in back up of flash
APP	5-57	C	Make the current parameter set the power up default values
?AQ	5-58	Q	Query current axis
AR	5-58	C	Any following commands are for the R axis
AS	5-59	C	Any following commands are for the S axis
AT	5-59	C	Any following commands are for the T axis
AU	5-60	C	Any following commands are for the U axis
AV	5-60	C	Any following commands are for the V axis
AX	5-61	C	Any following commands are for the X axis (default on reset)
AY	5-61	C	Any following commands are for the Y axis
AZ	5-62	C	Any following commands are for the Z axis

ALPHABETICAL COMMAND SUMMARY			
COMMAND	PAGE #	Q = QUERY C=CMD	COMMAND DESCRIPTION
BC	5-63	C	Set Backlash Compensation
?BC	5-63	Q	Report Backlash Compensation
BD	5-64	C	Set the direction of the general purpose I/O bits
BH	5-65	C	Set selected I/O bit high (off)
BI	5-66	C	Bipolar, set the analog torque outputs to bipolar
BL	5-67	C	Set selected I/O bit low (on)
BR	5-68	C	Define the state of the general output bits at power-up and reset
?BR	5-68	Q	Report the state of the general output bits at power-up and reset
BS	5-69	C	Set all bits of the general purpose output port to the state specified by the hex argument.
BW	5-70	C	Wait for input to go low
BX	5-70	Q	Return bit status in hex format
CA	5-71	C	Clear done flag of currently addressed axis
CD	5-72	C	Define a contour
CE	5-74	C	End contour definition, ramp to a stop
CG	5-75	C	Execute a constant velocity contour and prevent MT parsing
CK	5-77	C	End contour definition, immediately stop step pulses
CN	5-78	C	Cosine on, enable cosine velocity profiles
CR	5-79	C	Circular interpolation, move in a circle
CV	5-80	C	Contouring velocity, definition
CW	5-80	C	Clear while
CX	5-81	C	Contour execute
?DA	5-82	Q	Print a custom ramp
DAB	5-83	C	Define custom ramp breakpoint
DAE	5-84	C	End custom ramp definition
DAR	5-85	C	Begin custom ramp definition
?DB	5-86	Q	Report direction bit logic
DBI	5-87	C	Invert direction bit
DBN	5-88	C	Normalize direction bit
DC	5-89	C	Set the deceleration rate that will be used by the GU command
?DC	5-90	Q	Report Deceleration Rate
?DE	5-90	Q	Report an acceleration ramp definition table entry
DOV	5-91	C	Output Auxiliary DAC Voltage
DOZ	5-92	C	Set Auxiliary DAC Zero Offset
?DS	5-93	Q	Report the size of a custom acceleration ramp table
EA	5-94	Q	Encoder status, return encoder status of currently addressed axis
EH	5-95	C	Defining Encoder Home
?EH	5-96	Q	Query Encoder Home

ALPHABETICAL COMMAND SUMMARY			
COMMAND	PAGE #	Q = QUERY C=CMD	COMMAND DESCRIPTION
ER	5-97	C	Encoder ratio, set encoder count to motor count ratio
?ER	5-98	Q	Report motor:encoder ratio
ES	5-99	C	Encoder slip tolerance, set tolerance before slip or stall is flagged
?ES	5-100	Q	Report encoder slip tolerance
ET	5-101	C	Encoder tracking, set encoder tracking mode
FL	5-102	C	Flush an axis queue
FP	5-103	C	Force position, flush queue and attempt to stop at specified position
FX	5-103	C	Enable axis gantry mode
GD	5-104	C	Go and reset done flags
GN	5-106	C	Go and notify when done
GO	5-107	C	Go command, start execution of motion
GS	5-108	C	Go and use the home switch to monitor for motor slip
GU	5-109	C	Go and use the AC values to accelerate and the DC values to decelerate
HD	5-110	C	Hold deadband, specify deadband tolerance for position hold
?HD	5-110	Q	Report position maintenance deadband
HE	5-111	C	Encoder home mode, set home on encoder logic
HF	5-112	C	Disable PID
HG	5-113	C	Hold gain, specify position hold gain parameter
?HG	5-113	Q	Report position maintenance gain
HH	5-114	C	Home high, home switches are active high
HL	5-115	C	Home low, home switches are active low
HN	5-119	C	Enable PID
HM	5-116	C	Home, find home and initialize the position counter
?HM	5-118	Q	Report home state selection
HR	5-121	C	Home reverse, find home in reverse direction and initialize position counter
HS	5-123	C	Home switch, enable home switch mode
?HS	5-123	Q	Report home switch true state selection
HV	5-124	C	Hold velocity, specify maximum position hold correction velocity
?HV	5-125	Q	Report position maintenance velocity
IC	5-126	C	Interrupt clear, clear done interrupt status and error flags
ID	5-127	C	Interrupt host when done and set done flag
II	5-128	C	Interrupt independent
IN	5-129	C	Interrupt when nearly done
IO	5-130	C	I/O Bit Direction Selection
IP	5-131	C	Interrupt when in position
IS	5-132	C	Interrupt slip, interrupts host on slip or stall detection

ALPHABETICAL COMMAND SUMMARY			
COMMAND	PAGE #	Q = QUERY C=CMD	COMMAND DESCRIPTION
JF	5-133	C	Jog at fractional rates
JG	5-134	C	Jog command, run motor at specified velocity until a new velocity command is sent or it is stopped by a stop or kill command
KA	5-136	C	Acceleration Feedforward coefficient, used in tuning servo systems
?KA	5-136	Q	Report acceleration feed-forward
KB	5-137	C	PID Upper Bound Limit Coefficient
?KB	5-137	Q	Report axis PID upper Bound limit
KD	5-138		Derivative Gain coefficient, used for PID filter
?KD	5-138	Q	Report PID derivative gain
KF	5-139	C	Set servo axis PID friction coefficient
?KF	5-139	Q	Report servo axis friction offset
KI	5-140	C	Integral Gain coefficient, used for PID filter
?KI	5-140	Q	Report PID integral gain
KL	5-141	C	Kill, flush queue and terminate pulse generation immediately on all axes without decelerating
KM	5-142	C	Home and kill pulse generation
KO	5-143	C	Offset coefficient
?KO	5-143	Q	Report PID closed- loop offset
KP	5-144	C	Proportional Gain coefficient, used for PID filter
?KP	5-144	Q	Report PID proportional gain
KR	5-145	C	Home in reverse and kill pulse generation
KS	5-146	C	Kill selected axes
KU	5-147	C	PID integration sum upper limit
?KU	5-147	Q	Report PID integration sum upper limit
KV	5-148	C	Velocity Feedforward coefficient, used in tuning servo systems
?KV	5-148	Q	Report velocity feedforward
LA	5-149	C	Linear ramp selection per axis
LE	5-150	C	Loop end, terminate most recent LS command
LF	5-151	C	Disable limit switches for selected axis
LH	5-152	C	Limit high, limit switch is active high
LL	5-153	C	Limit low, limit switch is active low
?LM	5-154	Q	Report axis overtravel enable/disable
LN	5-155	C	Enable limit switches for selected axis
LO	5-156	C	Set axis motor position but not the encoder position
LP	5-157	C	Load position, load position counter with parameter
LS	5-158	C	Loop start, set loop counter, from 1 to 2,147,483,647 loops; (may be nested to 4 levels)
?LS	5-160	Q	Report limit active state
MA	5-161	C	Move absolute, move to absolute position

ALPHABETICAL COMMAND SUMMARY

COMMAND	PAGE #	Q = QUERY C=CMD	COMMAND DESCRIPTION
MD	5-163	C	Define a temporary macro
ML	5-164	C	Move linear, move specified distance relative from current position
MM	5-166	C	Move minus, set minus direction for MV type move
MO	5-167	C	Move one pulse in current direction
MP	5-167	C	Move plus, set positive direction for MV type move
MR	5-168	C	Move relative, move specified distance from current position
MT	5-170	C	Move to, move to specified position
MV	5-171	C	Move velocity, move to first parameter (absolute position) at second parameter velocity without stopping at end of move
MX	5-173	C	Execute a macro command string
NV	5-173	C	Set a new contour velocity from with in a contour
PA	5-174	C	Power automatic, turn power on before each move and off after the move
?PA	5-175	Q	Report power automatic state
PE	5-176	Q	Report encoder positions of all encoder and servo axes
PF	5-176	C	Parabolic off, disable parabolic ramps; i.e. linear ramps will be generated
PH	5-177	C	Power high
PL	5-177	C	Power low
PM	5-178	C	Print a macro command string
?PM	5-179	Q	Report hold state
PN	5-180	C	Parabolic on, enable parabolic ramps
PP	5-181	Q	Report motor positions of all axes
PR	5-182	C	Parabolic ramp selection per axis
?PS	5-183	Q	Report Motor Type
PSE	5-184	C	Configure the axis as a stepper axis with encoder
PSM	5-184	C	Configure current axis as a servo axis
PSO	5-185	C	Configure current axis as an open-loop stepper
PT	5-185	C	Preserve a temporary macro
QA	5-186	Q	Query status of switches and flags for addressed axis without affecting flags
QI	5-186	Q	Query status of switches and flags on all axes without affecting flags
QL	5-187	Q	Query all limit sensors
RA	5-188	Q	Return status of switches and flags and reset flags
RB	5-189	Q	Report bit direction
RC	5-189	Q	Return current acceleration or deceleration of the current axis
RDB	5-190	C	Restore default parameter in back up space
RDF	5-191	C	Restore the current parameter set to the factory default values
RDP	5-192	C	Restore power up defaults
RE	5-193	Q	Request encoder position, return current encoder position

ALPHABETICAL COMMAND SUMMARY			
COMMAND	PAGE #	Q = QUERY C=CMD	COMMAND DESCRIPTION
RI	5-193	C	Return status of switches and flags for all axes and reset flags
RL	5-194	Q	Return slip status of each axis
RM	5-195	Q	Return remainder of position divided by parameter in position counter
RP	5-196	Q	Request position, returns current position
RQ	5-197	Q	Report Contour Queue Size
RQA	5-197	Q	Report axis argument queue
RQC	5-198	Q	Report axis command queue
?RT	5-199	Q	Report ramp type
RU	5-200	Q	Return current position in user units
RV	5-201	Q	Return current velocity at which the axis is moving
SA	5-201	C	Flush queue and stops all axes with deceleration
SC	5-202	C	Cosine ramp selection per axis
SD	5-203	C	Stop all axes and clear any done flags
SE	5-204	C	Settling time
?SE	5-204	Q	Report settling time
SF	5-205	C	Soft limit off, restore normal overtravel operation
SI	5-206	C	Stop selected motors
?SK	5-207	Q	Report Axis Slip Kill mode selection
SL	5-208	C	Soft limit mode, allow pulse train to ramp down on overtravel
?SL	5-209	Q	Report soft limit status
SO	5-210	C	Stop at a designated position using a specified ramp down distance
?SO	5-211	Q	Report analog output mode
SP	5-211	C	Stop at position, stop at specified position if possible after all commands have been executed
SR	5-212	C	Select a custom acceleration ramp for use by the currently active axis
SS	5-212	C	Selects a predefined S-curve accelleration
ST	5-213	C	Stop, flush and decelerate queue to stop
?SV	5-214	Q	Report servo voltage invert selection
SVI	5-215	C	Invert servo voltage
SVN	5-216	C	Normalize servo voltage
SW	5-217	C	Sync wait, wait for the input bit to be released (non-active)
TF	5-219	C	Turn encoder slip kill off
TL	5-220	C	Set software travel limits
?TL	5-221	Q	Report software overtravel ranges
TM	5-222	C	Jog at the current velocity for the specified number of milliseconds
TN	5-223	C	Turn encoder slip kill on
TX	5-224	C	Track the X axis
UF	5-225	C	User units off, turn off user unit translation

ALPHABETICAL COMMAND SUMMARY			
COMMAND	PAGE #	Q = QUERY C=CMD	COMMAND DESCRIPTION
UN	5-226	C	Unipolar, set the analog torque outputs to unipolar
#UR	5-227	C	Set update rate
?UR	5-226	Q	Report the controller's motor update rate
UU	5-228	C	User units, multiply acceleration, velocity and distance parameters by specified parameter
?UU	5-229	Q	Report axis user units' axis assignment
VB	5-230	C	Base velocity, set base velocity
?VB	5-231	Q	Report axis base velocity
VL	5-232	C	Set maximum velocity to be used in profile
?VL	5-233	Q	Report axis velocity limit
VS	5-234	C	Velocity stream, slave velocity mode for profiling
WA	5-235	C	Wait until all moves on all axes are finished
WD	5-236	C	While end, WS loop terminator
WG	5-236	C	Terminate WH loop
WH	5-237	C	While, execute all commands until WG loop terminator, until flag cleared by CW command
WQ	5-239	C	Wait until current axis queue is empty
WS	5-240	C	While sync, execute while sync of input is true
WT	5-241	C	Wait, wait for specified number of milliseconds
WY	5-241	Q	Who are you , returns model and software revision

COMMAND SUMMARY BY SECTION		
COMMAN D	PAGE#	COMMAND DESCRIPTION
IDENTIFICATION COMMANDS – SECTION 5.4.1		
?AB	5-40	Report Auxiliary bit state
?AC	5-42	Report AC command
?AD	5-42	Report Auxiliary bit state
?AJ	5-52	Report Custom S-curve Parameters
?AQ	5-58	Query current axis
?BC	5-63	Report Backlash Compensation
?BR	5-68	Report the state of the general output bits at power-up and reset
?DA	5-82	Print a custom ramp
?DB	5-86	Report direction bit logic
?DC	5-90	Report Deceleration Rate
?DE	5-90	Report an acceleration ramp definition table entry
?EH	5-96	Query Encoder Home
?ER	5-98	Report motor:encoder ratio
?ES	5-100	Report encoder slip tolerance
?HD	5-110	Report position maintenance deadband
?HG	5-113	Report position maintenance gain
?HM	5-118	Report home state selection
?HS	5-123	Report home switch true state selection
?HV	5-125	Report position maintenance velocity
?KA	5-136	Report acceleration feed-forward
?KB	5-137	Report AXIS PID upper Bound limit
?KD	5-138	Report PID derivative gain
?KF	5-139	Report servo axis friction offset
?KI	5-140	Report PID integral gain
?KO	5-143	Report PID closed- loop offset
?KP	5-144	Report PID proportional gain
?KU	5-147	Report PID integration sum upper limit
?KV	5-148	Report velocity feedforward
?LM	5-154	Report axis overtravel enable/disable
?LS	5-160	Report limit active state
?PA	5-175	Report power automatic state
?PM	5-179	Report hold state
?PS	5-183	Report Motor Type
?RT	5-199	Report ramp type
?SE	5-204	Report settling time
?SK	5-207	Report Axis Slip Kill mode selection
?SL	5-209	Report soft limit status
?SO	5-211	Report analog output mode

COMMAND SUMMARY BY SECTION		
COMMAN D	PAGE#	COMMAND DESCRIPTION
?SV	5-214	Report servo voltage invert selection
?TL	5-221	Report software overtravel ranges
?UR	5-226	Report the controller's motor update rate
?UU	5-229	Report axis user units' axis assignment
?VB	5-231	Report axis base velocity
?VL	5-233	Report axis velocity limit
WY	5-241	Who are you , returns model and software revision
POWER-UP DEFAULTS – SECTION 5.4.2		
APB	5-56	Archive current parameters in back up of flash
APP	5-57	Make the current parameter set the power up default values
?AQ	5-58	Query current axis
RDB	5-190	Restore default parameter in back up space
RDF	5-191	Restore the current parameter set to the factory default values
RDP	5-192	Restore power up defaults
QUEUE SELECTION COMMANDS – SECTION 5.4.3		
AA	5-39	Any following commands are for the AA (All Axes) mode
AM	5-53	Axes multitasking mode
AR	5-58	Any following commands are for the R axis
AS	5-59	Any following commands are for the S axis
AT	5-59	Any following commands are for the T axis
AU	5-60	Any following commands are for the U axis
AV	5-60	Any following commands are for the V axis
AX	5-61	Any following commands are for the X axis (default on reset)
AY	5-61	Any following commands are for the Y axis
AZ	5-62	Any following commands are for the Z axis
QL	5-187	Query all limit sensors
QUEUE STATUS COMMANDS – SECTION 5.4.4		
FL	5-102	Flush an axis queue
RQ	5-197	Report Contour Queue Size
RQA	5-197	Report Axis argument queue
RQC	5-198	Report Axis command queue
USER UNIT COMMANDS – SECTION 5.4.5		
UF	5-225	User units off, turn off user unit translation
?UR	5-226	Report the controller's motor update rate
UU	5-228	User units, multiply acceleration, velocity and distance parameters by specified parameter
?UU	5-229	Report axis user units' axis assignment

AXIS STATUS COMMANDS – SECTION 5.4.6		
EA	5-94	Encoder status, return encoder status of currently addressed axis
PSE	5-184	Configure the axis as a stepper axis with encoder
PSM	5-184	Configure current axis as a servo axis
PSO	5-185	Configure current axis as an open-loop stepper
QA	5-186	Query status of switches and flags for addressed axis without affecting flags
QI	5-186	Query status of switches and flags on all axes without affecting flags
RA	5-188	Return status of switches and flags and reset flags
RI	5-193	Return status of switches and flags for all axes and reset flags
MACROS – **SECTION 5.4.7**		
MD	5-163	Define a temporary macro
MX	5-173	Execute a macro command string
PM	5-178	Print a macro command string
PT	5-185	Preserve a temporary macro
AUXILIARY CONTROL COMMANDS – **SECTION 5.5.1**		
?AD	5-42	Report Auxiliary bit state
ADH	5-43	Set Auxiliary default to high
ADL	5-44	Set Auxiliary default to low
AN	5-54	Auxiliary on
PA	5-174	Power automatic, turn power on before each move and off after the move
?PA	5-175	Report power automatic state
PH	5-177	Power high
PL	5-177	Power low
SE	5-204	Settling time
?SE	5-204	Report settling time
GENERAL PURPOSE I/O CONTROL – **SECTION 5.5.2**		
BD	5-64	Set the direction of the general purpose I/O bits
BH	5-65	Set selected I/O bit high (off)
BL	5-67	Set selected I/O bit low (on)
?BR	5-68	Report the state of the general output bits at power-up and reset
BR	5-68	Define the state of the general output bits at power-up and reset
BS	5-69	Set all bits of the general purpose output port to the state specified by the hex argument.
BX	5-70	Return bit status in hex format
IO	5-130	I/O Bit Direction Selection
RB	5-189	Report bit direction

HOME CONTROL COMMANDS – SECTION 5.5.3		
EH	5-95	Defining Encoder Home
?EH	5-96	Query Encoder Home
HE	5-111	Encoder home mode, set home on encoder logic
HH	5-114	Home high, home switches are active high
HL	5-115	Home low, home switches are active low
HS	5-123	Home switch, enable home switch mode
LIMIT CONTROL COMMANDS – SECTION 5.5.4		
LF	5-151	Disable limit switches for selected axis
LH	5-152	Limit high, limit switch is active high
LL	5-153	Limit low, limit switch is active low
?LM	5-154	Report axis overtravel enable/disable
LN	5-155	Enable limit switches for selected axis
?LS	5-160	Report limit active state
QL	5-187	Query all limit sensors
SF	5-205	Soft limit off, restore normal overtravel operation
?SK	5-207	Report Axis Slip Kill mode selection
SL	5-208	Soft limit mode, allow pulse train to ramp down on overtravel
?SL	5-209	Report soft limit status
TL	5-220	Set software travel limits
?TL	5-221	Report software overtravel ranges
ANALOG I/O COMMANDS – SECTION 5.5.5		
AI	5-48	Analog Input
AO	5-55	Set analog input zero offset
DOV	5-91	Output Auxiliary DAC Voltage
DOZ	5-92	Set Auxiliary DAC Zero Offset
SERVO VOLTAGE CONTROL COMMANDS – SECTION 5.6.1		
BI	5-66	Bipolar, set the analog torque outputs to bipolar
?DB	5-86	Report direction bit logic
DBI	5-87	Invert direction bit
DBN	5-88	Normalize direction bit
?DC	5-90	Report Deceleration Rate
KO	5-143	Offset coefficient
?KO	5-143	Report PID closed- loop offset
?SO	5-211	Report analog output mode
?SV	5-214	Report servo voltage invert selection
SVI	5-215	Invert servo voltage
SVN	5-216	Normalize servo voltage
UN	5-226	Unipolar, set the analog torque outputs to unipolar

PID COMMANDS – SECTION 5.6.2		
HF	5-112	Disable PID
HN	5-119	Enable PID
KA	5-136	Acceleration Feedforward coefficient, used in tuning servo systems
?KA	5-136	Report acceleration feed-forward
KB	5-137	PID Upper Bound Limit Coefficient
?KB	5-137	Report Axis PID upper Bound limit
KD	5-138	Derivative Gain coefficient, used for PID filter
?KD	5-138	Report PID derivative gain
KF	5-139	Set servo axis PID friction coefficient
?KF	5-139	Report servo axis friction offset
KI	5-140	Integral Gain coefficient, used for PID filter
?KI	5-140	Report PID integral gain
KP	5-144	Proportional Gain coefficient, used for PID filter
?KP	5-144	Report PID proportional gain
KU	5-147	PID integration sum upper limit
?KU	5-147	Report PID integration sum upper limit
KV	5-148	Velocity Feedforward coefficient, used in tuning servo systems
?KV	5-148	Report velocity feedforward
?PM	5-179	Report hold state
STEP ENCODER CONTROL COMMANDS – SECTION 5.7.1		
ER	5-97	Encoder ratio, set encoder count to motor count ratio
?ER	5-98	Report motor:encoder ratio
HD	5-110	Hold deadband, specify deadband tolerance for position hold
?HD	5-110	Report position maintenance deadband
HG	5-113	Hold gain, specify position hold gain parameter
?HG	5-113	Report position maintenance gain
HV	5-124	Hold velocity, specify maximum position hold correction velocity
?HV	5-125	Report position maintenance velocity
STEP ENCODER SLIP COMMANDS – SECTION 5.7.2		
ES	5-99	Encoder slip tolerance, set tolerance before slip or stall is flagged
?ES	5-100	Report encoder slip tolerance
IS	5-132	Interrupt slip, interrupts host on slip or stall detection
RL	5-194	Return slip status of each axis
TF	5-219	Turn encoder slip kill off
TN	5-223	Turn encoder slip kill on
ENCODER SLAVE COMMANDS – SECTION 5.7.3		
ET	5-101	Encoder tracking, set encoder tracking mode
TX	5-224	Track the X axis

HOMING COMMANDS – SECTION 5.7.4		
HM	5-116	Home, find home and initialize the position counter
?HM	5-118	Report home state selection
HR	5-121	Home reverse, find home in reverse direction and initialize position counter
?HS	5-123	Report home switch true state selection
KM	5-142	Home and kill pulse generation
KR	5-145	Home in reverse and kill pulse generation
POSITION COUNTERS – SECTION 5.7.5		
LO	5-156	Set axis motor position but not the encoder position
LP	5-157	Load position, load position counter with parameter
PE	5-176	Report encoder positions of all encoder and servo axes
PP	5-181	Report motor positions of all axes
RE	5-193	Request encoder position, return current encoder position
RM	5-195	Return remainder of position divided by parameter in position counter
RP	5-196	Request position, returns current position
RU	5-200	Return current position in user units
VELOCITY COMMANDS – SECTION 5.8.1		
RV	5-201	Return current velocity at which the axis is moving
VB	5-230	Base velocity, set base velocity
?VB	5-231	Report axis base velocity
VL	5-232	Set maximum velocity to be used in profile
?VL	5-233	Report axis velocity limit
ACCELERATION COMMANDS – SECTION 5.8.2		
AC	5-41	Acceleration, set acceleration/deceleration register
?AC	5-42	Report AC command
DC	5-89	Set the deceleration rate that will be used by the GU command
RC	5-189	Return current acceleration or deceleration of the current axis
PROFILE COMMANDS – SECTION 5.8.3		
AJ	5-49	Custom S-curve Definition
?AJ	5-52	Report Custom S-curve Parameters
CN	5-78	Cosine on, enable cosine velocity profiles
LA	5-149	Linear ramp selection per axis
PF	5-176	Parabolic off, disable parabolic ramps; i.e. linear ramps will be generated
PN	5-180	Parabolic on, enable parabolic ramps
PR	5-182	Parabolic ramp selection per axis
?RT	5-199	Report ramp type
SC	5-202	Cosine ramp selection per axis
SR	5-212	Select a custom acceleration ramp for use by the currently active axis
SS	5-212	Selects a predefined S-curve accelleration

CUSTOM PROFILE COMMANDS – SECTION 5.8.4		
?DA	5-82	Print a custom ramp
DAB	5-83	Define custom ramp breakpoint
DAE	5-84	End custom ramp definition
DAR	5-85	Begin custom ramp definition
?DE	5-90	Report an acceleration ramp definition table entry
?DS	5-93	Report the size of a custom acceleration ramp table
JOGGING COMMANDS – SECTION 5.9.1		
JF	5-133	Jog at fractional rates
JG	5-134	Jog command, run motor at specified velocity until a new velocity command is sent or it is stopped by a stop or kill command
MO	5-167	Move one pulse in current direction
TM	5-222	Jog at the current velocity for the specified number of milliseconds
MOVE SPECIFICATION COMMANDS – SECTION 5.9.2		
MA	5-161	Move absolute, move to absolute position
ML	5-164	Move linear, move specified distance relative from current position
MR	5-168	Move relative, move specified distance from current position
MT	5-170	Move to, move to specified position
MOVE EXECUTION COMMANDS – SECTION 5.9.3		
GD	5-104	Go and reset done flags
GN	5-106	Go and notify when done
GO	5-107	Go command, start execution of motion
GS	5-108	Go and use the home switch to monitor for motor slip
GU	5-109	Go and use the AC values to accelerate and the DC values to decelerate
MOVE TERMINATION COMMANDS – SECTION 5.9.4		
KL	5-141	Kill, flush queue and terminate pulse generation immediately on all axes without decelerating
KS	5-146	Kill selected axes
SA	5-201	Flush queue and stops all axes with deceleration
SD	5-203	Stop all axes and clear any done flags
SI	5-206	Stop selected motors
SO	5-210	Stop at a designated position using a specified ramp down distance
ST	5-213	Stop, flush and decelerate queue to stop
MOVE COMPLETION NOTIFICATION COMMANDS – SECTION 5.9.5		
CA	5-71	Clear done flag of currently addressed axis
IC	5-126	Interrupt clear, clear done interrupt status and error flags
ID	5-127	Interrupt host when done and set done flag
II	5-128	Interrupt independent
IN	5-129	Interrupt when nearly done
IP	5-131	Interrupt when in position

VELOCITY STAIRCASING COMMANDS – SECTION 5.9.6		
FP	5-103	Force position, flush queue and attempt to stop at specified position
MM	5-166	Move minus, set minus direction for MV type move
MP	5-167	Move plus, set positive direction for MV type move
MV	5-171	Move velocity, move to first parameter (absolute position) at second parameter velocity without stopping at end of move
SP	5-211	Stop at position, stop at specified position if possible after all commands have been executed
VELOCITY STREAMING COMMANDS – SECTION 5.9.7		
VS	5-234	Velocity stream, slave velocity mode for profiling
CONTOURING COMMANDS – SECTION 5.9.8		
AF	5-47	Auxiliary off
AN	5-54	Auxiliary on
CE	5-74	End contour definition, ramp to a stop
CD	5-72	Define a contour
CG	5-75	Execute a constant velocity contour and prevent MT parsing
CK	5-77	End contour definition, immediately stop step pulses
CR	5-79	Circular interpolation, move in a circle
CV	5-80	Contouring velocity, definition
CX	5-81	Contour execute
FX	5-103	Enable axis gantry mode
MT	5-170	Move to, move to specified position
NV	5-173	Set a new contour velocity from with in a contour
WAITING COMMANDS – SECTION 5.10.1		
BW	5-70	Wait for input to go low
SW	5-217	Sync wait, wait for the input bit to be released (non-active)
WA	5-235	Wait until all moves on all axes are finished
WQ	5-239	Wait until current axis queue is empty
WT	5-241	Wait, wait for specified number of milliseconds
LOOPING COMMANDS – SECTION 5.11.1		
CW	5-80	Clear while
LE	5-150	Loop end, terminate most recent LS command
LS	5-158	Loop start, set loop counter, from 1 to 2,147,483,647 loops; (may be nested to 4 levels)
WD	5-236	While end, WS loop terminator
WG	5-236	Terminate WH loop
WH	5-237	While, execute all commands until WG loop terminator, until flag cleared by CW command
WS	5-240	While sync, execute while sync of input is true

SPECIAL COMMANDS – SECTION 5.12		
AJ	5-49	Custom S-curve Definition
?AJ	5-52	Report Custom S-curve Parameters
BC	5-63	Set Backlash Compensation
?BC	5-63	Report Backlash Compensation
#UR	5-227	Set update rate

5.4 SYSTEM STATUS AND CONTROL COMMANDS

5.4.1 IDENTIFICATION COMMANDS

These commands allow the host to request the status of various move parameters, including the status of limit and home switches and/or allow control of various system parameters and operating modes to allow the user to optimize the response of the system for the application.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
?AB	5-40	Report Auxiliary bit state
?AC	5-42	Report AC command
?AD	5-42	Report Auxiliary bit state
?AJ	5-52	Report Custom S-curve Parameters
?AQ	5-58	Query current axis
?BC	5-63	Report Backlash Compensation
?BR	5-68	Report the state of the general output bits at power-up and reset
?DA	5-82	Print a custom ramp
?DB	5-86	Report direction bit logic
?DC	5-90	Report Deceleration Rate
?DE	5-90	Report an acceleration ramp definition table entry
?EH	5-96	Query Encoder Home
?ER	5-98	Report motor:encoder ratio
?ES	5-100	Report encoder slip tolerance
?HD	5-110	Report position maintenance deadband
?HG	5-113	Report position maintenance gain
?HM	5-118	Report home state selection
?HS	5-123	Report home switch true state selection
?HV	5-125	Report position maintenance velocity
?KA	5-136	Report acceleration feed-forward
?KB	5-137	Report axis PID upper Bound limit
?KD	5-138	Report PID derivative gain
?KF	5-139	Report servo axis friction offset
?KI	5-140	Report PID integral gain

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
?KO	5-143	Report PID closed- loop offset
?KP	5-144	Report PID proportional gain
?KU	5-147	Report PID integration sum upper limit
?KV	5-148	Report velocity feedforward
?LM	5-154	Report axis overtravel enable/disable
?LS	5-160	Report limit active state
?PA	5-175	Report power automatic state
?PM	5-179	Report hold state
?PS	5-183	Report Motor Type
?RT	5-199	Report ramp type
?SE	5-204	Report settling time
?SK	5-207	Report Axis Slip Kill mode selection
?SL	5-209	Report soft limit status
?SO	5-211	Report analog output mode
?SV	5-214	Report servo voltage invert selection
?TL	5-221	Report software overtravel ranges
?UR	5-226	Report the controller's motor update rate
?UU	5-229	Report axis user units' axis assignment
?VB	5-231	Report axis base velocity
?VL	5-233	Report axis velocity limit
WY	5-241	Who are you , returns model and software revision

5.4.2 POWER-UP DEFAULTS

The MAXv can store most user-settable parameters and reload them when the board powers-up or is reset. The following commands can be used to store these parameters, return the board to factory default, reload the stored parameters, and reset the board to reload the stored parameters. The following list of parameters can have their values saved to flash memory: AC, BI/UN, BR, EH, ER, ES, HD, HG, HH/HL, HV, KA, and KB. KD, KF, KI, KO, KP, KU, KV, LA/SC/PR/SS/AJ, LH/LL, LN/LF, PA0/PA1, SE, SF/SL, VB, VL, and UU,

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
APB	5-56	Archive current parameters in back up of flash
APP	5-57	Make the current parameter set the power up default values
?AQ	5-58	Query current axis
RDB	5-190	Restore default parameter in back up space
RDP	5-192	Restore power up defaults

5.4.3 QUEUE SELECTION COMMANDS

The following commands set the context to direct the commands which follow to the appropriate axis. They remain in effect until superseded by another command of the same type, specifying a different axis.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
AA	5-39	Any following commands are for the AA (All Axes) mode
AM	5-53	Axes multitasking mode
AR	5-58	Any following commands are for the R axis
AS	5-59	Any following commands are for the S axis
AT	5-59	Any following commands are for the T axis
AU	5-60	Any following commands are for the U axis
AV	5-60	Any following commands are for the V axis
AX	5-61	Any following commands are for the X axis (default on reset)
AY	5-61	Any following commands are for the Y axis
AZ	5-62	Any following commands are for the Z axis

5.4.4 QUEUE STATUS COMMANDS

Commands sent to the MAXv are either queued or immediate. As the type names imply, queued commands are stored in first-in-first-out buffers to be executed in the order they were sent while immediate commands are executed the moment they are received. There are several internal queues in the MAXv corresponding to the various axis and command modes and each of these queues has a limited amount of storage space. For example, the X axis command queue can hold 2559 “units” and its argument queue can hold 12799 “units”. Each command requires some number of storage units. The amount of storage required is listed in a table with each command. The following commands provide control and monitoring capability for the queues.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
FL	5-102	Flush an axis queue
RQ	5-197	Report Contour Queue Size
RQA	5-197	Report Axis argument queue
RQC	5-198	Report Axis command queue

5.4.5 USER UNIT COMMANDS

The following commands allow specification of move parameters in user defined units. In the UU mode, the controller will automatically convert all move parameters to these units once they have been initialized

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
UF	5-225	User units off, turn off user unit translation
UU	5-228	User units, multiply acceleration, velocity and distance parameters by specified parameter
?UU	5-229	Report axis user units' axis assignment

5.4.6 AXIS STATUS COMMANDS

The MAXv monitors the various inputs and conditions that can affect motor movement and system status. This information is frequently needed by host applications so that proper motion decisions can be made and appropriate actions taken. The following commands provide this status feedback to the host.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
EA	5-94	Encoder status, return encoder status of currently addressed axis
QA	5-186	Query status of switches and flags for addressed axis without affecting flags
QI	5-186	Query status of switches and flags on all axes without affecting flags
RA	5-188	Return status of switches and flags and reset flags
RI	5-193	Return status of switches and flags for all axes and reset flags

5.4.7 MACROS

In applications that must perform frequent, repetitive tasks, macros can be used to minimize communication bandwidth consumption and speed up initial task execution. Macros are storage areas in the MAXv of which 5 are "temporary"; i.e. not saved at power-off, and 20 are "permanent"; i.e. stored in non-volatile flash RAM.

Macros can be edited, stored, read back to the host, and executed using the following commands.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
MD	5-163	Define a temporary macro
MX	5-173	Execute a macro command string
PM	5-178	Print a macro command string
PT	5-185	Preserve a temporary macro

5.5 I/O CONTROL COMMANDS

5.5.1 AUXILIARY CONTROL COMMANDS

Each axis of the MAXv has an associated auxiliary output line. Though this line can be used as a general purpose output, it also has a special purpose: Power-Automatic Mode. In power-automatic mode, the auxiliary line will invert at the beginning of every motion and return to normal at the end. The "normal" state of this line is user-controllable as is the amount of time to delay, allowing the motor to settle, before returning the line to normal at the end of a move. The following commands provide this control as well as feedback regarding the state and function of each auxiliary line.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
?AD	5-42	Report Auxiliary bit state
ADH	5-43	Set Auxiliary default to high
ADL	5-44	Set Auxiliary default to low
AN	5-54	Auxiliary on
PA	5-174	Power automatic, turn power on before each move and off after the move
?PA	5-175	Report power automatic state
PH	5-177	Power high
PL	5-177	Power low
SE	5-204	Settling time
?SE	5-204	Report settling time

5.5.2 GENERAL PURPOSE I/O CONTROL

The MAXv has 16 configurable general purpose I/O bits. From the factory they are configured as 8 inputs and 8 outputs. The following commands can be used to set outputs high or low individually or as a group and to read inputs as a group.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
BD	5-64	Set the direction of the general purpose I/O bits
BH	5-65	Set selected I/O bit high (off)
BL	5-67	Set selected I/O bit low (on)
?BR	5-68	Report the state of the general output bits at power-up and reset
BR	5-68	Define the state of the general output bits at power-up and reset
BS	5-69	Set all bits of the general purpose output port to the state specified by the hex argument.
BX	5-70	Return bit status in hex format
IO	5-130	I/O Bit Direction Selection
RB	5-189	Report bit direction

5.5.3 HOME CONTROL COMMANDS

System homing is an essential step in most systems. To accommodate a wide range of homing methods, the following commands provide the ability to set home inputs active high or low and to enable or disable encoder index signals as part of home detection.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
EH	5-95	Defining Encoder Home
?EH	5-96	Query Encoder Home
HE	5-111	Encoder home mode, set home on encoder logic
HH	5-114	Home high, home switches are active high
HL	5-115	Home low, home switches are active low
HS	5-123	Home switch, enable home switch mode

5.5.4 LIMIT CONTROL COMMANDS

Limit conditions are treated as critical errors in the MAXv. When a limit is encountered, the axis involved will cease motion and flush any pending motion commands for that axis. However, since needs vary from application to application, the following commands will allow limit behavior customization to fit almost any system.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
LF	5-151	Disable limit switches for selected axis
LH	5-152	Limit high, limit switch is active high
LL	5-153	Limit low, limit switch is active low
?LM	5-154	Report axis overtravel enable/disable
LN	5-155	Enable limit switches for selected axis
?LS	5-160	Report limit active state
QL	5-187	Query all limit sensors
SF	5-205	Soft limit off, restore normal overtravel operation
SL	5-208	Soft limit mode, allow pulse train to ramp down on overtravel
?SL	5-209	Report soft limit status
TL	5-220	Set software travel limits
?TL	5-221	Report software overtravel ranges

5.5.5 ANALOG I/O COMMANDS

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
AI	5-48	Analog Input
AO	5-55	Set analog input zero offset
DOV	5-91	Output Auxiliary DAC Voltage
DOZ	5-92	Set Auxiliary DAC Zero Offset

5.6 SERVO CONTROL COMMANDS

The following commands are valid only for servo axes and should never be executed while the specific axis is in motion.

5.6.1 SERVO VOLTAGE CONTROL COMMANDS

Different servo amplifiers have different requirements for their control inputs. Some simply behave differently despite similar input requirements. To enable the use of a wide range of amplifiers, the MAXv will accept the following commands for use in configuring servo outputs.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
BI	5-66	Bipolar, set the analog torque outputs to bipolar
KO	5-143	Offset coefficient
?KO	5-143	Report PID closed- loop offset
?SO	5-211	Report analog output mode
?SV	5-214	Report servo voltage invert selection
SVI	5-215	Invert servo voltage
SVN	5-216	Normalize servo voltage
UN	5-226	Unipolar, set the analog torque outputs to unipolar

5.6.2 PID COMMANDS

The MAXv uses a PID filter for servo position maintenance. The following commands provide user-control over the filter parameters and feedback of the same. See section 2.5 Connect and Checkout the Servo System for more information regarding the use of these commands for tuning your servo motors.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
HF	5-112	Disable PID
HN	5-119	Enable PID
KA	5-136	Acceleration Feedforward coefficient, used in tuning servo systems
?KA	5-136	Report acceleration feed-forward
KB	5-137	PID Upper Bound Limit Coefficient
?KB	5-137	Report axis PID upper Bound limit
KD	5-138	Derivative Gain coefficient, used for PID filter
?KD	5-138	Report PID derivative gain
KF	5-139	Set Servo axis PID friction coefficient
?KF	5-139	Report servo axis friction offset
KI	5-140	Integral Gain coefficient, used for PID filter
?KI	5-140	Report PID integral gain
KP	5-144	Proportional Gain coefficient, used for PID filter
?KP	5-144	Report PID proportional gain
KU	5-147	PID integration sum upper limit
?KU	5-147	Report PID integration sum upper limit
KV	5-148	Velocity Feedforward coefficient, used in tuning servo systems
?KV	5-148	Report velocity feedforward
?PM	5-179	Report hold state

5.7 STEP ENCODER CONTROL COMMANDS

5.7.1 STEP ENCODER CONTROL COMMANDS

Stepper systems, like servo systems, use encoder for position feedback. However, stepper systems do not use PID filters due to operating constraints in the stepper motors themselves. Instead, the MAXv uses the following commands to perform position maintenance for stepper axes.

It is important to note that stepper motor position cannot be maintained over the course of a move but rather at the end of the move. Once the axis has initially stopped, the axis will begin moving again to correct for any error encountered during the course of the full move. This process will continue until the encoder position is within the dead band of the motor's target position.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
ER	5-97	Encoder ratio, set encoder count to motor count ratio
?ER	5-98	Report motor:encoder ratio
HD	5-110	Hold deadband, specify deadband tolerance for position hold
?HD	5-110	Report position maintenance deadband
HG	5-113	Hold gain, specify position hold gain parameter
?HG	5-113	Report position maintenance gain
HV	5-124	Hold velocity, specify maximum position hold correction velocity
?HV	5-125	Report position maintenance velocity

5.7.2 STEP ENCODER SLIP COMMANDS

In applications that require notification when a stepper motor slips beyond a given tolerance, the following commands will be of assistance. These commands do nothing to maintain position. Instead, they tell the MAXv to react to a slip condition by notifying the host or ceasing motion.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
ES	5-99	Encoder slip tolerance, set tolerance before slip or stall is flagged
?ES	5-100	Report encoder slip tolerance
IS	5-132	Interrupt slip, interrupts host on slip or stall detection
RL	5-194	Return slip status of each axis
TF	5-219	Turn encoder slip kill off
TN	5-223	Turn encoder slip kill on

5.7.3 ENCODER SLAVE MODES

Encoder tracking modes connect a motor to an axis at a given ratio. For each turn of the encoder, the motor will move proportionately.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
ET	5-101	Encoder tracking, set encoder tracking mode
TX	5-224	Track the X axis

5.7.4 HOMING COMMANDS

Section 5.5.3 Home Control Commands details the commands available for customizing homing operations. The commands below initiate the physical homing process.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
HM	5-116	Home, find home and initialize the position counter
?HM	5-118	Report home state selection
HR	5-121	Home reverse, find home in reverse direction and initialize position counter
?HS	5-123	Report home switch true state selection
KM	5-142	Home and kill pulse generation
KR	5-145	Home in reverse and kill pulse generation

5.7.5 POSITION COUNTERS

Applications frequently need to know the actual positions of motors and encoders as opposed to the assumed positions the applications keep track of. The following commands are available for retrieving that information as well as forcibly setting those positions. This can be useful for setting “floating zero” positions.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
LO	5-156	Set axis motor position but not the encoder position
LP	5-157	Load position, load position counter with parameter
PE	5-176	Report encoder positions of all encoder and servo axes
PP	5-181	Report motor positions of all axes
RE	5-193	Request encoder position, return current encoder position
RM	5-195	Return remainder of position divided by parameter in position counter
RP	5-196	Request position, returns current position
RU	5-200	Return current position in user units

5.8 PROFILE CONTROL COMMANDS

5.8.1 VELOCITY COMMANDS

Part of configuring any system involves defining velocity limits. The commands below provide control over setting these limits and reporting them back to the host.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
RV	5-201	Return current velocity at which the axis is moving
VB	5-230	Base velocity, set base velocity
?VB	5-231	Report axis base velocity
VL	5-232	Set maximum velocity to be used in profile
?VL	5-233	Report axis velocity limit

5.8.2 ACCELERATION COMMANDS

Along with velocity limits, acceleration limits are also critical to most systems. The following commands allow customization of these parameters.

COMMANDS IN CHAPTER 5		
COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
AC	5-41	Acceleration, set acceleration/deceleration register
?AC	5-42	Report AC command
DC	5-89	Set the deceleration rate that will be used by the GU command
RC	5-189	Return current acceleration or deceleration of the current axis

5.8.3 PROFILE COMMANDS

Often, the default linear acceleration profile is not optimum for a given system. To meet the needs of those systems, the MAXv has a number of commands that allow partial or even complete customization of the profile. The commands below allow the use of parabolic, cosine, and even custom ramps. See Section 5.8.4 Custom Profile Commands for commands to define custom ramps.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
AJ	5-49	Custom S-curve Definition
?AJ	5-52	Report Custom S-curve Parameters
CN	5-78	Cosine on, enable cosine velocity profiles
LA	5-149	Linear ramp selection per axis
PF	5-176	Parabolic off, disable parabolic ramps; i.e. linear ramps will be generated
PN	5-180	Parabolic on, enable parabolic ramps
PR	5-182	Parabolic ramp selection per axis
?RT	5-199	Report ramp type
SC	5-202	Cosine ramp selection per axis
SR	5-212	Select a custom acceleration ramp for use by the currently active axis
SS	5-212	Selects a predefined S-curve acceleration

5.8.4 CUSTOM PROFILE COMMANDS

When linear, parabolic, and cosine acceleration ramps are insufficient, custom ramps can be defined to meet virtually any profiling need. The following commands provide the capability to define a custom ramp. For even more control over the custom ramp's definition, it is recommended to use the S-curve commands. (See AJ, SS, ?AJ)

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
?DA	5-82	Print a custom ramp
DAB	5-83	Define custom ramp breakpoint
DAE	5-84	End custom ramp definition
DAR	5-85	Begin custom ramp definition
?DE	5-90	Report an acceleration ramp definition table entry
?DS	5-93	Report the size of a custom acceleration ramp table

5.9 MOTION GENERATION COMMANDS

5.9.1 JOGGING COMMANDS

When an application requires a motor to move without stopping or, perhaps, to move until told to stop, the jogging commands that follow will be useful. These commands will start motion on an axis, ramping up to the specified jog velocity, and continue indefinitely, stopping only when told to stop, a limit is reached, or a timeout occurs.

The JG command is very useful when first setting up and testing a system because it generates a continuous stream of step pulse that can easily be tracked.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
JF	5-133	Jog at fractional rates
JG	5-134	Jog command, run motor at specified velocity until a new velocity command is sent or it is stopped by a stop or kill command
MO	5-167	Move one pulse in current direction
TM	5-222	Jog at the current velocity for the specified number of milliseconds

5.9.2 MOVE SPECIFICATION COMMANDS

The following commands define motions on one or more axes that terminate at specified positions. Full profiles are generated that guarantee position achievement either on a per axis basis or in a coordinated fashion.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
MA	5-161	Move absolute, move to absolute position
ML	5-164	Move linear, move specified distance relative from current position
MR	5-168	Move relative, move specified distance from current position
MT	5-170	Move to, move to specified position

5.9.3 MOVE EXECUTION COMMANDS

The following commands initiate moves defined by commands in section 5.9.2 Move Specification Commands. A number of different commands are available, tailored to various application needs.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
GD	5-104	Go and reset done flags
GN	5-106	Go and notify when done
GO	5-107	Go command, start execution of motion
GS	5-108	Go and use the home switch to monitor for motor slip
GU	5-109	Go and use the AC values to accelerate and the DC values to decelerate

5.9.4 MOVE TERMINATION COMMANDS

The following commands allow termination of move sequences in process. When things go wrong or a motion simply needs to be commanded to stop prematurely, the commands below will be useful. These commands can be used to stop motors gracefully or abruptly, depending on the needs of the application.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
KL	5-141	Kill, flush queue and terminate pulse generation immediately on all axes without decelerating
KS	5-146	Kill selected axes
SA	5-201	Flush queue and stops all axes with deceleration
SD	5-203	Stop all axes and clear any done flags
SI	5-206	Stop selected motors
SO	5-210	Stop at a designated position using a specified ramp down distance
ST	5-213	Stop, flush and decelerate queue to stop

5.9.5 MOVE COMPLETION NOTIFICATION COMMANDS

These commands allow the synchronization of moves with external events or multiple axis sequences. If an application needs to know when a move or series of commands has completed and fully processed, the following commands can be used to generate notifications. Commands are available to generate a simple notification or perform a somewhat more complex analysis to decide when and how to notify the host.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
CA	5-71	Clear done flag of currently addressed axis
IC	5-126	Interrupt clear, clear done interrupt status and error flags
ID	5-127	Interrupt host when done and set done flag
II	5-128	Interrupt independent
IN	5-129	Interrupt when nearly done
IP	5-131	Interrupt when in position

5.9.6 VELOCITY STAIR CASING COMMANDS

The following commands describe the velocity staircase mode. This mode is useful in applications requiring a change in velocity at a prescribed position without stopping. Similar to the jogging commands, velocity stair casing will move an axis at a specified velocity. The difference is that the next stair casing command in the queue will not be processed until a specified position is reached. Stair casing also allows the host to specify a position for the motor to stop, unlike the jogging commands.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
FP	5-103	Force position, flush queue and attempt to stop at specified position
MM	5-166	Move minus, set minus direction for MV type move
MP	5-167	Move plus, set positive direction for MV type move
MV	5-171	Move velocity, move to first parameter (absolute position) at second parameter velocity without stopping at end of move
SP	5-211	Stop at position, stop at specified position if possible after all commands have been executed

5.9.7 VELOCITY STREAMING COMMANDS

Velocity streaming is a specialized form of velocity stair casing. Streaming simply produces specified velocities without ramping or other processing. In effect, streaming allows the host to put velocities directly into the MAXv's internal velocity registers for the X and Y axes.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
VS	5-234	Velocity stream, slave velocity mode for profiling

5.9.8 CONTOURING COMMANDS

The MAXv will attempt to generate any profile which it is asked to do. It is the responsibility of the host to be sure the acceleration required when generating a circle or any other change in direction is possible within the mechanical constraints of the system. All corners must be defined by arcs and tangents to those arcs, else the change in direction will be instantaneous and generate very large accelerations. The arc radius must be chosen so that the acceleration constraints of the system are met.

Constant velocity contouring is similar to a series of discrete move commands except that it allows multiple discrete moves to be executed in series, without stopping, maintaining a constant vector velocity among the involved axes. The commands below are those that are available in contouring mode.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
AF	5-47	Auxiliary off
AN	5-54	Auxiliary on
CE	5-74	End contour definition, ramp to a stop
CD	5-72	Define a contour
CG	5-75	Execute a constant velocity contour and prevent MT parsing
CK	5-77	End contour definition, immediately stop step pulses
CR	5-79	Circular interpolation, move in a circle
CV	5-80	Contouring velocity, definition
CX	5-81	Contour execute
MT	5-170	Move to, move to specified position
NV	5-173	Set a new contour velocity from with in a contour

5.10 SYNCHRONIZATION COMMANDS

5.10.1 WAITING COMMANDS

The commands below provide several methods of command and move synchronization. By forcing the MAXv to wait a specified amount of time or wait until a set of axes has stopped moving before processing the next command in the queue, the host gains fine-grained control over the motion process.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
BW	5-70	Wait for input to go low
SW	5-217	Sync wait, wait for the input bit to be released (non-active)
WA	5-235	Wait until all moves on all axes are finished
WQ	5-239	Wait until current axis queue is empty
WT	5-241	Wait, wait for specified number of milliseconds

5.11 LOOPING COMMANDS

5.11.1 LOOPING COMMANDS

Often, applications have need of the ability to repeat a sequence of commands until some event occurs. The commands in this section will allow looping over a given series of commands until a timeout or an I/O event such as a rising or falling edge occurs or simply to loop a specific number of times. It should be noted that only queued commands can be looped; immediate commands will be executed immediately and will not stay in the queue to be part of a loop.

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
CW	5-80	Clear while
LE	5-150	Loop end, terminate most recent LS command
LS	5-158	Loop start, set loop counter, from 1 to 2,147,483,647 loops; (may be nested to 4 levels)
WD	5-236	While end, WS loop terminator
WG	5-236	Terminate WH loop
WH	5-237	While, execute all commands until WG loop terminator, until flag cleared by CW command
WS	5-240	While sync, execute while sync of input is true

5.12 SPECIAL COMMANDS

5.12.1 S-CURVE ACCELERATION

An S-curve acceleration profile is an alternative to the traditional trapezoidal profile. A trapezoidal profile has a constant rate of acceleration, or jerk, on both the acceleration and deceleration sides of the profile. S-curve provides a more controlled and efficient mode of operation for those situations that require optimal speed and smooth starting and stopping of motion. The S-curve acceleration is a means for softening the jerk, controlling shifting materials such as liquids, and to prevent overshoot of high inertial loads.

Functional Description: An S-curve acceleration profile is one that starts with increasing jerk, then transitions to constant jerk, then transitions to decreasing jerk until zero acceleration is reached at the desired velocity. When it is time to start decelerating, the S-curve profile starts increasing negative jerk and then transitions to constant negative jerk, and finally transitions to decreasing negative jerk until zero velocity is reached.

The S-curve profile is not truncated in moves that do not reach full velocity. This happens when the total distance of the move is less than the distance required to accelerate to full velocity plus the distance to decelerate from the full velocity to a stop. In this case, the controller calculates the entire profile and selects the velocity that will preserve selected S-curve profile, by limiting the velocity on short moves to a velocity that is sufficiently small to allow the entire profile to be preserved. Typical S-curve Acceleration Profile (Symmetrical)

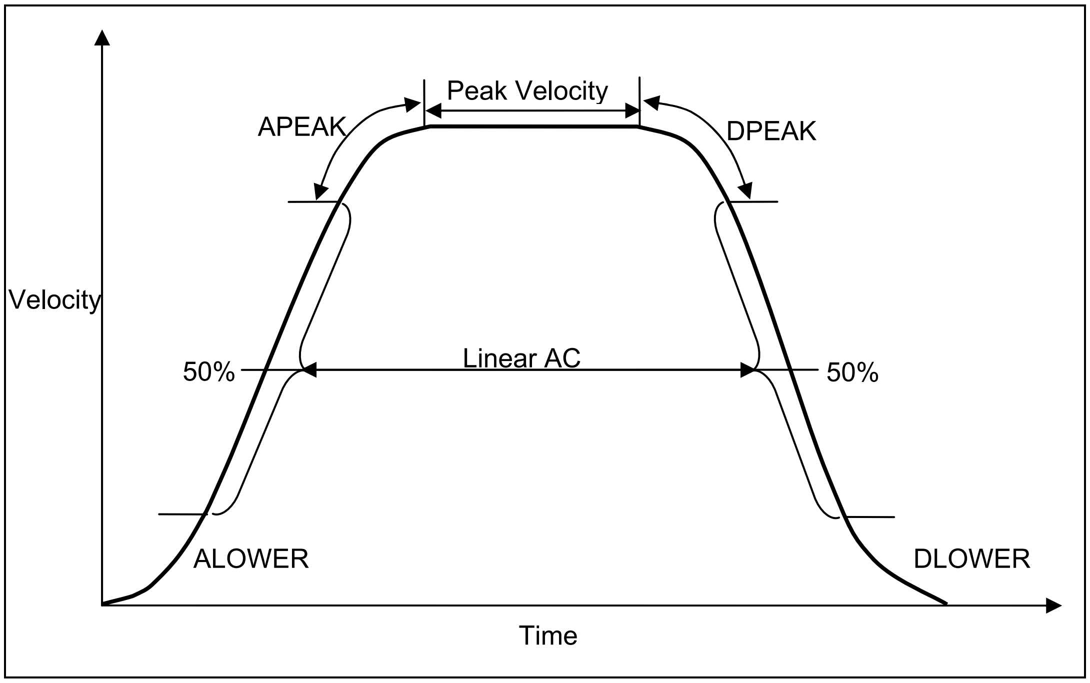


Figure 5-1 Symmetrical S-curve

5.12.1.1 BACKLASH COMPENSATION

Some motion systems require adjustments for backlash inherent in gearing systems when the direction of the drive motion is changed. This type of adjustment is particularly necessary when the gear train wears due to use and it may be necessary to adjust the number of steps compensated as the gear train ages.

The MAXv family of controllers includes the backlash compensation capability and is implemented for both stepper and servo axes. Note that the implementation is slightly different between the stepper and servos to accommodate the inherent differences.

The BC <step count> command is used to provide backlash compensation for an axis, and it can be saved as a parameter with the APP command. The command BC0**;** (zero) negates the backlash compensations for the selected axis.

5.12.1.2 STEPPER MOTOR AXIS

If the referenced axis controls a stepper motor, additional motor steps (specified by the number of steps in <step count> are generated), whenever a move command causes the motor to reverse direction to compensate for backlash. The backlash compensation count is output to the motor during the first motor update cycle following a direction change. This permits the compensation to be used with constant velocity contours. Note: it is assumed that the stepper motor is able to develop sufficient torque to accommodate this burst of motor steps. The backlash compensation for stepper motors is limited to a range of from 0 to 50 steps.

Example: For a stepper system, provide for 10 steps Backlash Compensation, i.e., cause the controller to output and additional 10 motor steps whenever a move causes the motor direction to be reversed.

Enter: AX;
BC10;

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
BC#;	AX – AT	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

This command line will set the backlash compensation for the X-Axis to 10 steps adding 10 steps on a change of direction for the selected axis.

5.12.1.3 SERVO MOTOR AXIS

If the referenced axis is a servo motor then offsetting the motor's position "set point" from the commanded position will compensate for backlash.

If the motor's "home position" has been set by moving in the positive direction, then the servo set point will automatically include the compensation for backlash in the positive direction. In this case the backlash compensation should be entered as a negative number of steps.

The backlash compensation for a servo motor occurs in the direction of the sign of the number in the BC command, and is limited to a range of from minus 50 to plus 50 encoder counts.

Example: Provide a 10 step backlash compensation for a servo motor on the X-Axis that has been homed in the positive direction.

Enter:

```
(1)
AX;
BC-10;
MA1000;
GO;
(2)
AX;
MA-1000;
GO;
```

This would cause the controller to go to a position set point of 1000. Note the RP command would report the motor's commanded position of 1000. The RE command would report the motor's actual position of 1000.

However, in the second command you would be changing directions so that the motor will go 2010 steps, ending at -1000, and the encoder will read –1010.

Note: The recommended method of using the backlash compensation is to assign backlash compensation for each axis and archive the command, so that the compensation will occur automatically, for each appropriate axis, For example, AX;BC-10;AY;BC5;AZ;BC7; APP;

Related Commands: APP, ?BC

COMMAND	SECTION PAGE NUMBER	COMMAND DESCRIPTION
AJ	5-49	Custom S-curve Definition
?AJ	5-52	Report Custom S-curve Parameters
BC	5-63	Set Backlash Compensation
?BC	5-63	Report Backlash Compensation
#UR	5-227	Set update rate

5.13 COMMAND DESCRIPTIONS

AA ALL AXES

The AA command performs a context switch to multi-axis mode. All commands entered after this one will be treated as "all axes" commands which must be formatted for multi-axis use rather than single-axis use. Each command will be executed in the order in which it is received. This is true even if the second command affects axes other than those affected by the first command. For example, if AA mode is entered followed by a move of the X axis and then a move of the Y axis, the Y axis move will not begin until the X axis move has completed.

Example: Perform an absolute move using the X and Y axes. When this move is complete, perform a relative move using the Y, Z, and T axes.

Enter:

```
AA ;
MA12000,14000;
GO;
MR,5000,1500,100000;
GO;
```

Response: None

NOTE: This command changes the axis mode immediately, but has axis queue requirements because it places synchronization entries in all axis queues.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AA	AX – AS	1	1
AA	AA-AM	1	1
-	AA/CD	Not Valid	

Related commands: AM, ?AQ, AR, AS, AT, AU, AV, AX, AY, AZ, CD

?AB REPORT AUXILIARY BIT STATE

This command returns the bit state of the auxiliary bit of the current axis, (AN or AF).

Example: Determine if the X axis auxiliary bit is set on.

Enter: AX;
?AB

Response: =on<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?AB	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: AF, AN, PA, ?SE, SE

AC ACCELERATION

The AC command sets the acceleration/deceleration register to the operand which follows the command. The parameter must be greater than zero (zero is not valid) and less than 8,000,000, and the unit is in steps per second per second. All the following move commands for the axis being programmed will accelerate and decelerate at this rate until another AC command is entered. See the APP command, page 5-57 to preserve the AC settings as the power-up/reset values. The default value is 2,000,000.

RANGE: 1 ≤ AC ≤ 8000000

Example: In the single axis mode, set the Y axis acceleration to 200,000 counts per second per second.

Enter: AY;
AC200000;

Response: None

Example: In the AA mode, set the acceleration of the X axis to 200,000 and the Z axis to 50,000 and leave the other axes with their previous values.

Enter: AA ;
AC200000,,50000;

Response: None

QUEUE REQUIREMENTS				
FORMAT	MODE	Axis Ramp Type	COMMAND	ARGUMENT
AC#; or AC#,#,#,#,#,#,#,#;	AX-AS or AA-AM	Linear (LA or PF)	3	3
AC#; or AC#,#,#,#,#,#,#,#;	AX-AS or AA-AM	Short Parabolic (PN0; or PR0:)	3	5
AC#; or AC#,#,#,#,#,#,#,#;	AX-AS or AA-AM	All other Parabolic Forms	3	12
AC#; or AC#,#,#,#,#,#,#,#;	AX-AS or AA-AM	Custom Ramps (SR)	3	No. of ramp segments +12
AC#; or AC#,#,#,#,#,#,#,#;	AX-AS or AA-AM	S-curve (AJ)	6	63

Related commands: ?AC, DC, RC, VB, VL

?AC REPORT AC COMMAND

This command will reply with the current acceleration value for the selected axis.

Example: Report the current AC value for this axis.

Enter: ?AC

Response: =200000<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?AC	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: AC

?AD REPORT DEFAULT AUXILIARY BIT STATE

This command reports the power up default selection for the axis Aux bit..

Example: Report the power up state of the Y axis auxiliary bit

Enter: AY;
?AD

Response: =on<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?AD	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ADH, ADL

ADH SET AUXIILIARY DEFAULT TO HIGH

The ADH command sets the default power up or reset state of the auxiliary line for the current axis to high. This change is stored as a power up parameter in flash automatically and need not be stored via the AP command. Since this command writes to non-volatile memory it should be used only when necessary and not in repeatedly called functions.

NOTE: This command will also archive all other parameter values as power up defaults.

Example:	Set the power up state of the Z axis auxiliary line to high

Enter:	AZ;
	ADH;

Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ADH	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?AD, ADL

ADL SET AUXILIARY DEFAULT TO LOW

The ADL command sets the default power up or reset state of the auxiliary line for the current axis to low. This change is stored in nonvolatile memory automatically and need not be stored via the APP command. Since this command writes to non-volatile memory it should be used only when necessary and not in repeatedly called functions.

NOTE: This command will also archive all other parameter values as power up defaults.

Example: Set the power up state of the Y axis auxiliary line to low

Enter: AY;
ADL;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ADL	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?AD, ADH

AEL LOAD AUXILIARY ENCODER POSITION

This command sets the encoder position of the specified auxiliary encoder channel.

AEL#, value;
# specifies the auxiliary encoder channel to load

RANGE: 0 ≥ # ≥ 1

Value specifies the encoder position value to set

RANGE: +/- position range

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AEL#;	AX – AS	Immediate	
AEL#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Example: Set the encoder position to 0 for the auxiliary encoder channel number 1.

Enter: AEL1,0;

Response: None

Related commands: AER

AER REPORT AUXILIARY ENCODER POSITION

This command reports the encoder position of the specified auxiliary encoder channel.

AER#
# specifies the auxiliary encoder channel to report

RANGE: 0 ≥ # ≥ 1

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AER#;	AX – AS	Immediate	
AER#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Example: Examine the current encoder position of auxiliary channel 0.

Enter: AER0;

Response: 12345<LF>

Related commands: AEL

AF AUXILIARY OFF

The AF command turns off the selected auxiliary outputs. That is, it causes the signal to be driven low. The AF command may be used to change power level on driver modules so equipped or as a user specified output. Note that this command will turn power automatic (PA) mode off.

Example: Turn off the Y axis auxiliary output in the single axis mode.

Enter:
```
AY;
AF;
```

Response: None

Example: Turn off the X and Z axes auxiliary outputs when in the AA command mode. The Y axis is unchanged in this example.

Enter:
```
AA;
AF1,,1;
```

Response: None

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
AF	AX – AS	1	0	N/A
AFb,b,b,b;	AA-AM	1	0	N/A
AFb,b,b,b;	AA/CD	N/A	N/A	2*

*When AF is used in a contour definition the Aux bits of all axes included in that definition will be turned off when the contour is executed.

Related commands: AN, BH, BL, BS, PA

AI REPORT AN ANALOG INPUT PORT VALUE

The AI command reports the value of one of the four analog input ports. The command takes a parameter between 0 and 5. The value is displayed in decimal.

Example:	Read the value of the first analog input port.
Enter:	AI0
Response:	AI0=1.797<LF>

NOTE: This command is not related to AN axis and works in AA/AM mode.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AI#;	AX – AS	Immediate	
AI#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AO

AJ CUSTOM S-CURVE DEFINITION

The AJ command defines the ramp up and ramp down portions of the S-curve. It can accept from 2 to 7 parameters. The command parameters are as follows:

AJ#[,ALOWER][,APEAK][,ARADIUS][,DLOWER][,DPEAK][,DRADIUS]
Parameters in brackets [] are optional and if omitted will default to the values listed below.

# specifies the S-curve profile number.

Range: 1≤ # ≤ number of axes on board

ALOWER specifies the flat portion of the lower half of the ramp up in a percentage.

Range: 0≤ ALOWER ≤ 1.0
Default value: 0.0

APEAK specifies the flat portion of the upper half of the ramp up in a percentage.

Range: 0≤ APEAK ≤1.0
Default value: alower

ARADIUS specifies the stretch factor in the curved portions of the ramp up.

Range: 1.0≤ ARADIUS ≤10.0
Default value: 1.0

DLOWER specifies the flat portion of the lower half of the ramp down in a percentage

Range: 0≤ DLOWER ≤1.0

DPEAK specifies the flat portion of the upper half of the ramp down in a percentage.

Range: 0≤DPEAK ≤1.0
Default value: DLOWER for ALOWER if DLOWER is not specified

DRADIUS specifies the stretch factor in the curved portions of the ramp down.

Range: 1.0≤ DRADIUS ≤10.0
Default value: 1.0

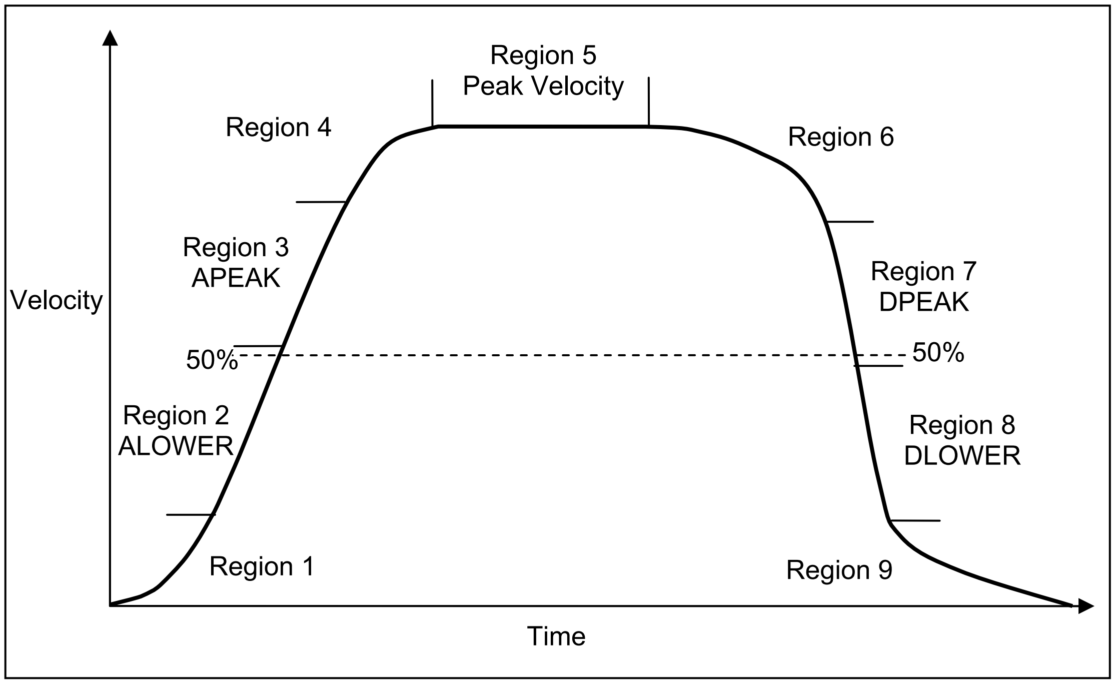


Figure 5-2

Example of a custom S-curve profile with the defined regions identified.

Region 1 = 1-ALOWER. It is the percentage of the lower half of the ramp up that S-curved.
Region 2 = ALOWER. It is the percentage of the lower half of the ramp up that is flat.
Region 3 = APEAK. It is the percentage of the upper half of the ramp up that is flat.
Region 4 = 1-APEAK. It is the percentage of the upper half of the ramp up that S-curved.
ARADIUS can "stretch" regions 1 and 4 in the time dimension.
Region 5 = Portion of profile running at maximum velocity.
Region 6 = 1-DPEAK. It is the percentage of the upper half of the ramp down that S-curved.
Region 7= DPEAK. It is the percentage of the upper half of the ramp down that is flat.
Region 8 = DLOWER. It is the percentage of the lower half of the ramp down that is flat.
Region 9 = 1-DLOWER. It is the percentage of the lower half of the ramp down that S-curved.
DRADIUS can "stretch" regions 6 and 9 in the time dimension. If the parameters are not given to define regions 6, 7, 8, and 9, they will be symmetrical with regions 4, 3, 2, and 1 respectively.

Example: Define a custom S-curve profile.

Enter:

```
AX;
AJ1,0.5,0.5,1.0;
        *Defines a profile where ramp down is symmetrical with ramp up.
AZ;
AJ2,0.1,0.1,1.0,0.8,0.8,1.0;
        *Defines an asymmetrical S-curve profile.
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AJ#,#,#,#,#,#,#,#;	AX-AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?AJ, ?RT, SS

?AJ REPORT CUSTOMER S-CURVE PARAMETERS

This command reports the parameters for a given custom S-curve profile.

Range: 1 ≤ Profile Number ≤ 8

Example: Report the parameters for custom S-curve profile number 5.

Enter:
```
AX;
?AJ5;
```

Response:
```
=5.0000000,0.50000000,0.50000000,1.00000000,0.10000000,
0.10000000, 1.00000000<LF>
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?AJ#;	AX-AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: AJ, ?RT, SS

AM AXES MULTITASKING

The AM mode allows several tasks to be managed simultaneously. This command changes the mode of all future commands to multi-axis mode. In this mode, a task may be performing coordination motion on 2 axes, while a second task is performing unrelated but simultaneous motion on another axis. All commands sent in this mode must be formatted for multi-axis mode rather than single-axis mode.

Example: Perform a coordinated relative move on the X and Y axes, while moving the T axis as a separate move at the same time.

Enter:
```
AM;
ML2000,3000;
GO;
MA,,,10000;
GO;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AM	AX – AS	Immediate	
AM	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AA, ?AQ, AT, AU, AV, AR, AS, AX, AY, AZ

AN AUXILIARY ON

The AN command turns on the selected auxiliary output ports. That is, it allows the signal to be pulled high. This is the default mode for the auxiliary line at power up or reset. The AN command may be used to change power level on driver modules so equipped, trigger another board's input or as a user specified output.

A parameter must be supplied for the desired axes when used in the AA mode so that the other axes are not affected. No parameter is required in a single axis mode. Note this command will turn power automatic (PA) mode off.

Example: Turn on the Y axis auxiliary output in the single axis mode.

Enter:
```
AY;
AN;
```

Response: None.

Example: Turn on the X and Z axes auxiliary outputs when in the AA command mode. The Y axis is unchanged in this example.

Enter:
```
AA;
AN1,,1;
```

Response: None.

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
AN	AX – AS	1	0	N/A
ANb,b,b,b;	AA-AM	1	0	N/A
ANb,b,b,b;	AA/CD	N/A	N/A	2*

*When AN is used in a contour definition the Aux bits of all axes included in that definition will be turned on when the contour is executed.

Related commands: AF, BH, BL, BS, PA

AO SET ANALOG INPUT ZERO OFFSET

This command sets the zero voltage offset of an analog input port. The command provides a calibration value so the AI command will report a zero value when zero volts are applied to the analog input.

Example:	Set the zero offset of analog input 2 to -0.05 volts.
Enter:	AO2,-0.05;
Response:	None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AO#,#;	AX – AS	Immediate	
AO#,#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related Commands: AI

APB ARCHIVE CURRENTLY USED PARAMETERS AS BACK-UP ARCHIVE

The APB command stores the current user parameters into static flash memory so they will be preserved as the back up archive. These parameters can be restored via the RDB command. Also, if an error is found in the power up default parameter set, the parameters in the back up archive will be used during power up. The following list of parameters will have their values saved to flash memory when this command is used: AC, BI/UN, BR, EH, ER, ES, HD, HG, HH/HL, HV, KA, and KB, KD, KF, KI, KO, KP, KU, KV, LA/SC/PR/SS/AJ, LH/LL, LN/LF, PA0/PA1, SE, SF/SL, VB, VL, and UU,

Note: This command should not be issued when an axis is in motion and it should be used sparingly because the flash memory has a limited number of write cycles.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGURMENT
APB	AX – AS	Immediate	
APB	AA-AM	Immediate	
-	AA/CD	Not Valid	

Example:	Store currently used parameters in back up archive of flash memory.
Enter:	APB;
Response:	None

APP ASSIGN CURRENT PARAMETERS AS POWER-UP DEFAULTS

The APP command stores the current parameter set as the power-up default set of values. This is done by writing the current parameter set to flash memory. The following list of parameters will have their values saved to flash memory when this command is used: AC, BI/UN, BR, EH, ER, ES, HD, HG, HH/HL, HV, KA, and KB, KD, KF, KI, KO, KP, KU, KV, LA/SC/PR/SS/AJ, LH/LL, LN/LF, PA0/PA1, SE, SF/SL, VB, VL, and UU,

Note: This command should not be issued when an axis is in motion and it should be used sparingly because the flash memory has a limited number of write cycles.

Example:	Save the current parameter set to be the power up default set of values.
Enter:	APP;
Response:	None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
APP	AX – AS	Immediate	
APP	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: RDB, RDF, RDP

?AQ QUERY CURRENT AXIS

The ?AQ command reports the mode that the current axis is in, i.e. AA mode, AM mode, etc.

Example: Determine what mode the X axis is in.

Enter: AX;
?AQ

Response: =ax<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?AQ	AX – AS	Immediate	
?AQ	AA-AM	Immediate	
?AQ	AA/CD	Immediate	

AR AXIS R

The AR command directs all following commands to the R axis. Commands sent in this mode will be executed in the order in which they are received (subject to queued versus immediate execution) and are expected to be formatted for single-axis use.

Example: Move the R axis to absolute position -2468.

Enter: AR;
MA-2468;
GO;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AR	AX – AS	Immediate	
AR	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AA, AM, ?AQ, AT, AU, AV, AS, AX, AY, and AZ

AS AXIS S

The AS command directs all following commands to the S axis. Commands sent in this mode will be executed in the order in which they are received (subject to queued versus immediate execution) and are expected to be formatted for single-axis use.

Example: Move the S axis to absolute position -2468.

Enter: AS;
MA-2468;
GO;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AS	AX – AS	Immediate	
AS	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AA, AM, ?AQ, AT, AU, AV, AX, AY, AZ

AT AXIS T

The AT command directs all following commands to the T axis. Commands sent in this mode will be executed in the order in which they are received (subject to queued versus immediate execution) and are expected to be formatted for single-axis use.

Example: Move the T axis to absolute position -2468.

Enter: AT;
MA-2468;
GO;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AT	AX – AS	Immediate	
AT	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AA, AM, ?AQ, AU, AV, AR, AS, AX, AY, AZ

AU AXIS U

The AU command directs all following commands to the U axis. Commands sent in this mode will be executed in the order in which they are received (subject to queued versus immediate execution) and are expected to be formatted for single-axis use.

Example: Move the U axis to absolute position -2468.

Enter:
```
AU;
MA-2468;
GO;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AU	AX – AS	Immediate	
AU	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AA, AM, ?AQ, AT, AV, AR, AS, AX, AY, AZ

AV AXIS V

The AV command directs all following commands to the V axis. Commands sent in this mode will be executed in the order in which they are received (subject to queued versus immediate execution) and are expected to be formatted for single-axis use.

Example: Move the V axis to absolute position -2468.

Enter:
```
AV;
MA-2468;
GO;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AV	AX – AS	Immediate	
AV	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AA, AM, ?AQ, AT, AU, AR, AS, AX, AY, AZ

AX AXIS X

The AX command directs all following commands to the X axis. This is the default mode at power up or reset. Commands sent in this mode will be executed in the order in which they are received (subject to queued versus immediate execution) and are expected to be formatted for single-axis use.

Example: Make the X axis step at a rate of 5,000 steps/second.

Enter:

```
AX;
JG5000;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AX	AX – AS	Immediate	
AX	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AA, AM, ?AQ, AT, AU, AV, AR, AS, AY, AZ

AY AXIS Y

The AY command directs all following commands to the Y axis. Commands sent in this mode will be executed in the order in which they are received (subject to queued versus immediate execution) and are expected to be formatted for single-axis use.

Example: Examine the status of the Y axis.

Enter:

```
AY;
RA
```

Response: =PNNN<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AY	AX – AS	Immediate	
AY	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AA, AM, ?AQ, AT, AU, AV, AR, AS, AX, AZ

AZ AXIS Z

The AZ command directs all following commands to the Z axis. Commands sent in this mode will be executed in the order in which they are received (subject to queued versus immediate execution) and are expected to be formatted for single-axis use.

Example:	Move the Z axis 2,000 steps at a rate of 500 steps per second.

Enter:	AZ;
	VL500;
	MR2000;
	GO;

Response:	None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
AZ	AX – AS	Immediate	
AZ	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AA, AM, ?AQ, AT, AU, AV, AR, AS, AX, AY

BC SET BACKLASH COMPENSATION

The BC command sets the backlash compensation factor for the currently active axis. This is a numeric value of the number of steps output when a direction reversal occurs.

The numeric parameter must be between 0 and 50 for stepper motors and between -50 and +50 for servo motors.

RANGE:	0 ≤ backlash ≤50	Steppers
	-50 ≤ backlash ≤+50	Servos

Example: Set backlash compensation factor of axis X to 12 counts

Enter:
```
AX;
BC12;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?BC;	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: AN, AF, ?BC, BL, BS, BX

?BC REPORT BACK LASH COMPENSATION

The ?BC command reports the backlash compensation factor for the currently active axis. This is a numeric value of the number of steps added.

Example: Determine backlash compensation factor of axis X

Enter:
```
AX;
?BC
```

Response: =23<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?BC	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: AN, AF, BC, BL, BS, BX

BD SET THE DIRECTION OF THE GENERAL PURPOSE I/O BITS

This command sets the direction of the 16 general purpose I/O bits. Bit directions are encoded into a hexadecimal number. A one in a bit position specifies an output bit. A zero in a bit position specifies an input. Note the bit direction selection may be preserved by using the APP or APB commands to archive the controller's parameters.

RANGE: 0000 ≤ Bit Directions ≤ FFFF

Example:	To set bits 0 through 3 to outputs and bits 4 through 15 inputs.
Enter:	BD000F;
Response:	None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
BD#	AX – AS	Immediate	
BD#	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AF, AN, BC, BL, BS, BX

BH BIT HIGH

The BH command turns the selected general purpose output off (i.e. logic high). The default state of general purpose outputs is off at power up or reset, but can be changed by the BR command.

Note: Output bits should not be used as triggers for applying power to any device unless master power is applied separately and after the MAXv is fully configured. The states of the outputs are unpredictable during power-up and reset and can toggle several times before settling at a high level.

Range: 0 ≤ bit number ≤ 15

Example: Set general purpose bits 4 and 5 to high.

Enter:
```
BH4;
BH5;
```

Response: None.

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
BH#;	AX – AS	1	1	N/A
BH#;	AA-AM	2	1	N/A
BH#;	AA/CD	N/A	N/A	2

Related commands: AF, AN, BL, BS, BR, BX

BI BIPOLAR

The BI command sets the analog servo output of the current axis to bipolar. When bipolar is selected, a zero torque reference will result in a 0VDC output (+/- offset voltage). The analog output will range between +10VDC and -10VDC when bipolar is enabled. The BI command is valid only in the single axis mode and is the default mode at power up or reset.

Example: Set up servo axis X for bipolar operation.

Enter: AX;
BI;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
BI	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: DBI, DBN, ?SO, SVI, SVN, UN

BL BIT LOW

The BL command turns the selected general purpose output line on (i.e. logic low). The default states of all output bits at power-up are logic high (off), but can be changed by the BR command. The BS command can be used to set all outputs to a known state at once.

Note: Output bits should not be used as triggers for applying power to any device unless master power is applied separately and after the MAXv is fully configured. The states of the outputs are unpredictable during power-up and reset and can toggle several times before settling at a high level.

RANGE: 0 ≤ bit number ≤ 15

Example: Turn on output bits 4 and 5 after a move. Note that this is only valid for output bits; input bits cannot be modified.

Enter:
```
AX;
MA1000;
GO;
BL4;
BL5;
```

Response: None.

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
BL#;	AX – AS	1	1	N/A
BL#;	AA-AM	2	1	N/A
BL#;	AA/CD	N/A	N/A	2

Related commands: AF, AN, BR, BS, BX

BR OUTPUT RESET STATE

The BR command allows the user to define the state of the general output bits at power-up and reset of a MAX family motion controller. The BR command is valid for single axis, AM and AA modes of operation. It is an immediate command that requires no queue space. It is expected that the BR command is done manually to set up a system. The BR command is automatically archived.

Format: BR*n,m* where the parameter "***n*"** is the **Output Bit Number** and the parameter **"*m*"** is a binary one or zero depending on the desired state.

n is the output bit number from **0-15** or a question mark **('?')**. Note that an error is generated if ***n*** is a number not currently defined as an output.

If ***n*** is a '?' character, such as the command **BR?** , then the Output Reset State of all bits is reported in binary. For each bit that is defined as an output a 1 or 0 is displayed indicating the currently defined Output Reset State.
If the bit is defined as an input then '-' is displayed.

m is numeric or a question mark **(?)**
If ***m*** is numeric it represents the desired Output Reset State for that output bit – non-zero parameter defines the Output Reset State as HIGH and a zero parameter defines the Output Reset State as LOW.
If ***m*** is a **'?'** character, the currently defined Output Reset State of bit" ***n*** **"**is reported as 0 or 1. 0 represents LOW and 1 represents HIGH.

Defaults:
The factory default Output Reset State for all outputs is HIGH, which is represented as 1.
If an input is changed to an output, then that output is initialized to the state defined by the Output Reset State.
If an output is changed to an input, the Output Reset State of the former output remains defined, but has no effect.

Example: (assume default bit direction of BD=FF00, bits 8-15 outputs)

BR? - reports the Output Reset State of all output bits, displays 1 for HIGH and 0 for LOW.

BR15,0 – set I/O bit 15 to a LOW output state at reset.

BR11,1 – set I/O bit 11 to a HIGH output state at reset.

BR0,1 - COMMAND ERROR! – I/O bit 0 is not an output.

BR15,? – reports the currently defined Output Reset State. Displays the Reset State of I/O bit 15, where 1 is for HIGH and 0 is for LOW.

?BR - This command will act the same as **BR?**

BS BIT SET

Set all of the output bits to a known state at the same time. This command will affect all output bits, setting their states to the specified bit mask nearly simultaneously. The mask must be in ASCII hex format where the least significant bit (bit 0) is on the right. To set a line low, the corresponding bit in the hex mask must be a 0. A one (1) in any bit position will set the corresponding bit high.

Note: Output bits should not be used as triggers for applying power to any device unless master power is applied separately and after the MAXv is fully configured. The states of the outputs are unpredictable during power-up and reset and can toggle several times before settling at a high level.

Range: 0000 $\leq$ Hex number $\leq$ FFFF

Example: Assume I/O bit direction BD = FFFF, all outputs. Set output 0 high, 1-3 low, 4-6 high and 7-15 low (0071 = (hex) 0000000 01110001)

Enter: BS0071;

Response: None.

NOTE: Data written to input bits has no effect

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
BS#;	AX – AT	1	1
BS#;-	AA-AM	3	1
-	AA/CD	Not Valid	

Related commands: AF, AN, BH, BL, BX

BW WAIT FOR INPUT TO GO LOW

The BW command is just like the SW command except that it waits for the input line to reach a TTL low rather than a TTL high. Refer to the SW command for more detail.

RANGE: 0 ≤ Bit Number ≤ 15

Example: See the examples for the SW (see page 5-217) command

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
BW#;	AX – AS	1	1
BW#;	AA-AM	3	1
-	AA/CD	Not Valid	

Related commands: SW, WA, WT, WQ

BX BIT REQUEST IN HEX

The BX command returns the states of the general purpose I/O bits in a hex format. The rightmost character represents the least-significant-nibble (4 bits) and, if the nibble is rewritten as bits, the rightmost bit is the least-significant bit. An input set low will be represented as a binary 0 and a high as binary 1, similarly for an output.

Example: Assuming the default I/O bit direction of BD = FF00, bits 0-7 inputs and bits 8-15 outputs.

User input lines 0 and 2 are high and the remaining 6 inputs are low. User outputs 10 and 11 are high and the remaining 6 outputs are low.

Use the BX commands to verify these states.

Enter: BX

Response: 0C05<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
BX	AX – AS	Immediate	
BX	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: BH, BL, BS

CA CLEAR AXIS DONE FLAG

The CA command operates like the IC command, except it clears the done flag of the addressed axis only. In multi-axis modes, the CA command clears the flags of all selected axes. Unlike the IC command, CA will not clear other error flags in the status register such as slip and limit.

Example: After a multi-axis move, clear the Z axis done flag only.

Enter:
```
AA;
MR1000,2000,3000,4000;
GO;
ID
AZ;
CA;
```

Response: None.

Example: After a multi-axis move, clear the Y and Z axis done flags only.

Enter:
```
AA;
MR1000,2000,3000,4000;
GO;
ID
CA,1,1;
```

Response: None.

NOTE: In AA or AM mode, a null value in the argument list specifies the done bit of that axis is not to be cleared.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
CA	AX – AS	Immediate	
CAb,b,b,b,b,b,b,b;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: GD, IC, ID, II, IN, IP

CD CONTOUR DEFINE

The CD command enters contour definition mode and defines a constant velocity contour. The only way to exit this mode is to issue a CE or CK command. Commands following the CD command must be in multi-axis format and be commands valid in CD mode. All multi-axis commands entered in CD mode can address only those axes assigned in the CD command. No commands entered will be executed until a CX command is received which must be issued outside of CD mode.

The CD command takes up to eight parameters. Each parameter specifies a starting point for each axis to be involved in the contour in absolute coordinates. If an axis is not to be involved in the contour, its parameter position should be skipped just as it would be in any other multi-axis command. Commands issued within CD mode must be formatted to include only those axes indicated in the CD command. Those that are not in the CD command simply do not exist in the formatting of commands entered in CD mode. For example, if a CD command is issued that uses the X, Y, and T axes (such as CD100,300,,400;), commands entered within CD mode will only consider the X, Y, and T axes. An MT command that moves X to 200, Y to 600, and T to 200 would take the form: MT200,600,200;. Note the lack of a placeholder comma for the Z axis.

Contours that will include circular interpolation (CR) must be defined for only 2 axes in the CD command. Contours involving more than 2 axes may not use the CR command. The size of the contour definition buffer for the MAXv is 32,763 positions.

When the contour is executed, the MAXv will use the distance between the current position and the contour starting point to linearly ramp up each axis such that all involved axes reach a combined, vectored velocity equal to the value set with the CV command. If this distance is zero, no ramp will be generated resulting in an instantaneous jump to contour velocity. Most stepper systems cannot achieve this and servos will tend to oscillate wildly before settling down if at all. Care should be taken to allow sufficient ramping distance between the contour starting position and the current position when the CX command is issued.

Once the contour is completely executed, the MAXv will ramp the axes to a stop using the rate defined with the AC command. This ramp down will take each axis beyond the final point of the contour. Without manually calculating the ramp down distance for each axis, there is no way to force the contour to come to a complete stop at a predetermined point.

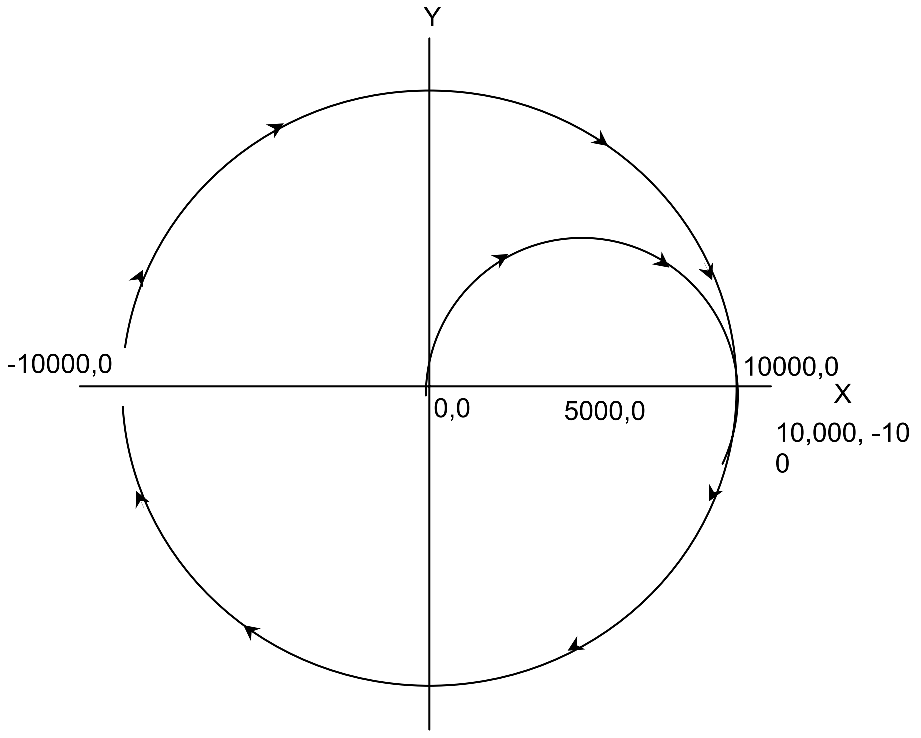
Example: The following demonstrates cutting a hole with a 10,000 count radius using constant velocity contouring and circular interpolation. The contouring velocity is set to 1000 pulses per second. A contour is then defined beginning at coordinates 0,0 on the X and Y axes. The auxiliary output of the Z axis is turned on, which could turn on the cutting torch or laser starting the cut at the center of the circle. A half circle is cut from the center to the outside of the hole, positioning the cutting tool at the start of the desired hole. The hole is then cut, the torch turned off, the stage stopped and the definition is complete. The stage is then positioned and the hole cut with the CX command. The AN and AF commands must have commas for all axes since they can all be addressed from within the contour definition.

Enter:
```
AA;
CV1000;
CD0,0;
AN,,1;
CR5000,0,-3.1415926;
CR0,0,-6.2831853;
AF,,0;
MT10000,-1000;
CE;
MT10000,0;
GO;
CX;
```

Response: None.

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
-	AX – AS	Not Valid		
CD#,#,#,#,#,#,#,#;	AA-AM	5 + number of axes in the contour		
-	AA/CD	N/A		

Related commands: AF, AN, BH, BL, CE, CK, CR, CV, CX, NV, MT



CE CONTOUR END

The CE command marks the end of the contour sequence. It will terminate the CD mode and, when executed, ramp to a stop and exit to the AA command mode. The end of the contour should contain at least a short linear segment just prior to the CE command to initialize the parameters for the deceleration of the stage.

Example: (see CD command on page 5-72)

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
-	AX – AT	Not Valid		
-	AA-AM	Not Valid		
CE	AA/CD	N/A	N/A	2

Related commands: CD, CK

CG CONTOUR PRIORITY

The CG command is a form of the CX command, use if CE is followed by a MT command. When CG is used to execute the contour, the following MT commands will be on hold until the contour execution is complete. After the contour has been executed, the MT commands that follow can be parsed.

CG is preferred over the CX when the MT commands are issued after a contour is executed. The CG command ensures that the MT command that follows starts from a known position. This makes for a more accurate calculation of the MT move.

Example: See page 5-72 (CD), Make sure that the hole is completely cut before executing the move to the new command position.

Enter:

```
AA;
CV1000;
CD0,0;
AN,,1;
CR5000,0,-3.1415926;
CR0,0,6.2831853;
AF,,0;
MT10000,-1000;
CE;
MT-10000,0;
GO;
CG;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	AX – AT	Not Valid	
CG	AA-AM	7*	2*
CG	AA/CD	Not Valid	

— If PA (power automatic) mode is selected add 2 to the command queue.
— If an aux bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.
— If the axis is stepper and encoder or servo add 1 to the command queue.
— Add the following queue requirements for the ramp types listed.

QUEUE REQUIREMENTS		
Axis Ramp Type	COMMAND QUEUE	ARGUMENT QUEUE
Linear (PF, LA)	4	4
Short Form Parablic (PN0, PR0)	4	8
All other Parabolic Forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

Related commands: AF, AN, BH, BL, CD, CK, CR, CV, CX, NV, MT

CK CONTOUR END AND KILL

The CK command will end the contour sequence, like the CE command, except there is no ramp down; i.e. motion will stop abruptly. It is used in place of the CE command.

NOTE: This command should be used with caution to prevent the stage from slipping or losing its correct position.

Example: Same scenario as CD command, but we want to end the contour with the minimum ramp down.

Enter:

```
AA;
CV1000;
CD0,0;
AN,,;
CR5000,0,-3.1415926;
CR0,0,-6.2831853;
AF,,0;
MT10000,-1000;
CK;
MT-1000,0;
GO;
CX
```

Response: None.

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
-	AX – AT	Not Valid		
-	AA-AM	Not Valid		
CK	AA/CD	N/A	N/A	2

Related commands: CD, CE

CN COSINE ON

The CN command enables cosine velocity ramps; i.e. half sinusoid acceleration profiles, for all axes. The cosine profile is not truncated in moves cannot reach full velocity, but instead the velocity is reduced sufficiently to preserve the cosine profile. This command should not be given while an axis is in motion or the results may be unpredictable. This command affects all axes even if issued in the single axis mode. The PF command is used to return to linear motion profiles. See the APP command (see page 5-57) to preserve the CN setting as the Power up/Reset ramp.

Example:	Set the board to be in cosine mode.
Enter:	CN;
Response:	None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
CN	AX – AS	5	24
CN	AA-AM	5	24
-	AA/CD	Not Valid	

Related commands: LA, PF, PN, PR, ?RT, SC, SR

CR CIRCULAR INTERPOLATION

The CR command defines a move in a circular pattern from the entry position. The first two parameters are the center of the circle in absolute units and the third parameter is the distance to move in radians. Positive radians equal counterclockwise movement. Negative radians equal counter clockwise movement. The radius of the circle is the linear distance between the current position and the first two parameters of the CR command.

The CR command generates a circle by breaking the circle into a series of linear segments. The number of segments will be equal to the total distance traveled divided by the contour velocity divided by the update rate. If finer resolution is required, the MT command may be used to produce smaller segments but the segments must be calculated by the user

RANGE:
Min Pos. Value≤Parameter 1 (First Coordinate for Center of Circle) ≤Max Pos. Value
Min Pos. Value≤Parameter 2 (Second Coordinate for Center of Circle) ≤Max Pos. Value

Note: The minimum and maximum position value depends on the settings used on MAXv, see the specifications for more detail.

Example: (see CD command on page 5-72)

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
-	AX – AS	Not Valid		
-	AA-AM	Not Valid		
CR#,#,#;	AA/CD	N/A	N/A	8

Related commands: CD, MT

CV CONTOUR VELOCITY

The CV command allows specification of a contouring velocity. It is executed from the AA mode before a contour definition. A contour defined by a CD command cannot be executed if followed by a CV command. Changing this parameter will make any previously defined contours invalid. The contour velocity defaults to 1000 at power up or reset. Use WQ between contour definitions to avoid having a CV associated with a second contour definition affect a prior contour still in motion. A CV cannot be issued between a CD and CE command. To change the contour velocity within a contour definition use the NV command.

RANGE: 1 ≤ CV ≤ 4,000,000

Example: (see CD command on page 5-72)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	AX – AS	Not Valid	
CV#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: CD, NV

CW CLEAR WHILE

The CW command breaks the WH loop upon execution of the remaining commands in the loop; i.e. the current execution of the loop is finished. The WH loop is always executed at least one time since the test for the flag is at the bottom.

Example: (see WH command page 5-237)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
CW	AX – AS	Immediate	
CW	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: WG, WH

CX CONTOUR EXECUTE

The CX command will execute the previously defined contour sequence. The stage must be positioned such that it can accelerate to speed by the absolute position specified by the CD command it is executing and must be traveling in the proper direction. Once a contour is defined it may be executed at any time by executing a CX command until it is replaced by another contour definition. The CX command cannot be placed within a loop or while construct.

Example: (see CD command on page 5-72)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	AX – AT	Not Valid	
CX	AA-AM	6*	2*
-	AA/CD	Not Valid	

- If the axis is a stepper and encoder or servo add 1 to the Command queue.
- If PA (power automatic) mode is active add 2 to the Command queue.
- If an auxiliary output bit settle time has been specified add 2 to the Command queue and add 1 to the Argument queue.

QUEUE REQUIREMENTS		
Axis Ramp Type	COMMAND QUEUE	ARGUMENT QUEUE
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	8
All other Parabolic Forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

Related commands: CD, CE, CK, ID

?DA PRINT A CUSTOM RAMP

This command will report the entries of a previously defined custom ramp table. RANGE: 1 ≤ Ramp Table Numbers ≤ 8

Example:	Print out custom ramp table #2
Enter:	?DA2;
Response:	DAR2<LF>
	DAB0.10000,0.20000<LF>
	DAB90000,0.80000<LF>
	DAB10000,1.00000<LF>
	DAE<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?DA#;	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: DAB, DAE, DAR, ?DE, ?DS

DAB DEFINE CUSTOM RAMP BREAKPOINT

The DAB command sets a breakpoint in a custom ramp table. This is the only command that should be used after DAR and before DAE. Each custom ramp may contain up to 25 breakpoints, each defined by a DAB command.

The DAB command takes two parameters; the first specifies the acceleration level that should be used to achieve the second parameter, velocity level. Both levels are expressed in terms of percentage in decimal format; i.e. 1.00 is 100%. At no time should a DAB command be entered in which the velocity parameter is less than the velocity parameter of the prior DAB. The MAXv will not flag this as a command error but the results of such a ramp will be unpredictable. Each DAB command sent should be equal to or greater than the DAB command that preceded it. It is the user's responsibility to make sure this command is used properly.

RANGE:
0.0 ≤ Parameter 1 ≤ 1.0
0.0 ≤ Parameter 2 ≤ 1.0

Example: See the DAR command (page 5-85) for a complete example of a custom profile.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DAB#,#;	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?DA, DAR, DAE, ?DS, ?DE, SR

DAE END CUSTOM RAMP DEFINITION

The DAE command terminates a custom ramp table definition initiated by the DAR command.

Example: See the DAR command (page 5-85) for a complete custom ramp table definition.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DAE	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?DA, DAR, DAB, ?DS, ?DE, SR

DAR BEGIN CUSTOM RAMP DEFINITION

The DAR command starts the definition of a custom ramp table. A parameter supplied with this command, from 1 to 8, specifies which ramp table to create. If a ramp table by that number has already been defined, it will be overwritten.

Once the DAR command has been issued, only the DAB and DAE commands will be valid. A series of ramp table breakpoints may be entered using the DAB command which define the profile breakpoints for this ramp table. Up to 25 breakpoints may be defined but a smaller number may be used. A ramp table containing no breakpoints is invalid and will result unpredictably if used.

RANGE: 1 ≤ DAR ≤ 8

Example:	Create a ramp table definition resembling a jerk-limited linear profile.	
Enter:	DAR3;	* Store as table #3
	DAB.1,.05;	* Ramp at 10% of AC until 5% of VL
	DAB.3,.1;	* Ramp at 30% of AC until 10% of VL
	DAB.9,.9;	* Ramp at 90% of AC until 90% of VL
	DAB.3,.95;	* Ramp at 30% of AC until 95% of VL
	DAB.1,1;	* Ramp at 10% of AC until 100% of VL
	DAE;	* End table definition
Response:	None.	

QUEUE REQUIREMENTS

FORMAT	MODE	COMMAND	ARGUMENT
DAR#;	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?DA, DAB, DAE, ?DE, ?DS, SR

?DB REPORT DIRECTION BIT LOGIC

The ?DB command reports the direction bit logic: inverted or normal. If the direction bit is low when moving positive, this command will return 'normal'. If the direction bit has been inverted, this command will return inverted.

Example: Report whether the direction bit for the T axis is low or high when making positive moves

Enter: AT;
?DB

Response: =i<LF> (The inverted result indicates the T axis direction bit is high for positive moves)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?DB	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: DBI, DBN, ?SV

DBI INVERT DIRECTION BIT

The DBI command inverts the logic of the direction control output of the addressed axis or axes. By default, the direction output of an axis is a TTL low when traveling in the positive direction and high when traveling negative. After using the DBI command, the direction bit will be high when traveling positive and low when traveling negative. This is useful for inverting the logical direction of a motor when the encoder counts opposite the motor direction. This command can be canceled using the DBN command. To make this the default at power up or reset, use the APP command.

Example: Set the direction outputs for axes Z and T to output high when traveling positive and low when traveling negative. Leaves X and Y as they are.

Enter:

```
AZ;
DBI;
AT;
DBI;
Or
AA;
DBI,,1,1;
```

Response: None.

Note: In AA or AM mode a null value in the argument list specifies that the sense of that direction bit is not to be changed.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DBI	AX – AS	1	0
DBIb,b,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: BI, ?DB, DBN, SVI, SVN, UN

DBN NORMALIZE DIRECTION BIT

The DBN command normalizes the logic of the direction control output of the addressed axis or axes, returning their output logic to default; i.e. TTL low when traveling in the positive direction and high when traveling negative. This command negates the effect of the DBI command. To make this the default at power up or reset when DBI has already been made the default, use the APP command.

Example: Set the direction outputs for axes Z and T to default output logic; i.e. output high when traveling positive and low when traveling negative. Leave X and Y as they are.

Enter:
```
AZ;
DBN;
AT;
DBN;
Or
AA;
DBN,,1,1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DBN	AX – AS	1	0
DBNb,b,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: BI, ?DB, DBI, SVI, SVN, UN

DC DECELERATION

The DC command sets a deceleration rate overriding the AC parameter when the GU command is used to initiate a move. Only the GU command will use the DC value. The deceleration rate defaults to 200,000 and will take on whatever value is entered via the AC command. Therefore, the DC command must be reentered after using AC if a different deceleration rate is desired.

RANGE: 1 ≤ DC ≤ 8000000

Example: Send the Y axis on a 100,000 count move that accelerates at 100,000 counts per second per second up to 50,000 counts per second and decelerates at 20,000 counts per second per second.

Enter:
```
AY;
AC100000;
DC20000;
VL50000;
MR100000;
GU;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DC#;	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: AC, GU, RC, VB, VL

?DC REPORT DECELERATION RATE

The ?DC command reports the deceleration rate that will be used by the GU command.

RANGE= 1≤ DC ≤ 8000000

Example: Determine deceleration rate of none symmetric motion profiles of X axis.

Enter: AX;
?DC

Response: =20000<LF >

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?DC	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: AC, DC, GU

?DE REPORT A CUSTOM RAMP TABLE ENTRY

The ?DE command will return a specific entry from a specific custom ramp table. The first parameter specifies the table to examine and the second parameter specifies the entry to return from the table.

RANGE:
1 ≤ Parameter1 ≤ 8
1 ≤ Parameter2 ≤ 25

Example: We can't remember what the 23rd breakpoint in table 4 was set to. Use the ?DE command to find out.

Enter: ?DE4,23;

Response: <LF> (there is no 23rd entry in table 4)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?DE#,#;	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?DA, DAB, DAE, DAR, ?DS

DOV OUTPUT AUXILIARY DAC VOLTAGE

This command sends the specified analog voltage out the specified auxiliary DAC channel.

DOV#,*Volts*;
# specifies the auxiliary DAC channel to set

RANGE: 0 ≤ # ≤ 1

Volts specifies the voltage

RANGE: -10.0 ≤ # ≤ +10.0

Example:	Set the auxiliary DAC channel 1 to 5.5 volts
Enter:	DOV1,5.5;
Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
DOV#,*Volts*;	AX – AS	Immediate	
DOV#,*Volts*;	AA-AM	Immediate	
-	AA/CD	Not Valid	

DOZ SET AUXILIARY DAC ZERO OFFSET

This command set a voltage offset for the specified auxiliary DAC channel. The zero offset is determined experientially such that setting the auxiliary DAC output to zero volts results in a reading of zero volts on the auxiliary DAC's output.

DOZ#,*Volts*;
# specifies the auxiliary DAC channel to set the zero offset

RANGE: 0 ≤ # ≤ 1 OR?

Volts specifies the zero offset voltage

RANGE: -10.0 ≤ # ≤ +10.0

Example 1:	Set the zero offset of auxiliary DAC channel 0 to -0.15 volts
Enter:	DOZ0,-0.15;
Response:	None

Example 2:	Report the zero offset of auxiliary DAC channel 0
Enter:	DOZ0,?;
Response:	=-0.15<LF>

QUEUE REQUIREMENTS

FORMAT	MODE	COMMAND	ARGUMENT
DOZ#, *Volts*;	AX – AS	Immediate	
DOZ#, *Volts*;	AA-AM	Immediate	
-	AA/CD	Not Valid	

?DS REPORT THE SIZE OF A CUSTOM RAMP TABLE

The ?DS command returns the number of segments in the specified custom ramp table.

RANGE: 1 ≤ ?DS ≤ 8

Example:	The 3rd custom ramp should be 17 breakpoints long. Make sure this is true.
Enter:	?DS3;
Response:	=17<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?DS#;	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?DA, DAB, DAE, DAR, ?DE

EA ENCODER STATUS

The EA command returns encoder status of the currently addressed axis or axes in the following format:

EA COMMAND RESPONSE DESCRIPTION		
CHAR	SENT	DESCRIPTION
1	E	Slip detection enabled
	D	Slip detection disabled
2	E	Position maintenance enabled
	D	Position maintenance disabled
3	S	Slip or stall detected (reset by execution of EA command)
	N	No slip or stall detected
4	P	Position Maintenance within deadband
	N	Position not within deadband
5	P	Encoder is at home condition
	N	Encoder is not at home
6	LF	Line feed

Example: Examine the status of the Y axis encoder.

Enter: AY;
EA

Response: DDNNN<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
EA	AX – AS	Immediate	
EA	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: QA, QI, RA, RI

EH DEFINING ENCODER HOME

HH and HL commands are active in HE mode. This allows for an encoder home operation with the Home Switch being active HIGH. Previous OMS controllers required and active LOW Home Switch to do an encoder home operation, and this will be the "default" setting here as well. This command sets the true states of the A, B, and Index for the Encoder Home condition.

The New **EHbbb command provides** variants that will completely determine the logic states of encoder signals A, B, and Index that make for a true encoder home event. This allows the user more flexibility to adapt to the schemes of most encoder manufacturers. All **bbb** are ones **(1)** or Zeros **(0)** representing the states of the Index, Phase B and Phase A respectively.

Note: The EH command is only valid for closed-loop Steppers and Servo axes. If the command is entered for an open-loop stepper axis, a command error will result. The EH command is an immediate command, and therefore requires no queue space

Format: AX-AS: EHbbb
All b's are 0 or 1 representing the states of Index, Phase B, and Phase A respectively. A 1 represents a high true state and a 0 represents a low true state.

AA-AM: EHbbb,bbb,...
If the parameter bbb is missing for an axis, then the states for that axis are left unchanged.

Example: AX**-**AS**:** for the currently selected axis, the **EH** command: sets true states as follows:

EH100;	Index high/PhaseB low/PhaseA low
EH010;	Index low/PhaseB high/PhaseA low
EH001;	Index low/PhaseB low/PhaseA high

AA**-**AM**:** for the multi axis modes the **EH** command sets true states as follows:
EH001,,,100; sets X-Axis Index low/PhaseB low/PhaseA high and T axis Index high/PhaseB low/PhaseA low.
All other axes are unaffected.

QUEUE REQUIREMENTS

FORMAT	MODE	COMMAND	ARGUMENT
EHbbb	AX – AS	Immediate	
EHbbb,bbb,bbb,bbb,bbb,bbb,bbb,bbb	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: ?EH

?EH QUERY ENCODER HOME

The ?EH reports the state of the Index, Phase A and Phase B true states in both single axis and multi-axis modes. It is an immediate command and does not require any queue space.

Example: **?EH** AX**-** AS Single axis mode

The controller's response to the ?EH command has the following meaning:

000	Index low/Phase B low/Phase A low
010	Index low/Phase B high/Phase A low
101	Index High/Phase B low/Phase A high

Example: **?EH** AA**/**AM Multi-axis mode

The controller's response to the ?EH is:
000,---,111, ---,---,---,---,101 and has the following meaning:
X axis: Index low/Phase B low/Phase A low
Y axis: Not an encoder axis
Z axis: Index high/Phase B high/Phase A high
T axis: Not an encoder axis
U axis: Not an encoder axis
V axis: Not an encoder axis
R axis: Not an encoder axis
S axis: Index high/Phase B low/Phase A high

NOTE: The encoder home pattern set by the EH command and the true logic state of the home switch as set by the HL or HH command are archived in the flash with the parameter archive commands.

The factory default encoder home pattern and home switch state would be the same as the current OMS standard home pattern, that is, Encoder Home pattern is Index HIGH, Phase A HIGH, Phase B LOW and the Home Switch is active LOW.

ER ENCODER RATIO

The ER command allows specification of encoder:motor ratio for position maintenance mode. This command is not designed for use with servo motors. ER takes two arguments: encoder counts and motor counts. Both parameters must be integers unless user units (UU) are enabled. The ratio need not be per full revolution; reduce the fraction as far as possible and use those values.

The factory default ratio is 1:1. See the APP command (page 5-57) to preserve the ER settings as the Power up/Reset values.

NOTE: That if an encoder ratio has been defined by the ER command, then the slip tolerance defined by the ES command is defined in encoder units and not in motor units.

Parameter 1 = Encoder Counts
Parameter 2 = Motor Counts

Example: You have an encoder connected to a stepper motor through a series of gears. When the motor steps 25,000 times, the encoder produces 10,000 counts. Set up an encoder ratio so hold mode will work correctly.

Enter: ER10000,25000;
or
ER2,5;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ER#,#;	AX – AS	1	0
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?ER

?ER REPORT MOTOR:ENCODER RATIO

The ?ER command reports the motor-to-encoder ratio as set with the ER command.

Example: Find out what the last ER command sent was.

Enter: ?ER

Response: =2.000000<LF>
(The encoder produces 1 count for every 2 steps of the motor.)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?ER	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ER

ES ENCODER SLIP TOLERANCE

The ES command parameter specifies the slip detection tolerance before slip or stall is flagged in the status register and in the RL command response. The encoder may get off target by as much as this value before the MAXv will consider the axis slipped. This mode must be turned on with an IS command and off with an HF command. The factory default value is 1. For servo motors, the default is 32767. This value determines when the PID is disabled due typo excessive following error. See APP command (see page 5-57) to preserve the ES settings as the Power up/Reset values.

NOTE: If an encoder ratio is defined with the ER command, then the slip tolerance defined by the ES command is in encoder units and not in motor units.

NOTE: The GS command allows an open-loop stepper axis to perform a limited form of slip detection by using the home switch as a reference when executing a move command.

STEPPER RANGE: 0 ≤ ES ≤ 65,535

SERVO RANGE: 0 ≤ ES ≤ 327,670

Example: Your application can tolerate being up to 5 steps from the desired position before the controlling program should be notified of a slip condition.

Enter:
ES5;
IS;

Response: None.

Note: That if an encoder ratio has been defined by the ER command, then the slip tolerance defined by the ES command is defined in encoder units and not in motor units.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ES#;	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?ES, IS, RL, TF, TN

?ES REPORT ENCODER SLIP TOLERANCE

The ?ES command reports the current value of the slip detect tolerance assigned to an axis.

Example:	Report the current value for encoder slip detection tolerance
Enter:	?ES
Response:	=15<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?ES	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ES

ET ENCODER TRACKING

The ET command turns on the encoder tracking mode. The axis will track its encoder input, thus allowing one axis to follow the activity of another or a thumbwheel for manual positioning or the movement of another device that produces a signal compatible to the encoder inputs. No acceleration or deceleration ramps are generated. The axis will duplicate the encoder input. The ER command allows the user to scale the motor's movements relative to the encoder. This command is intended to be used with stepper motors with encoders and not with servo motors.

Example: Set up the Y axis so it will follow its encoder input.

Enter:
```
AY;
ET;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ET	AY – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ER, HF

FL FLUSH

The FL command will flush an individual axis' queues. This command is similar in operation to the KL and ST commands except that current motion will remain unaffected by the FL command. All unexecuted commands remaining in the current axis queue will be flushed upon receipt of this command.

Example: Several motion commands have been sent to the X axis but a situation arose and now those commands must be cleared out. The currently executing motion must be allowed to complete to avoid damage to the product.

Enter: AX;
FL;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
FL	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: KL, KS, SA, ST

FP FORCE POSITION

The FP command will flush the command queue and attempt to stop at the specified position. The axis will overshoot if there is insufficient distance left to stop at the programmed acceleration. This command should not be given to a servo axis while it is in motion; the results may be unpredictable. Force the axis to stop at a specified position.

Example:	Force axis to stop at 25,000.
Enter:	FP25000;
Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
FP#;	AX – AS	2*	1*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

If PA (power automatic) mode is active add 1 to the command queue
If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: MM, MP, MV, SP

FX ENABLE AXIS GANTRY MODE

The FX command pairs the current axis with the X axis to form a gantry crane. (i.e. the axis follows the X axis.)

Example:	Pair the X and Y axis to form a gantry crane.

Enter:	AY;
	FX;

Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
FX	AY – AT	1	0
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

GD GO AND RESET DONE

The GD command may be substituted for a GO command. It will reset the done flags and then initiate the move which has been previously programmed with such commands as MA, MR, MT, and ML just as the GO command does. In single axis mode, only the done flag for the selected axis will be reset.

In AA mode, all the done flags will be reset. In the AM mode, only the axes involved in the move will be reset. This allows the host to reset the interrupts on the axis involved in the next move without affecting other axes which may be still active. Note that this command is probably only useful in applications where commands are queued in advance since the interrupt may be reset before the host has the opportunity to service it if the GD command is waiting in the queue.

If this command is issued without having defined a move, the results are undefined. Issuing a GD command to execute an already-executed move also has undefined results. Only one GD command should be issued per defined move.

Example: In the single axis mode, move the Y axis 12345 counts in the negative direction and set the done flag when the move is completed. Then clear the done flag, move the motor 12345 counts in the positive direction, and set the done flag again when the move is completed.

Enter:
```
AY;
MR-12345;
GO;
ID;
MR12345;
GD;
ID;
```

Response: None.

Example: In AA mode, perform a linear absolute move with the X and Y axes to the position 10000,20000 and set the done flag when the move is completed. Then clear the done flag, perform a linear relative move on both axes moving the X axis 10000 steps in the negative direction and the Y axis 20000 steps in the negative direction, and set the flag once again.

Enter:
```
AA;
MT10000,20000;
GO;
ID;
ML-10000,-20000;
GD;
ID;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
GD	AX – AS	6*	0*
GD	AA-AM	8*	0*
-	AA/CD	Not Valid	

If the axis is stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.
If PA (power automatic) mode is active add 2 to the Command queue
If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

Related commands: MA, MR, MT, ML, GO, GS, GU, GN

GN GO AND NOTIFY WHEN DONE

The GN command will initiate a move which has been previously programmed with such commands as MA, MR, MT, and ML and set the axis done flags when the move is complete. No operand is required with the GN command. If this command is issued without having defined a move, the results are undefined. Issuing a GN command to execute an already-executed move also has undefined results. Only one GN command should be issued per defined move.

Example:	In the single axis mode, move the X axis to absolute position 12345.

Enter:	AX;
	MA12345;
	GN;

Response:	None

Example:	In the AA mode, move the X axis 2468 steps in the positive direction and the Y axis 2468 steps in the negative direction.

Enter:	AA;
	MR2468,-2468;
	GN;

Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
GN	AX – AS	6*	0*
GN	AA-AM	7*	0*
-	AA/CD	Not Valid	

— If the axis is a stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.
— If PA (power automatic) mode is active add 2 to the command queue
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: GD, GN, GS, GU, MA, ML, MR, MT

GO GO

The GO command will initiate a move which has been previously programmed with such commands as MA, MR, MT, and ML. No operand is required with the GO command. If this command is issued without having defined a move, the results are undefined. Issuing a GO command to execute an already-executed move also has undefined results. Only one GO command should be issued per defined move.

Example: In the single axis mode, move the X axis to absolute position 12345.

Enter:
```
AX;
MA12345;
GO;
```

Response: None

Example: In the AA mode, move the X axis 2468 steps in the positive direction and the Y axis 2468 steps in the negative direction.

Enter:
```
AA;
MR2468,-2468;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
GO	AX – AS	5*	0*
GO	AA-AM	7*	0*
-	AA/CD	Not Valid	

— If the axis is a stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.
— If PA (power automatic) mode is active add 2 to the command queue
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: GD, GN, GS, GU, MA, ML, MR, MT

GS GO AND MONITOR SLIP TRIGGER

The GS command works exactly like the GO command except that the home switch will be monitored during the motion. If the home switch becomes active the slip flag will be set for the axis. The host application can read the slip flag and see that the home switch was encountered during the move. This is useful in applications that register slip conditions by means other than encoder position verification; in fact, this command is not valid in controls with encoder feedback which includes servo motors.

If this command is issued without having defined a move, the results are undefined. Issuing a GD command to execute an already-executed move also has undefined results. Only one GD command should be issued per defined move.

Example:	Move the X axis 50,000 counts in the positive direction. If the motor slips it will close a switch wired to the home input of the X axis. Monitor this switch during the move and set the slip flag for axis X if the switch becomes active.

Enter:	AX;
	MR50000;
	GS;

Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
GS	AX – AS	5*	0*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

— If the axis is a stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.
— If PA (power automatic) mode is active add 2 to the command queue
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: GD, GN, GO, GU, MA, MR

GU GO ASYMMETRICAL

The GU command initiates a previously defined move using the AC value for acceleration and the DC value for deceleration. This command may be used with only one axis at a time; i.e. it is not valid with the ML and MT commands.

If this command is issued without having defined a move, the results are undefined. Issuing a GU command to execute an already-executed move also has undefined results. Only one GU command should be issued per defined move. GU is used only with linear acceleration ramps. Use S-curve command (see AJ, ?AJ, SS) for defining more complex asymmetrical motion profiles.

Example: Move the Y axis to position 1,500 using the current acceleration and velocity and a deceleration of 5,000 counts per second per second.

Enter:
```
AY;
DC5000;
MA1500;
GU;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
GU	AX – AS	3*	0*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

— If the axis is a stepper and encoder or servo axis add 1 to the command queue and add 2 to the argument queue.
— If PA (power automatic) mode is active add 2 to the command queue
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: AC, DC, GD, GN, GO, GS, MA, MR

HD HOLD DEADBAND

The HD command specifies dead band counts for position maintenance mode. If the encoder count is within this distance of target, it is considered in position and no further correction will be made. This parameter interacts with the HG and HV commands; i.e. a larger dead band will allow a larger gain parameter in many applications. This command is designed to work with stepper motor applications using encoders and is not designed for use with servo motors. The factory default value is zero. See the APP command to preserve the HD settings as the Power up/Reset values.

RANGE: 0 ≤ HD ≤ 64,000

Example: (see HN command page 5-119)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HD#;	AX – AS	1	1
HD#,#,#,#;	AA-AM	1	1
-	AA/CD	Not Valid	

Related commands: ?HD, HF, HG, HN, HV

?HD REPORT POSITION MAINTENANCE DEADBAND

The ?HD command reports the current setting of the HD command. This command will only work on stepper axes with encoders.

Example: Find out what HD was last set to.

Enter: ?HD

Response: =5<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?HD	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HD, ?HG, ?HV

HE HOME ENCODER

The HE command enables encoder index, and phase A and B signals to be considered in addition to the home switch input when an HM or HR command is executed. This command does not execute the home motion which must be initiated with an HM, HR, KM, or KR command. Default home behavior (considering only the home switch for home commands) may be reestablished via the HS command. In the HE mode, home is defined as the logical AND of the encoder index, the external home enable and the encoder quadrant where channel A is positive and channel B is negative. The default home logic expressed in Boolean terms is:

home = phase_A ∗ / phase_B ∗ index ∗ / home_switch

The default logic can be changed to any required combination with the use of HL and HH commands to set the true home switch state and the EH commands to set the encoder signal (Index, A and B) states required for a true home condition.

Example:	Set up the Y axis so it will use the encoder signals to recognize the home position.

Enter:	AY;
	HE;
	or
	AA;
	HE,1;

Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HE	AX – AS	Immediate	
HEb,b,b,b;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: EH, HH, HL, HM, HR, HS, KM, KR

HF HOLD OFF

The HF command disables position hold, stall detection and tracking modes, for stepper with an encoder axis. If the current axis is a servo, this command will open the loop and turn off the PID. If the current mode is multi-axis, all axes will go into open-loop mode. This is the default mode at power up or reset.

Example: Turn off encoder hold mode on the X axis.

Enter: AX;
HF;

Response: None

NOTE: In AA or AM mode, a null value in the argument list specifies feedback for that axis is not to be disabled.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HF	AX – AS	2*	0*
HF	AA-AM	2*	0*
-	AA/CD	Not Valid	

***Values in table are for a stepper with encoder axis. For a servo axis the command queue requires 1, and the argument queue requires 1.**

Related commands: HN, ?PM

HG HOLD GAIN

The HG command allows the user to specify the position hold gain parameter. This gain parameter is multiplied by the position error in determining the velocity during correction. The velocity used will not exceed the value set with the hold velocity (HV) command. This command is designed to work with stepper motor applications using encoders and is not designed for use with servo motors. The parameter should be set experimentally by increasing it until the system is unstable then reducing it slightly below the threshold of stability. The factory default value is 1. See the APP command (page 5-57) to preserve the HG settings as the Power up/Reset values.

RANGE: 1 ≤ HG ≤ 32,000

Example: (see HN command (page 5-119))

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HG#;	AX – AS	1	1
HG#,#,#,#;	AA-AM	1	1
-	AA/CD	Not Valid	

Related commands: HD, HF, ?HG, HN, HV

?HG REPORT POSITION MAINTENANCE GAIN

The ?HG command reports the current setting of the position maintenance hold gain constant for the current axis. This command works only with stepper + encoder axes.

Example: Position corrections seem slow. Check the setting of HG to be sure it is correct.

Enter: ?HG

Response: =100<LF >

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?HG	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?HD, HG, ?HV

HH HOME HIGH

The HH command sets the sense of the home switch on the current axis to active TTL high. This command allows TTL logic high to be treated as the "true" state for applications where this is more convenient. Once this command has been sent to the MAXv, active high homes can be made the power-up default by using the APP command; refer to the APP command for details. This command sets the home switch "true" state to TTL logic high for both the HS and HE modes of the home operation.

Example: (see HL command on page 5-115)

Note: In AA or AM modes, a null argument in the parameter list specifies that axis home switch to be unchanged.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HH	AX – AS	1	0
HHb,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: EH, HE, HL, HM, HR, HS, KM, KR

HL HOME LOW

The HL command sets the sense of the home switch on the current axis to active TTL low. This is the power-up and reset default "true" state unless HH has been saved as a user power-up default (See the APP command.). This command sets the "true" state of a home input to a TTL logic low.

This command sets the home switch "true" state to TTL logic low for both HS and HE modes of home operation.

Example: The stage is moved through home with the home switch set for active low at low speed to meet the less than 2048 steps per second requirement of the home command.

Enter:
```
AX;
VL2000;
HL;
HM0;
```

Response: None

Note: In AA or AM modes, a null argument in the parameter list specifies that axis home switch to be unchanged.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HL	AX – AS	1	0
HLb,b,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: EH, HH, HE, HS, HM, HR, KM, KR

HM HOME

The HM command will cause the current axis or specified axes to move in the positive direction at the predefined velocity until the home is detected for each axis. The position counters will be initialized to the positions supplied as parameters.

How home is detected depends on whether the HS or HE modes has been selected. The default mode, HS, detects home based on an external home input only. The HE mode detects home based on the external home input and the encoder signals (Index, A and B).

The velocity should be less than the update rate to maintain accuracy of the home position loaded. A velocity set to less than or equal to the update rate will provide a homing accuracy of +/-0 counts. Every multiple of the update rate adds +/-1 count to the error range.

Each axis will, when home is detected, reset its position counter to the parameter specified in the HM command. Once the counter is reset, the axis will ramp to a stop at the rate specified previously via the AC command. This will result in the axis stopping beyond the home switch so care should be taken to ensure adequate stopping distance is available beyond the switches. The axis can be easily returned to the precise home position by using an MA command after the HM command.

If no parameter is specified in single axis modes, the HM command will use zero as a default value. Parameters must be specified in multi-axis modes to inform the MAXv which axes are to be homed.

Example: Find the physical home position of the X axis of the stage. The motor runs until the home switch input is activated and then initializes the position counter to the parameter supplied. Since the motor decelerates to a stop after reaching home, it is necessary to do an MA to the same position as specified in the home command if it is desired to physically position the device at home. The following commands will find home, initialize it to 1000 counts, then return to home. In many cases it will not be necessary to return home, only find the position and synchronize the controller to it.

Enter:
```
AX;
VL1000;
HM1000;
MA1000;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HM#;	AX – AS	3*	1*
HM#,#,#,#,#,#,#,#;	AA-AM	3*	11*
-	AA/CD	Not Valid	

— If the axis is a stepper and encoder or servo axis add 2 to the argument queue.
— If PA (power automatic) mode is active add 2 to the command queue.
— If an auxiliary output bit settle time has been specified add 2 to the command queue and 1 to the argument queue.
— If the last profile move, just prior to this home command was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the home. This will add to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	6
All other Parabolic forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: HE, HH, HL, HR, HS, KM, KR, LO, LP, RP

?HM REPORT HOME STATE SELECTION

Reports homing mode selected for an axis. The possible responses are 'switch' for home switch (HS) mode and 'encoder' for home encoder (HE) mode.

Example: Report the homing mode assigned to the X axis.

Enter: AX;
?HM

Response: =s<LF> or =e<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?HM	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HE, HH, HL, HM, HR, HS, KM, KR

HN HOLD ON

For servo axes, the HN command closes the loop, enabling the PID. For servo axes, this mode is disabled when the HF command is entered, when the servo error becomes too large or a limit is encountered.

For stepper with encoder feedback axes, the HN command enables position correction after a move and activates the HD, HG and HV commands for stepper axes with encoders. For stepper axes with encoders, this mode will be canceled (as though via an HF command) if an CG, CX, HM, HR, MV, SO, SP command is entered, if a limit is encountered or the maximum allowable position error is executed.

Example: (Stepper) The following commands could be used to set up the position correction mode on a stepper axis. This sequence sets up a move velocity of 100,000 steps per second and an acceleration of 500,000 steps per second per second. The position correction velocity is set for 50,000 steps per second, a dead band of 10 steps and correction gain of 2,000. The correction is then enabled. A 200,000 step move is performed, then that position is maintained within the 10 step dead band until commanded to a new position.

Enter:
```
AX;
VL100000;
AC500000;
HV50000;
HD10;
HG2000;
HN;
MR200000;
GO;
```

Response: None

Example: (Servo) Close PID loop or X axis after having modified one of the PID parameters.

Enter:
```
AX;
HN;
```

Response: None

NOTE: In AA or AM mode, a null value in the argument list specifies that encoder feedback is not to be enabled for that axis.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HN	AX – AS	1*	0
HNb,b,b,b;	AA-AM	1*	0
-	AA/CD	Not Valid	

*** Values in table are for a stepper with encoder axis. For a servo axis the command queue requires 2, and the argument queue requires 1.**

Related commands: HF, HD, HG, HV, ?PM

HR HOME REVERSE

The HR command will cause the current axis or specified axis to move in the negative direction at the predefined velocity, until the home is detected for each axis. When the home input is detected, the position and encoder counters are loaded with the parameter(s) supplied in the HR command. Then the axis is ramped to a stop. It behaves exactly like the HM command, except it travels in the reverse direction.

How home is detected depends on whether the HS or HE modes has been selected. The default mode, HS, detects home based on an external home input only. The HE mode detects home based on the external home input and the encoder signals (Index, A and B).

Example: In a long stage it may be awkward to travel the full distance to home at less than the update rate. The following will get close to home at higher speed, and then refine the position at lower speed in the reverse direction.

Enter:
```
AX;
VL100000;
HH;
HM;
VL1000;
HL;
HR;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HR#;	AX – AS	3*	1*
HR#,#,#,#;	AA-AM	3*	1*
-	AA/CD	Not Valid	

— If the axis is a stepper and encoder or servo axis add 2 to the argument queue.
— If PA (power automatic) mode is active add 2 to the command queue.
— If an auxiliary output bit settle time has been specified add 2 to the command queue and 1 to the argument queue.
— If the last profile move, just prior to this home command, was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the Home. This will add to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	6
All other Parabolic forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: HE, HH, HL, HM, HS, KM, KR, LO, LP

HS HOME SWITCH

The HS command disables HE mode and returns the MAXv to the default home behavior. Default behavior defines a home state to be active when the home switch input is either low in default or HL mode or high when in HH mode. This mode can also be used with encoders which contain internal home logic by connecting their output to the MAXv home input for the appropriate axis. The active level of this input may be controlled by the HH and HL commands.

Example: Set up the Y axis so it will ignore the encoder signals and only use the home input to recognize the home position.

Enter: AY;
HS;
or
AM;
HS,1;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HS	AX – AS	Immediate	
HSb,b,b,b;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: HE, HH, HL, HM, HR, KM, KR

?HS REPORT HOME SWITCH TRUE STATE SELECTION

The ?HS command reports if the true state of the home switch is set to be active high or low.

Example: Determine if true home switch selection of Z axis is high.

Enter: AZ;
?HS

Response =h<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?HS	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HE, HH, HL, HM, HS, KM, KR, LO, LP

HV HOLD VELOCITY

The HV command specifies the maximum velocity to be used when correcting position error. The factory default setting is zero; some value must be set for position correction to occur at all. See the APP command to preserve the HV settings as the Power up/Reset values. This command is not designed for use with servo motors.

Hold gain (HG) will be used to scale the HV value based on the total error that must be corrected. In most cases the HV value will never be reached unless the position error is very wide or the HG value is set very high.

RANGE: $1 \leq HV \leq 4{,}000{,}000$

Example: (see HN command, see page 5-119)

NOTE: In AA or AM mode, a null value in the argument list specifies that the hold velocity parameter is not to be changed for that axis.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
HV#;	AX – AS	1	1
HV#,#,#,#;	AA-AM	1	1
-	AA/CD	Not Valid	

Related commands: HD, HF, HG, HN, ?HV

?HV REPORT POSITION MAINTENANCE VELOCITY

The ?HV command reports the current setting of the position maintenance hold velocity limits for the current axis. This command works only with stepper + encoder axes.

Example: Check the peak correction velocity for the T axis

Enter: AT;
?HV

Response: =20000<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?HV	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?HD, ?HG, HV

IC INTERRUPT CLEAR

The IC command or the ASCII character Control-Y (hex 19) is used to clear the done and error flags in the status register and the done flag register on MAXv, otherwise the axis would always appear to be "done". This command will be executed immediately and will usually be placed in the done and error handler interrupt service routine to clear the interrupt and the associated flags. The flags may be polled by an RA or RI command which will also reset the flags.

Note: That this command is not recommended in Windows environments using OMS-supplied device drivers and DLLs. The preferred method is to use the device driver and DLL to handle the reporting and clearing status flags.

Example:	Clear the flags after an X axis move relative of 5000 steps was flagged as done when an ID executes.

Enter:	AX;
	MR5000;
	GO;
	ID; (done flag set)
	IC; (done flag cleared on MAXv)

Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IC	AX – AS	Immediate	
IC	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: CA, GD, ID, II, IN, IP

ID INTERRUPT WHEN DONE

The ID command will set the done flag for the axis or axes to which the command was issued. An interrupt to the host will be generated if interrupts have been enabled in the MAXv. In Windows environments, this interrupt is captured by the device driver which will store the flags to pass on to the host application when requested.

This command allows the MAXv to signal the host when a string of commands has been completed. In AA mode, the done flag register bits will be set as each axis encounters the ID in its command stream, but the done flag in the status register will not be set until all axes have executed the ID command. In the AM mode, only the axes active in the most recent move will set their done flags.

Though move commands are most commonly used with the ID command, others may be used as well. The ID command may be sent to the MAXv to tell it to set done flags whenever the host application could benefit from knowing an event or series of commands has completed.

Example: Interrupt the host CPU after the execution of Move Absolute is finished. When the move is finished the ID command will be encountered in the command queue and will set the done flags.

Enter:
```
AX;
MA100000;
GO;
ID;
```

Response: None

Example: Wait until an input line becomes false then notify the host that this occurred.

Enter:
```
SW2;
ID;
```

Response: None

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
ID	AX – AS	1	0	N/A
ID	AA-AM	1	0	N/A
ID	AA/CD	1	0	1

Related commands: II, IN, IP

II INTERRUPT INDEPENDENT

Like the ID command, the II command tells the MAXv to interrupt the host when each axis finishes a move. Unlike the ID command, only those axes which have been supplied parameters in the most recent move command will get their done flags set and cause interrupts.

Example: The following command sequence would cause interrupts when the Y and T axes finish. If they do not complete at the same time, two separate interrupts would be generated.

Enter:
```
AM;
MR,1000,,10000;
GO;
II;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	AX – AS	Not Valid	
II	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: ID, IN, IP

IN INTERRUPT NEARLY DONE

The IN command allows the MAXv to interrupt the host when the axis or combination of axes is nearly complete. When used in an application involving probing a part after a move, the probes could start accelerating down while the stage is finishing its move, improving the overall system throughput. This command is valid in all modes. The IN command must be entered before the GO or GD command since it is executed before the move is complete. The test is only performed during deceleration. If the IN parameter is greater than the ramp down distance, the interrupt will be generated when the control starts decelerating.

RANGE:
Min Position Range ≤ Parameter(Distance Before End of Move) ≤ MAX Position Range

NOTE: The position range depends on the settings used on MAXv, see specifications for details.

Example: The following sequence would interrupt the host when the X axis is complete and the Z axis is within 10,000 counts of being complete. The Y axis completion would be ignored in this example.

Enter:
```
AA;
IN0,,10000;
MR100000,100000;
GO;
MR,,50000;
GO;
```

Response: None

NOTE: In AA or AM mode, a null value in the argument list specifies the position tolerance for nearly done is not to be set for that axis.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IN#;	AX – AS	1	2
IN#,#,#,#,#,#,#,#;	AA-AM	1	2
-	AA/CD	Not Valid	

Related commands: ID, II, IP

IO I/O BIT DIRECTION SELECTION

This IO# command selects a general purpose I/O to be an input or output as determined by the designator where 1 is an output and 0 is an input.

Range: First parameter is 0-15
Second parameter is 0-1

Example: Select I/O bit 4 as an output and I/O bit 5 as an input.

Enter: AA;
IO4,1;
IO5,0;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IO#,#;	AX – AS	Immediate	
IO#,#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: BD, BH, BL, BX

IP INTERRUPT WHEN IN POSITION

The IP command operates like the ID command except the interrupt is deferred until the stage is within the specified step encoder dead band. If the position hold (HN) is not enabled for an axis, the command will behave like an ID command for that axis.

Example: Set the done flag when axis is within its dead band.

Enter:
```
AX;
HV1000;
HG100;
HD10;
HN;
MR1000;
GO;
IP;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IP	AX – AS	1	0
IP	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: HN, ID, II, IN

IS INTERRUPT ON SLIP

The IS command enables the MAXv to interrupt the host on slip or stall detection if the appropriate bit has been set in the interrupt control register. Slip detection are disabled if a ER, ET, HF, HM, HR, JG, LP, or TM command is entered or if a limit is encountered. If a slip occurs, slip detection must be re-enabled. The factory default slip tolerance (ES) value is 1. This command is intended to be used with stepper motors and not servo motors.

Example: (see ES command on page 5-99)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
IS	AX – AS	1	0
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ES, RL, TN, TF

JF JOG FRACTIONAL VELOCITIES

The JF command will jog one or more axes at the velocities specified, like the JG command. The parameter may include a fractional part allowing better resolution at low speeds. The velocity set by this command will remain the default velocity until altered by a VL, JG or another JF command.

RANGE: -4,000,000.000 ≤ JF ≤ 4,000,000.000

Example: Jog the Y axis at 2 ^{2}/3 steps per second.

Enter: AY;
JF2.667;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
JF#;	AX – AS	3*	0*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 1 to the Command queue
— If the axis is a stepper and encoder or servo axis add 1 to the Command queue
— If the profile move, just prior to this home command was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the home. This adds to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	6
All other Parabolic forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: JG, SA, ST, TM

JG JOG

The JG command is a velocity mode and will move one or more axes at the velocities supplied as parameters. The JG command will accelerate to the programmed velocity at the current AC rate and run until altered by an ST, SA, KL, HF (servo models), another JG command, or a limit switch is encountered while limits are enabled.

The jog velocity may be changed by following the command with another JG command of a different velocity. A change in direction between two JG commands will cause an axis to ramp to a stop then back up in the opposite direction.

This command modifies the move velocity parameter (VL) for the affected axis or axes. The JG command does not require any other command to start the motion.

RANGE: -4,000,000 ≤ JG ≤ 4,000,000

Example: Jog the motor at 100,000 counts per second then change to 35,000 counts per second when the second JG is entered, stay at that velocity for 5 seconds, then stop by decelerating to a stop. Next, jog the motor at 5,000 counts per second in the negative direction.

Enter:
```
JG100000;
JG35000;
WT5000;
ST;
JG-5000;
```

Response: None

Note: Output events waiting for completion of JG will begin when JG is up to its requested velocity. In this case, the motor will ramp from zero to 100,000, ramp back down to 35,000, flatten out at 35,000 for 5 seconds, then ramp to a stop, before moving in the negative direction at a velocity of 5,000.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
JG#;	AX – AS	3*	1*
JG#,#,#,#,#,#,#,#;	AA-AM	3*	1*
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 1 to the Command queue
— If the axis is a stepper encoder or servo axis add 1 to the Command queue
— If the profile move, just prior to this home command was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the home. This add to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue

Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	6
All other Parabolic forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: JF, SA, ST, TM

KA ACCELERATION FEEDFORWARD

KA is the acceleration feedforward gain coefficient used in the PID filter calculations. The factory default value is zero. See the APP command to preserve the KA settings as the power up/reset values. The default value is 0.00

RANGE: 0.00 ≤ KA ≤ 32767.00

Example: Define KA to be 2 on the T axis.

Enter: AT;
KA2;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KA#;	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HF, HN,?KA, KD, KI, KP, KV

?KA REPORT ACCELERATION FEED FORWARD

The ?KA command reports the current setting of the acceleration feed-forward constant (KA) for the current servo axis. Default value is 0.00

Example: Find out what the current KA value is for servo axis Y

Enter: AY;
?KA

Response: =10.50<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KA	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: KA, ?KV

KB PID UPPER BOUND LIMIT COEFFICIENT

The KB command sets the servo axis PID output value upper bound limit from DAC. (This can be useful in debugging servo drives.) The default value is 0.00

RANGE: 0.0 $\le$ KB $\le$ 10.0

Example: Limit the servo output from DAC to 9 max on Y axis. This is approximately half of its normal range.

Enter:
```
AY;
KB9;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KB	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?KB

?KB REPORT AXIS PID UPPER BOUND LIMIT

The ?KB command requests the value of the KB parameters for the upper bound limit to the DAC.

Example: Determine PID upper bound limit for X axis

Enter:
```
AX;
?KB
```

Response: =5.00<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KB	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: DBI, DBN, ?SV

KD DERIVATIVE GAIN COEFFICIENT

KD is the derivative gain coefficient used in the PID filter calculations. See Section 2 for more information regarding this parameter. The factory default value is 20.00. See the APP command to preserve the KD settings as the Power up/Reset values.

RANGE: 0.00 ≤ KD ≤ 32767.00

Example: Set KD to 56 on the Z axis.

Enter: AZ;
KD56;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KD#;	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?KD, KI, KP, HF, HN

?KD REPORT PID DERIVATIVE GAIN

The ?KD command reports the current setting of the derivative gain coefficient (KD) in the PID of the current servo axis.

Example: Forgot to write down the Y axis KD setting which is working well. Report the setting so it can be recorded.

Enter: AY;
?KD

Response: =5.12<LF >

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KD	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: KD, ?KI, ?KP,

KF SET SERVO AXIS PID FRICTION COEFFICIENT

The KF command sets the friction offset coefficient of a servo axis PID. This assists in the smooth start up motion of the X axis, and compensates for friction. The default value is 0.

Range: 0 ≤ KF ≤ 32767

Example: Set the Y axis friction coefficient to 100.

Enter: AY;
KF100;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KF#	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?KF

?KF REPORT SERVO AXIS FRICTION OFFSET

The ?KF command reports the value of the frictional offset coefficient.

Example: Verify Z axis friction offset coefficient is 12

Enter: AZ;
?KF

Response: =12<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KF	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: KD, ?KI, ?KP

KI INTEGRAL GAIN COEFFICIENT

KI is the integral gain coefficient used in the PID filter calculations. See the APP command to preserve the KI settings as the power up/reset values. Default value is 0.04.

RANGE: 0.00 ≤ KI ≤ 32767.00

Example: Define KI to be 3.42 on the X axis.

Enter: AX;
KI3.42;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KI#;	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HF, HN, KD, ?KI, KP

?KI REPORT PID INTEGRAL GAIN

The ?KI command reports the current setting of the integral gain constant (KI) in the PID of the current servo axis.

Example: Report the setting of the KI coefficient on the Z axis

Enter: AZ;
?KI

Response: =1.00<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KI	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?KD, KI, ?KP

KL KILL

The KL command will flush the command queues and terminate pulse generation of all axes immediately. It is intended for emergency termination of any program and to reset the input queues to a known state.

Step motors may not stop immediately even though no more step pulses are delivered due to inertia of the motor and system load. This may result in slippage of the motor. Therefore, the position counter may not accurately reflect the true position of the motor following this command. All axes should be re-homed to return the position counters to a known state.

Due to the encoders used in servo systems, position will not be lost, so re-homing servo axes is unnecessary.

Example: Stop all previously defined movement and flush the queue of a partially entered incorrect move command (you wanted a negative move not a positive one), before GO is entered.

Enter:

```
AX;
MR5000; (oops!)
KL;
MR-5000;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KL	AX – AS	1*	0*
KL	AA-AM	1*	0*
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 1 to the command queue
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: FL, KS, SA, SD, SI, SO, ST,

KM HOME AND KILL

The KM command will move the current axis in the positive direction until home is detected and then kill motion immediately; i.e. without using a deceleration ramp. The position counter will not be reset or cleared. Due to motor and/or payload inertia, the motor may not stop immediately but slip some distance instead. This will result in inaccurate position counters. This command's primary purpose is to move an axis out of the way quickly or to get the axis near home rapidly to speed up the homing process.

Example: Move the Y axis in the positive direction to the home sensor and stop movement as quickly as possible.

Enter: AY;
KM;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KM	AX – AS	1*	1*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 2 to the command queue.
— If an auxiliary output bit settle time has been specified add 2 to the cCommand queue and 1 to the argument queue.
— If the last profile move, just prior to this home command, was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the Home. This will add to the queue requirements under the following conditions:

Axis Ramp Type	Command queue	Argument queue
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	6
All other Parabolic forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: HE, HH, HL, HM, HR, HS, KR, LO, LP

KO OFFSET COEFFICIENT

The KO command defines the offset coefficient to cause the motor to remain stationary and compensate for additional torque on the motor from loading. The factory default value is zero. See the APP command to preserve the KO settings as the power up/reset values.

The factory default value is zero. Full-scale, the KO command has a range of +/- 32,767 which corresponds directly to the 16-bit range of the DAC less a few counts as a buffer zone. Each increment/decrement of the KO value will result in an approximate change in the output voltage of 0.0003 volts.

RANGE: -32767 ≤ KO ≤ 32767

Example: Define the offset coefficient to be –2000 (~ -610mV) on the Y axis.

Enter: AY;
KO-2000;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KO#;	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HF, HN, ?KO

?KO REPORT PID OFFSET

The ?KO command reports the voltage offset (KO) setting for the current servo axis.

Example: The open-loop offset is 218. Make sure the closed-loop offset is the same.

Enter: ?KO

Response: =218<LF >

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KO	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: KO

KP PROPORTIONAL GAIN COEFFICIENT

KP is the proportional gain coefficient used in the PID filter calculations. See the APP command to preserve the KP settings as the power up/reset values. Default value is 10.00.

RANGE: 0.00 ≤ KP ≤ 32767.00

Example: Define KP to be 45.6 on the Z axis.

Enter: AZ;
KP45.6;

Response: None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
KP#;	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HF, HN, KD, KI, ?KP

?KP REPORT PROPORTIONAL GAIN

The ?KP command reports the current setting of the proportional gain coefficient (KP) in the PID of the current servo axis.

Example: Find out what the X axis proportional gain is set to.

Enter: AX;
?KP

Response: =10.00<LF >

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KP	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?KD, ?KI, KP

KR HOME REVERSE AND KILL

The KR command will find home by issuing a JG in reverse. When home is found, it will stop generating pulses immediately; i.e. no deceleration ramp will be generated. This command is identical to the KM command except that the direction of motion is reversed.

Example: Move the Y axis in a negative direction to the home sensor and stop movement as quickly as possible.

Enter: AY;
KR;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KR#;	AX – AS	1*	1*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 2 to the command queue.
— If an auxiliary output bit settle time has been specified add 2 to the command queue and 1 to the argument queue.
— If the last profile move, just prior to this home command, was either a MT or ML move then the axis acceleration and velocity values will be reset to the AC and VL values just prior to the execution of the Home. This will add to the queue requirements under the following conditions:

QUEUE REQUIREMENTS		
Axis Ramp Type	Command queue	Argument queue
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	6
All other Parabolic forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2 x number of ramp segments) +2
S-curve (AJ)	9	104

Related commands: HE, HH, HL, HM, HR, HS, KM, LO, LP

KS KILL SELECTED AXES

This command performs the same operation as the KL (kill) command except that individual axes can be killed without affecting others. KS will flush only the selected axes' command queues rather than the entire board. Refer to the KL command for more details.

Example:	The Y axis has hit a limit switch and is now executing commands that were waiting in the queue. This axis must be reset but the other axes must be allowed to continue operation.

Enter:	AY;
	KS;
	or
	AA;
	KS,1;

Response:	None

NOTE: In AA or AM modes, null values in the argument list specify that motion on that axis is not to be killed.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KS;	AX – AS	1*	0*
KSb,b,b,b,b,b,b,b;	AA-AM	1*	0*
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 1 to the command queue
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: FL, KL, SA, SD, SI, SO, ST

KU PID IINTEGRATION SUM UPPER LIMIT

This command sets servo axis PID integration sum upper limit.

RANGE: 0 < KU < 32767

Example: Set the integration sum upper limit of X axis to 150

Enter: AX;
KU150;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KU#	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?KU

?KU REPORT PID INTEGRATION SUM UPPER LIMIT

Report servo axis PID integration sum upper limit.

Example: Check to make sure the integration sum upper limit of X axis is less than 200.

Enter: AX;
?KU

Response: =150<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KU	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HF, HN, KD, KP, KU

KV VELOCITY FEEDFORWARD

KV is the velocity feedforward coefficient used in the PID filter calculations. The factory default value is zero. See the APP command to preserve the KV settings as the power up/reset values.

RANGE: 0.00 ≤ KV ≤ 32767.00

Example: Set KV to 35.3 on the Y axis.

Enter: AY;
KV35.3;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
KV#;	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HF, HN, KA, KD, KI, KP, ?KV

?KV REPORT VELOCITY FEED FORWARD

The ?KV command reports the current velocity feed forward coefficient (KV) of the current servo axis. Default value is 0.00

Example: Make sure the velocity feed-forward setting of axis T is zero

Enter: AT;
?KV

Response: =0.00<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?KV	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?KA, KV

LA LINEAR RAMP PER AXIS

The LA command specifies that the linear acceleration ramp is to be used by the selected axes. This is the factory default for all axes. See the APP command to preserve the LA settings as the power up/reset values. This command is similar to the PF command but can be used to switch a single axis rather than all axes at once.

Example: Select a linear ramp for the X axis.

Enter: AX;
LA;

Response: None

Example: Select the linear ramp for the Y and T axes.

Enter: AA ;
LA,1,,1;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LA	AX – AS	5	6
LAb,b,b,b,b,b,b,b;	AA-AM	5	6
-	AA/CD	Not Valid	

Related commands: CN, PF, PN, PR, ?RT, SC, SR

LE LOOP END

The LE command terminates the most recent LS command. The axis will loop back and repeat the commands within the loop the number of times specified in the LS command. The loop will start repeating as soon as this command is issued.

Example: Perform a relative move on axis Z 5 times. After each Z move, wiggle the T axis 20 times.

Enter:

```
AZ;
LS5;
MR5000;
GO;
AT;
LS20;
MR50;
GO;
MR-50;
GO;
LE; (terminates the LS20; command)
AZ;
LE; (terminates the LS5; command)
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LE	AX – AS	1	1
LE	AA-AM	1	1
-	AA/CD	Not Valid	

Related commands: LS

LF LIMITS OFF

The LF command disables the limit switches for the addressed axis or axes. This allows the stage to move beyond the limit switches and should be used with caution. The MAXv will still recognize that a limit switch has been closed if the stage runs into one and will report this information via the query commands (See the QA command). However, the limit switch closure will have no effect on motion; i.e. the axis will not be forced to stop as a result.

NOTE: In systems not designed to handle motion beyond the limit switch points, this can potentially cause damage to the system and/or persons operating the system. This command should be used with Extreme caution.

Example:	Set up a board to ignore the Y axis limit switches.

Enter:	AY;
	LF;
	or
	AA;
	LF,1;

Response:	None

Note: In AA or AM modes, a null argument in the parameter list specifies that axis home switch to be affected.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LF	AX – AS	1	0
LFb,b,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: LH, LL, LN, SF SL

LH LIMITS HIGH

The LH command sets the senses of the limit switches on the current axis to active high. The default "true" states of the limits are TTL logic low. This command allows TTL logic high to be treated as the "true" state for applications where this is more convenient. Through the execution of the APP command, limits can be made to default as active high on power-up or reset; see the APP command for details.

Example: Select the limit switch high true condition for the X axis.

Enter: AX;
LH;

Response: None

Example: Select a high true limit condition for the Z and T axes.

Enter: AA;
LH,,1,1;

Response: None

Note: In AA or AM modes, a null argument in the argument list specifies that axis is not to be affected.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LH	AX – AS	1	0
LHb,b,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: LF, LL, LN, ?LS, SF, SL, TL

LL LIMITS LOW

The LL command specifies that over travel occurs when the limit input signal is low (active low). This is the factory default mode. See the APP command for information on how to preserve the LL settings as the power up/reset values.

Example: Select the limit switch low true condition for the X axis.

Enter: AX;
LL;

Response: None

Example: Select a low true limit condition for the Y and T axes.

Enter: AA;
LL,1,,1;

Response: None

Note: In AA or AM modes, a null argument in the argument list specifies that axis is not to be affected.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LL	AX – AS	1	0
LLb,b,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: LF, LH, LN, ?LS, SF, SL, TL

?LM REPORT LIMIT SWITCH SELECTION

The ?LM command reports the limit detection enable mode of an axis.

Example: Query MAXv to learn if limit detection is enabled for the Z axis.

Enter: AZ;
?LM

Response: =On<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?LM	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: LF, LN

LN LIMITS ON

The LN command restores the operation of the limit switches for the addressed axis or axes. This is the default mode at power up or reset. With limit switches enabled, if the axis encounters a limit switch in the course of executing any motion, the axis will be instructed to stop and the host will be notified of the event. Limit conditions are treated as critical errors and should not be used as simple positioning inputs to the MAXv.

Example: Set up the Y and T axes to stop immediately when a limit switch is encountered.

Enter:
```
AA;
LN,1,,1;
or
AY;
LN;
AT;
LN;
```

Response: None

Note: In AA or AM modes, a null argument in the parameter list specifies that axis is not to be affected.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LN	AX – AS	1	0
LNb,b,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: LF, LH, LL, SF, SL

LO LOAD MOTOR POSITION

The LO command sets the motor position independently of the encoder position unlike LP which sets both to the same supplied value. The LP command will override the LO command and reset the motor position. If the LP command is used and a different motor position value than the encoder position is desired, the LO command must be reentered. Any valid position within the allowable position range may be used.

Example: Set the motor position to 50,000 and the encoder position to 100,000 on the T axis

Enter: AT;
LP100000;
LO50000;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LO#;	AX – AS	1	1
LO#,#,#,#,#,#,#,#;	AA-AM	1	1
-	AA/CD	Not Valid	

NOTE: For AA and AM modes that any values in the argument list specify that axis is not to be affected.

Related commands: LP, RE, RM, RP, RU

LP LOAD POSITION

The LP command will immediately load the number supplied as a parameter in the absolute position registers of the axis. In models with the encoder option, the parameter will be loaded into the encoder position register and the parameter times the encoder ratio will be loaded into the position counter. If no parameter is supplied, the value of zero is used.

The LO command can be used after this command to set the motor position independently of the encoder position.

RANGE: Min. Position Value ≤ LP ≤ Max. Position Value

NOTE: The position value range varies with the specific settings used on MAXv.

Example: The following would load the X axis position register with 1000, and the Z axis position register with 2000.

Enter:
```
AA;
LP1000,,2000;
```

Response: None

Example: The following would load the Y axis position register with 20,000 and the encoder position register with 30,000 counts, in encoder models.

Enter:
```
AY;
ER3,2;
LP30000;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LP#;	AX – AS	1*	1*
LP#,#,#,#,#,#,#,#;	AA-AM	1*	1*
-	AA/CD	Not Valid	

— If the axis is a stepper and encoder or servo axis add 1 to the command queue and 1 to the argument queue.

Related commands: LO, RE, RM, RP, RU

LS LOOP START

The LS command sets the loop counter for the axis being programmed in the single axis mode and all axes in the AA mode. The command expects a loop counter operand following the command. The commands up to the LE loop terminator will be executed the number of times specified by the operand. Loops may be nested up to four levels deep on each axis. The parameter must be less than 2,147,483,647.

The first loop of commands will occur immediately as they are entered. The remaining loops will be executed after the loop terminator (LE) has been entered.

The axis mode (e.g. AX, AY, and AA) must be the same when entering and exiting the loop, otherwise the matching loop termination command will not be found by the board's command processor. The axis mode may be switched within the loop provided the board is in the same mode when the LE command is sent as when the LS command was sent.

If you want one axis to wait for another in the loop, you must be in the AA mode throughout the loop. If you are in the single axis mode in the loop, each axis' commands will go into their separate queues and execute independently of each other.

If, when entering a looping sequence of commands, the command queue is filled before the LE loop terminator is entered, the board will hang. This is because there is no space for the LE command. When programming a loop of more than four or five moves, the queue size should be examined with the RQ command to see if it is nearing zero.

RANGE: 1 ≤ LS ≤ 2,147,483,647

Example: Execute a 100,000 count relative move on the Z axis 5 times.

Enter:
```
AZ;
LS5;
MR100000;
GO;
LE;
```

Response: None

NOTE: The first move will occur immediately after entering the GO command. The remaining 4 moves will be executed after the loop terminator LE has been entered.

Example: Execute a 100,000 count move relative on the X axis together with a 100 count move on the T axis, followed by a move absolute to 100 counts on the X axis and 200 counts on the T axis, four times.

Enter:
```
AA;
LS4;
MR100000,,,100;
GO;
MA100,,,200;
GO;
LE;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
LS#;	AX – AS	1	1
LS#;	AA-AM	1	1
-	AA/CD	Not Valid	

Related commands: LE, WH, WS

?LS REPORT LIMIT ACTIVE STATE

The ?LS command reports the active state of the limits for the current axis. The LL and LH commands are used to set this value.

Example: Find out whether the Y axis limits are active high or active low.

Enter: AY;
?LS

Response: =l<LF> or =h<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?LS	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: LH, LL

MA MOVE ABSOLUTE

The MA command will set up one or more axes to move to the absolute positions supplied as parameters. In AA mode, an axis may remain stationary by entering a comma but omitting the parameter. The move is actually initiated by a GO or GD command.

In AA mode all axes begin their move at the same time. Each axis will use its predefined acceleration and velocity values to move to the new absolute position. Each axis may or may not get to the destination at the same time because each axis utilizes individual velocities and accelerations. The MT command will ensure all axes reach their target positions simultaneously.

Example:	In the single axis mode, move the X axis to absolute position 100,000 counts with the previously entered acceleration and velocity parameters.

Enter:	AX;
	MA100000;
	GO;

Response:	None
Example:	In the AA mode, move the Y axis to absolute position 10,000 counts and the T axis to absolute position 1,000 counts. The other axes will remain in their current positions.

Enter:	AA;
	MA,10000,,1000;
	GO;

Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MA#;	AX – AS	1*	1*
MA#,#,#,#,#,#,#,#;	AA-AM	1*	1*
-	AA/CD	Not Valid	

— If the axis is a stepper encoder or servo axis add 1 to both the command and argument queues.
— If the axis is using the cosine acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 1 to the command queue and add 10 to the argument queue.
— If the axis is using the S-curve acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 3 to the Command queue and add 41 to the argument queue.
— If the acceleration and velocity values need to be reset to their AC and VL values because the last move just prior to this move was either a MT or ML move, add the following queue requirements:

QUEUE REQUIREMENTS		
Axis Ramp Type	COMMAND QUEUE	ARGUMENT QUEUE
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	8
All other Parabolic Forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

Related commands: GD, GN, GO, GS, GU, ML, MR, MT

MD TEMPORARY MACRO DEFINE

MD is used to begin defining a temporary macro. A macro can contain up to 250 characters. Macros 0 through 4 are temporary and they will be erased when the controller is reset or power is turned off. Macros 5 through 24 are stored in non-volatile memory and will be preserved when the controller is reset or powered off. This command cannot be used to define a macro directly into numbers 5 through 24. They must be defined with this command and then moved into the non-volatile space with the PT command.

Enter the macro number immediately after the MD command. The macro number must be between 0 and 4. Next enter the command string, which is made up of up to 250 ASCII characters. After entering the command string for the macro, enter a control Z to end the macro definition. The control Z may be ASCII value 26 or the string “^Z” (carat, shift 6 on a US keyboard, followed by a Z.)

Be careful not to exceed 250 ASCII characters or the size of the axis queue when working with macros.

RANGE: 0 ≤ MD ≤ 4

Example: Define macro 2 to set velocities to 20000 on all axes of a two axes board.

Enter:
```
MD2;
AA;
VL20000,20000;
^Z
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MD#;	AX – AS	Immediate	
MD#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: MX, PM

ML MOVE LINEAR

The ML command uses linear interpolation to perform a straight line relative move. Input parameters are relative distances for each axis involved in the move. The ML command should be followed by a GO or GD to start the axes together. The velocity and acceleration parameters are scaled to allow the axes to move and finish together. All axes are scaled to the axis with the longest move time (the master axis). At the end of the move, all involved velocities and accelerations will be restored to their pre-move values.

Example: In the AA mode, move the Y, Z and T axes 10000, 100 and 1000 counts respectively with all axes starting and finishing together. The other axes remain in their previous positions.

Enter:
AA;
ML,10000,100,1000;
GO;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	AX – AS	Not Valid	
ML#,#,#,#,#,#,#,#;	AA-AM	2*	2*
-	AA/CD	Not Valid	

— If this is the master axis for this move and its acceleration and velocity values need to be reset to their original AC and VL values, because a previous move altered them, add the following queue requirements:

— If this is not the master axis, then the acceleration and velocity values will be modified, and the following queue requirements will be added:

QUEUE REQUIREMENTS		
Axis Ramp Type	COMMAND QUEUE	ARGUMENT QUEUE
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	8
All other Parabolic Forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

QUEUE REQUIREMENTS		
Axis Ramp Type	COMMAND QUEUE	ARGUMENT QUEUE
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	8
All other Parabolic Forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

— If the axis is using the cosine acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 1 to the command queue and 10 to the argument queue:

— If the axis is using the S-curve acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 3 to the command queue and add 41 to the argument queue.

Related commands: GD, GN, GO, MA, MR, MT

MM MOVE MINUS

The MM command sets the direction logic to move in the negative direction. The direction output for the current axis will change (if necessary) to reflect the new direction. All non-direction-specific move commands will now move in the negative direction.

Example: Set the direction line to move in the negative direction on the Y axis.

Enter: AY;
MM;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MM	AX – AS	1	0
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: FP, MO, MP, MV, SP

MO MOVE ONE PULSE

The MO command will output one count in the current direction (do not use the GO command). The direction may be reversed directly by use of the MM or MP command or indirectly via a move such as JG or MR. This command generates the output signal in one sample interval and thus eliminates the latency of generating a ramp with an MR1; GO command sequence.

Example: Move the Z axis one pulse in the negative direction.

Enter:
AZ;
MM;
MO;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MO	AX – AS	1	0
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: MM, MP

MP MOVE POSITIVE

The MP command sets the direction logic to move in the positive direction. The direction output for the current axis will change (if necessary) to reflect the new direction. All subsequent non-direction-specific motion commands will now move in the positive direction.

Example: (see MV command page 5-171)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MP	AX – AS	1	0
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: FP, MM, MO, MV, SP

MR MOVE RELATIVE

The MR command will set up one or more axes to move relative from their current positions at the time the move is executed. In the AA mode, an axis may remain stationary by entering a comma but omitting the parameter. The move is actually initiated by a GO or GD command.

In AA mode all axes will start at the same time. Each axis will use its predefined acceleration and velocity values to move to the new position. Each axis may, or may not, get to the destination at the same time, because each axis utilizes individual velocities and accelerations. To ensure all axes reach their destinations simultaneously use the ML command.

Example: In the single axis mode, move the X axis 2468 steps in the negative direction.

Enter:
```
AX;
MR-2468;
GO;
```

Response: None

Example: In the AA mode, move the X axis 12345 steps in the positive direction and the Y axis 6789 steps in the positive direction. Both axes will start at the same time.

Enter:
```
AA;
MR12345,6789;
GO;
```

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MR#;	AX – AS	1*	1*
MR#,#,#,#,#,#,#,#;	AA-AM	1*	1*
-	AA/CD	Not Valid	

— If the axis is a stepper encoder or servo axis add 1 to both the command and argument queues.
— If the axis is using the cosine acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 1 to the command queue and add 10 to the argument queue.
— If the axis is using the S-curve acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 3 to the command queue and add 41 to the argument queue.
— If the acceleration and velocity values need to be reset to their AC and VL values because the last move just prior to this move was either a MT or ML move, add the following queue requirements:

QUEUE REQUIREMENTS		
Axis Ramp Type	COMMAND QUEUE	ARGUMENT QUEUE
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	8
All other Parabolic Forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

Related commands: GO, GD, GN, GS, GU, MA, ML, MT

MT MOVE TO

The MT command uses linear interpolation to move two or more axes to the specified absolute positions. The syntax is similar to the ML command. This command is invalid if loops are being used due to the overhead involved. The command will become valid again after executing an ST or KL command. When used in the contour definition mode, only the axes being used in the contour must be provided for in the MT syntax. A GO or GD command initiates the move.

The axis that will reach its destination first (the master axis) is used as a gauge to modify the acceleration and velocity values of the other axes. This is done to ensure all involved axes arrive at their targets simultaneously. At the end of the move, any velocity or acceleration value that was modified is restored to its pre-move value.

Example: In the AA mode, move the X, Y and T axes to absolute positions 1000, 10000 and 100 counts, respectively, with each starting and finishing together. The unused axis remains in its previous position.

Enter:
```
AA;
MT1000,10000,,100;
GO;
```

Response: None.

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
-	AX – AS	Not Valid		
MT#,#,#,#,#,#,#,#;	AA-AM	2*	2*	N/A
MT#,#,#,#,#,#,#,#;	AA/CD	2*	2*	4 + (number of axes included in contour definition

— If this is not the master axis, then the acceleration and velocity values will be modified, and the following queue requirements will be added:

QUEUE REQUIREMENTS		
Axis Ramp Type	COMMAND QUEUE	ARGUMENT QUEUE
Linear (PF, LA)	4	4
Short Form Parabolic (PN0, PR0)	4	8
All other Parabolic Forms (PN, PR)	4	22
Cosine (CN, SC)	4	22
Custom (SR)	4	(2*number of ramp segments) + 2
S-curve (AJ)	9	104

— If the axis is using the cosine acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 1 to the command queue and 10 to the argument queue:

— If the axis is using the S-curve acceleration ramp, and velocity needs to be adjusted so the profile is not truncated, add 3 to the command queue and add 41 to the argument queue.

Related commands: GO, GD, GN, MA, ML, MR

MV MOVE VELOCITY

The MV command causes the current axis to move to a new absolute position (parameter 1) at a new velocity (parameter 2). When the destination is reached control will be passed to the next command which should be another MV command or a SP command. If the command is not received in time the controller will continue to move at the specified velocity. Note that this is a slave mode and it is the responsibility of the user to provide the commands in time. They may be queued ahead of time. If a new MV command is sent after the controller has already passed the destination specified in the command, the controller will continue to move at the old velocity.

The MAXv will ramp up or down as needed at the rate previously set with the AC command and travel at the new velocity until the new position is reached. The controller will not reverse direction if the position has already passed, but will behave as explained above. Thus the direction of the move must be specified before starting the move with the MP or MM commands. All destinations must be in absolute position; no position relative moves are allowed due to the nature of these commands. Cosine and parabolic acceleration will not apply.

RANGE:
Min Position Range ≤ Parameter 1 (Absolute Position)≤ Max Position Range
1 ≤ Parameter 2 (Velocity)≤ 4,000,000

NOTE: The position range depends on the settings used on MAXv, see specifications for more details.

Example:	Generate a velocity staircase with the breakpoints given in absolute position. Default acceleration (AC) of 200,000

Enter:	MP;
	MV10000,30000;
	MV20000,50000;
	MV30000,10000;
	SP35000;

Response:	None

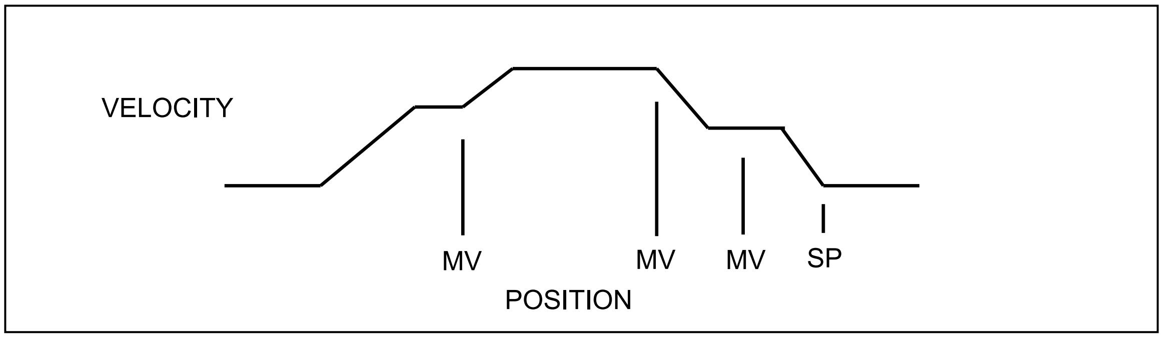


Figure 5-3 Velocity Staircase Profile

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MV#,#;	AX – AS	2*	2*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 1 to the command queue
— If this axis is a stepper encoder or servo axis add 2 to the command queue

Related commands: FP, MM, MP, SP

MX MACRO EXECUTE

The MX command will execute the command string stored in the specified macro. The macro number that is entered as the argument of the command must be between 0 and 24.

RANGE: 0 ≤ MX ≤ 24

Example:	Execute macro number 6.
Enter:	MX6;
Response:	None

NOTE: MX itself is an immediate command. However, the commands contained within the macro may have queue requirements.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
MX#;	AX – AS	Immediate	
MX#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: MD, PM, PT

NV NEW CONTOUR VELOCITY

The NV command will set a new velocity for a contour currently in execution. When the velocity changes, the MAXv will switch to the new velocity at the start of the next command in the contour queue without ramping. An NV command issued without a contour currently executing will have no effect.

RANGE: 1 ≤ NV ≤ 4,000,000

Example:	The contour is executing at 25,000 counts per second. Change the velocity to 30,000 counts per second upon execution of the next command in the contour queue.
Enter:	NV30000;
Response:	None

QUEUE REQUIREMENTS				
FORMAT	MODE	COMMAND	ARGUMENT	CONTOUR
-	AX – AS	Not Valid		
-	AA-AM	Not Valid		
NV#;	AA/CD	Immediate		

Related commands: CD, CV

PA POWER AUTOMATIC

The PA command will turn on or off the auxiliary outputs at the beginning of each GO or GD command execution and complement the outputs after the move is executed. The auxiliary will be turned on; i.e. pulled high, upon the execution of the GO or GD and off at the end of that move if the parameter is zero or not specified in the single axis mode. If the parameter is non-zero, the sense is reversed; i.e. the auxiliary output is turned off (driven low) upon the execution of the GO or GD command and on at the end of the move.

The SE command can be used to apply a settling time at the end of each move before complementing the auxiliary bit. This is useful for systems that need to retain torque for some specific amount of time before allowing the motor drive to reduce current output.

This mode need only be set once and can be turned off by using the AN, AF, PH, or PL commands. Axes can be selectively affected in the AA mode. The values of the included parameters set the state of the auxiliary line during the move. The following queue requirements apply to each GO or GD command in the command stream in the AA and single axis modes. This mode is off by factory default. See the APP command to preserve the PA settings as the Power up/Reset values.

Example: Turn on the Y axis auxiliary output at the beginning of a move and turn the T axis output off at the beginning of a move, while in the AA command mode. (note the reversed logic; i.e. 0 = on, 1 = off. "on" pulls the signal line to ground. "off" lets it rise to 5 volts or its pull-up reference voltage.)

Enter: AA;
PA,0,,1;

Response: None

NOTE: PA selects the mode immediately but places entries in the axis command queue to set the state of the aux bit to the idle state.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PAb;	AX – AS	1*	0
PAb,b,b,b;	AA-AM	1*	0
-	AA/CD	Not Valid	

***If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.**

Related commands: ADH, ADL, AF, AN, ?PA, SE

?PA REPORT POWER AUTOMATIC STATE

The ?PA command reports whether the current axis has power automatic mode enabled. The PA command is used to set this value. If power automatic mode is enabled, it will report whether the Aux bit goes high or low during a move. The axis is stationary if the aux bit is low.

Example:	Determine if X axis power automatic is high or low.
Enter:	AX; ?PA
Response:	=0<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?PA	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

*Power automatic mode is enabled for X axis such that the aux bit is high during a move.

Related commands: PA, ?SE

PE REPORT ENCODER POSITIONS

The PE command reports the encoder positions of all encoder and/or servo axes. All encoder positions will be reported even in single axis mode. (This is the same as AA;RE;.)

Example: Report the encoder positions of an eight axis servo board.

Enter: PE

Response: 0,50,156,0,0,1506,0,0<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PE	AX – AS	Immediate	
PE	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: PP, RE, RP

PF LINEAR ON

The PF command restores all axes to linear acceleration and deceleration ramps. This command should not be given while an axis is in motion or the results may not be predictable. This command affects all axes even if issued in the single axis mode. PF is the factory default setting. See the APP command to restore the PF setting as the power up/reset mode.

Example: Turn off cosine or parabolic ramps, returning to linear.

Enter: PF;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PF	AX – AS	5	6
PF	AA-AM	5	6
-	AA/CD	Not Valid	

Related commands: CN, LA, PN, PR, ?RT, SC, SR

PH POWER HIGH

The PH command will turn the currently active axis auxiliary port bit ‘on’ (high). This command disables power automatic (PA) mode.

Example: Set aux bit of axis T to high.

Enter: AT;
PH;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PHb;	AX – AS	1	0
PHb,b,b,b,b,b,b,b	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: PA, ?PA, PL, ?SE

PL POWER LOW

The PL command will turn the currently active axis auxiliary port bit ‘off’ (low). This command disables power automatic (PA) mode.

Example: Set aux bit of axis T to low.

Enter: AT;
PL;

Response: None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PLb;	AX – AS	1	
PLb,b,b,b,b,b,b,b	AA-AM	1	
-	AA/CD	Not Valid	

Related commands: PA, ?PA, PH, ?SE

PM PRINT MACRO

The PM command will return the command string stored in the specified macro number as a command response. The macro number entered as the argument for this command must be between 0 and 24.

RANGE: 0 ≤ PM ≤ 24

Example: Print the command string contained in macro 19.

Enter: PM19;

Response: If a macro string is defined for macro 19, the macro string will be the response. If no macro is defined there will be no response.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
PM#;	AX – AS	Immediate	
PM#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

NOTE: Macros are stored as ASCII character strings. If <LF> character is used as command terminator it will be sent back to the host computer by the PM command. If the application software stops reading a character string first it will appear that the PM command did not return the macro contents. To avoid this issue, save macros without <LF> terminator. Use semi-colons instead to terminate commands

Related commands: MD, MX, PT

?PM REPORT HOLD STATE

The ?PM command reports whether the PID for the current servo axis is enabled.

Example: A limit switch was hit by servo axis Y. See if the PID is still enabled for that axis.

Enter: AY;
?PM

Response: =off<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?PM	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: HF, HN

PN PARABOLIC ON

The PN command sets all axes to truncated parabolic ramps. This acceleration profile starts at 100% of the programmed acceleration and decreases in steps of 10% of the initial acceleration down to as low as 10%. The parameter supplied selects the number of steps. It must be in the range of 3 to 10 corresponding to 70% and 10% acceleration at the peak respectively. A parameter out of this range or no parameter supplied defaults to 70% or 3 steps. Note that the parameter is the number of steps, not the acceleration values. The larger number is a lower acceleration at the peak. This command should not be given while an axis is in motion or the results may not be predictable. This command affects all axes even if issued in the single axis mode. The PF command is used to return to the default linear motion profiles. See the APP command to preserve the PN setting as the Power up/Reset ramp.
See PR for single axis.

RANGE: 3 ≤ PN ≤ 10

Example:	Set the board to be in the smoothest parabolic acceleration ramp.
Enter:	PN10;
Response:	None

QUEUE REQUIREMENTS

FORMAT	MODE	AXIS RAMP TYPE	COMMAND	ARGUMENT
PN#;	AX – AS	Short Form Parabolic (PN0)	5	10
PN#;	AX – AS	All Other Parabolic (PN,PO)	5	24
PN#,#,#,#,#,#,#,#;	AA-AM	Short Form Parabolic (PN0)	5	10
PN#,#,#,#,#,#,#,#;	AA-AM	All Other Parabolic (PN,PO)	5	24
-	AA/CD	Not Valid		

Related commands: CN, LA, PF, PR, ?RT, SC, SR

PP REPORT MOTOR POSITIONS

The PP command reports the motor positions of all axes in ASCII format. All axes will be reported even in single axis mode. (This is the same as AA; RP)

Example: Report the motor positions of an eight axis controller.

Enter: PP

Response: 0,0,0,125,0,200,0,565<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PP	AX – AS	Immediate	
PP	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: PE, RE, RP

PR PARABOLIC RAMP PER AXIS

The PR command defines parabolic acceleration/deceleration ramps for use with one or more axes. This command is similar to the PN command except that only the specified axes are affected. The APP command can be used to store the settings of PR as the power-up/reset defaults.

RANGE: 3 ≤ PR ≤ 10

Example: Select a 10 step parabolic ramp for the T axis.

Enter: AT;
PR10;

Response: None

Example: Select a 10 step parabolic ramp for the Y axis and an 8 step parabolic ramp for the T axis and the R axis.

Enter: AA;
PR,10,,8,,,8;

Response: None

QUEUE REQUIREMENTS				
FORMAT	MODE	AXIS RAMP TYPE	COMMAND	ARGUMENT
PR#;	AX – AS	Short Form Parabolic (PN 0;)	5	10
PR#;	AX – AS	All Other Parabolic (PN,PR)	5	24
PR#,#,#,#,#,#,#,#;	AA-AM	Short Form Parabolic (PN 0)	5	10
PR#,#,#,#,#,#,#,#;	AA-AM	All Other Parabolic (PN,PO)	5	24
-	AA/CD	Not Valid		

Related commands: CN, LA, PF, PN, ?RT, SC, SR

?PS REPORT MOTOR TYPE

The ?PS command reports the type(s) of the active axes. In single axis mode it reports the motor type of the current axis. In AA mode it reports the types of all motors. The indicator reported is interpreted as follows:

O = Stepper Only
E = Stepper with Encoder
M = Servo

?PS is an immediate command and is active in single axis and AA/AM mode.

Example: Report the motor types of all motors.

Enter: AA;
?PS

Response: M,M,M,M,O,O,E,E<LF>

This means axes X, Y, Z and T are servo motors, axes U and V are open-loop stepper motors, and axes R and S are stepper motors with encoders.

Example: Report the motor type of X axis

Enter: AX;
?PS

Response: =M<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGURMENT
?PS	AX – AS	Immediate	
?PS	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: PSE, PSO, PSM

PSE CONFIGURE CURRENT AXIS AS A STEPPER AXIS WITH ENCODER

The PSE command specifies that the motor type for this axis is a stepper motor with encoder feedback. To preserve this setting after MAXv is powered down, use APP to archive this setting in flash memory.

Example: Set up the X axis to be a stepper with encoder feedback.

Enter: AX;
PSE;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PSE	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?PS, PSM, PSO

PSM CONFIGURE CURRENT AXIS AS A SERVO AXIS

The PSM command specifies that the motor type for this axis is a servo motor. To preserve this setting after the MAXv is powered down, use APP to archive this setting in flash memory.

Example: Set up the Y axis to run a servo motor.

Enter: AY;
PSM;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PSM	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?PS, PSE, PSO

PSO CONFIGURE CURRENT AXIS AS AN OPEN LOOP STEPPER

The PSO command specifies that the motor type for this axis is a stepper motor with no encoder feedback (open-loop). To preserve this setting after MAXv is powered down, use APP to archive this setting in flash memory.

Example:	Set up axis Z as an open-loop stepper.

Enter:	AZ;
	PSO;

Response:	None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
PSO	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?PS, PSE, PSM

PT PRESERVE A TEMPORARY MACRO

Use PT to save a temporary macro permanently by copying it to non-volatile memory. The temporary macro number, which is entered as an argument for this command, must be between 0 and 4. The non-volatile macro number, which is also entered as an argument for this command, must be between 5 and 24.

RANGE:
0 ≤ Parameter 1 ≤ 4
5 ≤ Parameter 2 ≤ 24

Example:	Copy temporary macro 3 to non-volatile macro 19.
Enter:	PT3,19;
Response:	None

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**Min**	**Max**
PT#,#;	AX – AT	Immediate	
PT#,#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: MD, MX, PM

QA QUERY AXIS STATUS

The QA command returns the status of the single addressed axis like the RA command except the limit and done flags will not be cleared. Refer to the RA command for details.

Example: Check the status of the X axis.

Enter: AX;
QA;

Response: PNNH<LF>

Refer to the table "Character Meaning" in the RA command on page 5-188.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
QA	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: EA, QI, RA, RI

QI QUERY INTERRUPT STATUS

The QI command returns the same information as the QA command but for all axes at once. The 4 character fields for each axis are separated by commas. The state of the status flags for all axes with out clearing the controller's copy of the done flags.

Example: Check the status of an eight axis board.

Enter: AA;
QI

Response: PNNN,MNNN,PDNN,MNLN,PDNN,MNNN,MNNN,MNNN<LF>

Refer to the table "Character Meaning" in the RA command on page 5-188.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	AX – AS	Not Valid	
QI	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: EA, QA, RA, RI

QL QUERY ALL LIMIT SENSORS

QL is an immediate command and is valid in single axes, AA, and AM modes of operation.

The response of the QL command is a hexadecimal value representing the state of the response of the positive and negative limit sensors of each axis. A limit sensor in the TTL "High" state will have a value of ONE (1) and a limit sensor in a TLL "Low" state will have value of Zero (0). The order of the limit sensors is as follows:

8000	S Axis Positive Limit	0080	S Axis Negative Limit
4000	R Axis Positive Limit	0040	R Axis Negative Limit
2000	V Axis Positive Limit	0020	V Axis Negative Limit
1000	U Axis Positive Limit	0010	U Axis Negative Limit
0800	T Axis Positive Limit	0008	T Axis Negative Limit
0400	Z Axis Positive Limit	0004	Z Axis Negative Limit
0200	Y Axis Positive Limit	0002	Y Axis Negative Limit
0100	X Axis Positive Limit	0001	X Axis Negative Limit

Example: Query all limit sensors

Enter: AA;
QL;

Response: A500<LF> (This means that the S, V, Z, and X positive limit sensors are high and the R, U, T, and Y positive limit sensors are low. All negative limit sensors are low.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
QL	AX – AS	Immediate	
-	AA-AM	Immediate	
-	AA/CD	Not Valid	

RA REQUEST AXIS STATUS

The RA command returns the state of the limit and home switches and the done and direction flags for the currently addressed axis. The done flag register will be reset by this command.
See RI command for single axis mode.

The status is returned in the following format:

CHARACTER MEANING		
CHAR	SENT	DESCRIPTION
1	P	Moving in positive direction
	M	Moving in negative direction
2	D	Done (ID, II or IN command has been executed, set to N by this command or IC command)
	N	No ID executed yet
3	L	Axis in overtravel. Char 4 tells which direction. Set to N when limit switch is not active.
	N	Not in overtravel in this direction
4	H	Home switch active. Set to N when home switch is not active.
	N	Home switch not active
5	LF	Line feed

Example: The Y axis just encountered a limit, verify its status.

Enter: AY;
RA

Response: PNLN<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RA	AX – AT	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: EA, QA, QI, RI

RB REPORT BIT DIRECTION

The RB command reports the bit direction of the general I/O bits.

Example: Report direction of all I/O's. In this example I/O bits 0-3 are set as outputs and I/O bits 4-15 are set as inputs.

Enter: RB

Response: 000F<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	AX – AS	Immediate	
RB	AA-AM	Immediate	
-	AA/CD	Not Valid	

RC REQUEST ACCELERATION

The RC command will return the current acceleration rate of the current axis. This may differ from the programmed acceleration if a cosine (CN) or parabolic (PN) ramp is being generated. When the stage is stopped, the parameter returned will be zero (0). When the stage is running at programmed speed; i.e. not accelerating, the parameter returned will be zero (0). While a contour is executing, the value computed to generate the appropriate lead in will be returned. The response to the RC command is surrounded by linefeed + carriage return pairs.

Example: Display current acceleration values for all axes on an eight axis board.

Enter: AA;
RC;

Response: 2000000, 2000000, 2000000, 2000000, 2000000, 2000000, 2000000, 2000000<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RC	AX – AS	Immediate	
RC	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: ?AC, AC, RV

RDB RESTORE DEFAULT PARAMETER IN BACK-UP

The RDB command restores the parameters from the “backup” archive in flash memory.

Example: Restore the currently assigned parameters from the back up archive (See Section 5.4.2, Power-Up Defaults for list).

Enter: RDB;

Response: None.

Note: This command places entries in all axis command queues to set up the motion profile parameters.

QUEUE REQUIREMENTS				
FORMAT	MODE	AXIS RAMP TYPE	COMMAND	ARGUMENT
RDB	AX – AS	Linear Acceleration	5	6
RDB	AA-AM	Linear Acceleration	5	6
RDB	AX-AS	Short Parabolic Acceleration	5	10
RDB	AA-AM	Short Parabolic Acceleration	5	10
RDB	AX-AS	Standard Parabolic Acceleration	5	24
RDB	AA-AM	Standard Parabolic Acceleration	5	24
RDB	AX-AS	Cosine Acceleration	5	24
RDB	AA-AM	Cosine Acceleration	5	24
RDB	AX-AS	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDB	AA-AM	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDB	AX-AS	S-curve Acceleration	10	107
RDB	AA-AM	S-curve Acceleration	10	107
-	AA/CD	Not Valid		

Related commands: APB, APP, RDF, RDP

RDF RESTORE FACTORY DEFAULT VALUES

RDF restores the current parameter set to the factory default values. This process does not alter the parameters stored in flash via the APP command. To restore the flash memory to factory default, the RDF command must be issued followed by the APP command.

Example:	Assign the current parameter set to be the factory default values.
Enter:	RDF;
Response:	None.

QUEUE REQUIREMENTS			
FORMAT	MODE	Min	Max
RDF	AX – AT	Immediate	
RDF	AA-AM	Immediate	
-	AA/CD	Not Valid	

FACTORY DEFAULT SETTINGS
Motor type is open-loop stepper (?PSO)
Axis direction bit output state is normal(DBN)
Over travel limits are active low(LL)
Home inputs are active low(HL)
I/O direction (BDFF00)
Soft limits are disabled(SF)
Acceleration is 2000000 (AC2000000)
Use linear acceleration ramps(LA)
Early deceleration factor is zero
Output Reset State (BR11111111--------)
Velocity is 200000 (VL200000)
Base Velocities zero(VB0)
Analog Input zero offset (AO0,0)
User units are off(UF)
Auxiliary output bits settle times are zero(SE0)
Auxiliary output bits power control is disabled(AN)
Auxiliary output bit power up states are high(ADH)
Backlash Compensation (BL0)
Software over travel limits are disabled (TL0,0;)

ENCODER MODELS
Motor/encoder ratio is one to one(ER1,1)
Encoder slip tolerance is one(ES1)
Position maintenance dead band is zero(HD0)
Position maintenance velocity limit is zero(HV0)
Position maintenance hold gain is one(HG1)
Encoder Home pattern (EH101)

SERVO MODELS
PID output is bipolar(BI)
PID output voltage is normal(SVN)
PID proportional gain is 10 (KP10.00)
PID differential gain is 20 (KD20.00)
PID integrator gain is 0.04 (KI0.04)
PID acceleration feed forward is zero(KA0.00)
PID velocity feed forward is zero(KV0.00)
PID offset is zero(KO0)

Related commands: APB, APP, RDB, RDP

RDP RESTORE POWER-UP DEFAULT VALUES

The RDP command restores the motion parameters using power-up defaults.

Example: Restore the power-up default parameters from flash memory.

Enter: RDP;

Response: None.

NOTE: This command places entries in all axis command queues to set up the motion profile parameters.

QUEUE REQUIREMENTS				
FORMAT	MODE	AXIS RAMP TYPE	COMMAND	ARGUMENT
RDP	AX –AS	Linear Acceleration	5	6
RDP	AA-AM	Linear Acceleration	5	6
RDP	AX-AS	Short Parabolic Acceleration	5	10
RDP	AA-AM	Short Parabolic Acceleration	5	10
RDP	AX-AS	Standard Parabolic Acceleration	5	24
RDP	AA-AM	Standard Parabolic Acceleration	5	24
RDP	AX-AS	Cosine Acceleration	5	24
RDP	AA-AM	Cosine Acceleration	5	24
RDP	AX-AS	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDP	AA-AM	Custom Acceleration	5	5 + (2 * number of ramp segments)
RDP	AX-AS	S-curve Acceleration	10	107
RDP	AA-AM	S-curve Acceleration	10	107
-	AA/CD	Not Valid		

Related commands: APB, APP, RDB, RDF

RE REPORT ENCODER POSITION

The RE command returns the current encoder position of the currently addressed axis or axes in encoder counts.

Example: Examine the current encoder position of the Y axis.

Enter: AY;
RE;

Response: 12345<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RE	AX – AS	Immediate	
RE	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: PE, PP, RP

RI REQUEST INTERRUPT STATUS

The RI command is a multi-axis mode command that returns the same status information for all axes as the RA command does in single axis mode. The 4 character fields for each axis are separated by commas. The done flag is reset by this command as it would be via the RA command.

Example: Check the status of an 8 axis board.

Enter: AA;
RI;

Response: MDNN,MDNN,PNLN,PNNN,PNLN,PNNN,PNNN,PNNN,<LF>

Refer to the table "Character Meaning" in the RA command on page 5-188.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	AX – AS	Not Valid	
RI	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: EA, QA, QI, RA

RL REPORT SLIP STATUS

The RL command returns the slip detection status of all axes. S is returned if a slip condition has occurred for that axis, or else an N is returned. The number of characters returned corresponds to the number of axes available on the board. This command is intended to be used with stepper motors with encoders and not with servo motors. Open-loop stepper always returns "n" and servo axes always returns "N" in their RL response.

Example:	On an eight axis board, see if any axis has slipped.
Enter:	RL
Response:	NNSNNNNN<LF> (The Z axis has slipped.)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RL	AX – AS	Immediate	
RL	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: ES, IS, TF, TN

RM REMAINDER

The RM command divides the position counter by the parameter supplied and replaces the position counter with the resulting remainder. The parameter must be greater than zero and less than 65,000. This command is used in applications where the controller is managing the motion of a continuously rotating object. It allows the position counter to keep track of the absolute position without regard to the number of revolutions it may have rotated. This command has the same effect on the encoder position register on axes with the encoder feedback enabled.

RANGE: 0 < RM < 65,000

Example: The current position of a rotating stage with a full-revolution count of 6000 is needed. Since this stage has been rotated several times without regard for the position, the position counter has reached 163,279. Send an RM6000 command to find out what the real position of the axis is.

Enter: (Current position is 163,279)
RM6000;
(Current position is now 1,279)

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RM#;	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: LO, LP, RE, RP, RU

RP REQUEST POSITION

The RP command returns the current position of the currently addressed axis in single axis mode or all positions separated by commas in AA or AM mode. The position will be returned to the host in ASCII format. This command is not queued; i.e. the current position will be returned immediately even if the axis is in motion.

Example:	The current position on the Y axis is 12345. Use the RP command to verify the position.
Enter:	AY; RP
Response:	12345<LF>

Example:	Verify the positions of all axes.
Enter:	AA; RP
Response:	100,200,300,400,500,600,700,800<LF>

QUEUE REQUIREMENTS

FORMAT	MODE	COMMAND	ARGUMENT
RP	AX – AS	Immediate	
RP	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: RE, PE, PP

RQ REPORT CONTOUR QUEUE SIZE

The RQ command reports the number of available entries in the contour queue. This command is only valid while in the contour definition mode.

MAX Value = 32763

Example: Report the command queue space remaining in contour mode.

Enter: RQ;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	**MODE**	**COMMAND**	**ARGUMENT**
RQ	AX – AS	Not Valid	
RQ	AA-AM	Not Valid	
RQ	AA/CD	Immediate	

RQA REPORT AXIS ARGUMENT QUEUE

The RQA axis argument queue command returns the number of available entries in the axis argument queue.

MAX Value = 12799

Example: Report the command argument queue space remaining for the Z axis.

Enter: AZ;
RQA

Response: 10225<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RQA	AX – AS	Immediate	
RQA	AA-AM	Immediate	
-	AA/CD	Not Valid	

RQC REPORT AXIS COMMAND QUEUE

The RQC command returns the number of available entries in the axis command queue.

Max Value = 2559

Example: Report the command queue space remaining for all axes (on a four axes controller).

Enter: AA;
RQC

Response: 2559,2559,2559,2559<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RQC	AX – AS	Immediate	
RQC	AA-AM	Immediate	
-	AA/CD	Not Valid	

?RT REPORT RAMP TYPE

The ?RT command reports the current acceleration ramp assigned to the active axis. Possible responses are:

LA	Default linear ramp
PRn	Parabolic where n specifies number of segments
SC	Cosine ramp
SRn	Custom ramp where n specifies the table number
PR	Parabolic ramps
SSn	S-curve ramp where n specifies the S-curve profile number

Example: Make sure custom ramp #3 was assigned to the Y axis

Enter: AY;
?RT

Response: =SR3<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?RT	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: CN, LA, PF, PN, PR, SC, SR, SS

RU REPORT POSITION IN USER UNITS

The RU command returns the current position in user units (see UU command on page 5-228). The format of the response is a floating-point number.

Example: One revolution of a motor is 2000 steps. Define user units so moves can be referenced in revolutions. Move the Z axis 3 1/2 revolutions. Use RU to display the position when the move is complete.

Enter:

```
AZ;
UU2000;
LP0;
MR3.5;
GO;      (Wait until move is complete.)
RU
```

Response:

```
3.50000<LF>
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RU	AX – AS	Immediate	
RU	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: RE, RP, UU

RV REQUEST VELOCITY

The RV command will return the current velocity at which the axis is moving. This may differ from the programmed maximum velocity if the axis is accelerating or decelerate. If the JF command is executing, the command only reports the integer part of the velocity.

Example: Jog the Y axis at 12345 steps per second.
Display the current velocity.

Enter:
```
AY;
JG12345;
RV
```

Response: 12345<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
RV	AX – AS	Immediate	
RV	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: RC, ?VL, VL

SA STOP ALL

The SA command flushes all queues and causes all axes to decelerate to a stop at the rate previously specified in an AC command. All status and position information is retained. Even when executed in a single axis mode, this command will cause all axes to stop.

Example: Send all axes on a move, then ramp them to a stop before they finish.

Enter:
```
AA;

VL100,100,100,100,100,100,100,100;
MR1000,2000,3000,4000,3000,2000,1000,4000;
GO (wait awhile)
SA;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SA	AX – AS	1*	0*
SA	AA-AM	1*	0*
-	AA/CD	Not Valid	

Related commands: KL, KS, SD, SI, SO, ST

SC COSINE RAMP PER AXIS

The SC command specifies that the standard cosine acceleration ramp is to be used by the selected axis/axes. This command is similar to the CN command except that only the selected axis or axes are affected. The APP command will store the settings of SC as the power-up/reset defaults.

Example: Select the cosine ramp for the X axis.

Enter: AX;
SC;

Response: None.

Example: Select the cosine ramp for the Y, T, and R axes.

Enter: AA;
SC,1,,1,,,1;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SC	AX – AS	5	24
SCb,b,b,b,b,b,b,b;	AA-AM	5	24
-	AA/CD	Not Valid	

Related commands: CN, LA, PF, PN, PR, ?RT, SR

SD STOP AND RESET DONE

The SD command may be substituted for the SA command. It will reset the done flags for all axes, stop all axes at the rates previously specified via the AC command, and then flush all axis command queues. This allows the host to be interrupted when all axes have stopped by using the ID command after the SD. The SA ID combination may flag the completion early if one of the axes is already done from a previously executed ID.

Example: Stop all axes and reset all done flags. When all axes have stopped set all done flags.

Enter:
```
AA;
SD;
ID;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SD	AX – AS	2*	0*
SD	AA-AM	2*	0*
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 1 to the command queue.
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

Related commands: KL, KS, SA, SI, SO, ST

SE SETTLING TIME

The SE command allows specification of a settling time, in milliseconds, to be used before the auxiliary output is complemented when using PA mode. The parameter may be any value up to 1000 milliseconds. Specification of a parameter of zero disables SE mode.

The factory default settling time is zero. See the APP command to preserve the SE settings as the Power up/Reset values.

RANGE: $0 \leq SE \leq 1000$

Example: Turn on the Z axis auxiliary output upon execution of a move and have it remain on for 500 milliseconds after the move is complete.

Enter:
```
AZ;
PA;
SE500;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SE#;	AX – AS	Immediate	
SE#,#,#,#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: AF, AN, PA, ?SE

?SE REPORT SETTLING TIME

The ?SE command reports the settling time setting (SE) used with power automatic mode (PA) for the current axis.

Example: Report the current settling time for axis X

Enter:
```
AX;
?SE
```

Response: =250<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?SE	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?PA, SE

SF SOFT LIMITS OFF

The SF command restores the default operation of the limit switches; i.e. causes the affected axis or axes to abruptly halt when a limit switch is encountered. If soft limits have been made the power-up default, the APP command must be used to store hard limit operation as the default.

Example: Set up a board to make the X axis stop immediately when a limit is encountered.

Enter: AX;
SF;

Response: None.

Example: Set up a board to make the Y and T axes to stop immediately when a limit is encountered.

Enter: AA;
SF,1,,1;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SF	AX – AS	1	0
SFb,b,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: LF, LH, LL, LN, ?SL, SL, TL

SI STOP INDIVIDUAL

This command can be used to stop only certain axes. In a single axis mode, the SI command behaves identically to ST. In a multi-axis mode, however, SI can be used to stop any number of axes and can be used in place of SA. Like SA, SI will ramp those axes to be stopped using the rate previously specified via the AC command. This command is useful for stopping a specific axis when the current axis mode is unknown and for stopping several axes without affecting current motion on other axes.

Each parameter represents an axis from X through S. Any non-zero value in a parameter will cause the corresponding axis to be stopped.

Example: Start a motion on all four axes. When input bit 1 becomes true, stop axes Y and T without affecting X and Z.

Enter:

```
AM;
MR15000,30000,20000,40000;
GO;
SW1;
SI,1,,1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SI	AX – AS	1*	0*
SIb,b,b,b,b,b,b,b;	AA-AM	1*	0*
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 1 to the command queue.
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

Related commands: KL, KS, SA, SD, SO, ST

?SK REPORT AXIS SLIP KILL MODE SELECTION

The ?SK command reports whether slip kill mode is currently enabled for the active axis.

Example: Find out whether encoder slip kill mode is enabled for axis Z

Enter: AZ;
?SK

Response: =on<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?SK	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: TF, TN

SL SOFT LIMITS ON

The SL command enables the MAXv to ramp an axis to a stop rather than abruptly killing the motion when a limit switch is encountered on that axis. The output queue is not flushed except for the current move. This mode is effective for point to point and JG moves only. Soft limits can be made the power-up default via the APP command.

Example: Set up a board to allow the X axis to ramp to a stop when a limit is encountered.

Enter:
```
AX;
SL;
```

Response: None.

Example: Set up a board I/O to allow the Y and T axes to ramp to a stop when a limit is encountered.

Enter:
```
AA;
SL,1,,1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SL	AX – AS	1	0
SLb,b,b,b;	AA-AM	1	0
-	AA/CD	Not Valid	

Related commands: LF, LH, LL, LN, SF, ?SL, TL

?SL REPORT SOFT LIMIT STATUS

The ?SL command reports whether soft limits are currently enabled for the active axis.

Example: Find out whether soft limits are enabled for axis Z

Enter: AZ;
?SL

Response: =on<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?SL	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: SF, SL

SO STOP AT POSITION BY RAMPIING FROM DISTANCE

The SO command instructs the MAXv to continue moving until reaching a specified distance (parameter 2) from a specified stop point (parameter 1). The axis will then ramp to a stop within the specified distance. This allows the user to control the point at which deceleration begins, the rate of deceleration, and the stop point, all with a single command.

RANGE:

Min. Position Range ≤ Parameter 1 (Stop Position) ≤ Max. Position Range
Min. Position Range ≤ Parameter 2 (Distance from Stop Position to Start Decelerating) ≤ Max. Position Range

NOTE: The position range is dependent on the MAXv's settings, see Specifications for more detail.

Example: The X axis is jogging at 10,000 steps per second. We want the axis to stop at position 50,000 but it must not start ramping until reaching position 46,000.

Enter: SO50000,4000;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SO#,#;	AX – AS	3*	2*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

— If the axis is a stepper and encoder or servo axis, add 1 to the command queue
— If PA (power automatic) mode is active, add 1 to the command queue
— If an AUX bit settling time has been specified, add 2 to the command queue and add 1 to the argument queue

Related commands: KL, KS, SA, SD, SI, ST

?SO REPORT ANALOG OUTPUT MODE

The ?SO command reports whether the analog output type for the current servo axis is bipolar or unipolar. The possible responses are BI and UN, the same commands used to set one mode or the other.

Example: The Y axis should be setup with unipolar outputs. Use ?SO to make sure.

Enter: AY;
?SO

Response: =un<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?SO	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: BI, UN

SP STOP AT POSITION

The SP command will cause the controller to attempt to stop at the specified destination. If there is insufficient distance to stop at the previously specified deceleration when the command is received, the controller will stop as soon as possible at that deceleration. This command is not compatible with the JG command.

Example: (see MV command on page 5-171)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SP#;	AX – AS	2*	1*
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

— If the axis is a stepper and encoder or servo axis add 1 to the command queue
— If PA (power automatic) mode is active add 1 to the command queue
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue

Related commands: FP, MM, MP, MV

SR SELECT CUSTOM RAMP

The SR command selects a previously defined custom ramp profile for use with a currently active axis. This command will override previous ramp type selection for the given axis such as PN and CN.

RANGE: 1 ≤ SR ≤ 8

Example: Select custom ramp number 4 for use with axis Y.

Enter:
```
AY;
SR4;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SR#;	AX – AS	15	5 + (2* number of segments in ramp definition)
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: CN, DAB, DAE DAR, ED, LA, PF, PN, PR, ?RT, SC

SS SELECT CUSTOM S-CURVE PROFILE

The SS command selects a custom S-curve profile to use for an axis or axes.

RANGE: 1 ≤ Ramp Number ≤ 8

Example: Selects custom S-curve profile #1 for Y axis.

Enter:
```
AY;
SS1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SS#:	AX – AS	10	107
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?AJ, AJ, ?RT

ST STOP

The ST command flushes the queue for the current axis or axes only and causes the axis/axes to decelerate to a stop at the rate previously specified via the AC command. This command is used to stop one or more motors in a controlled manner from jog mode or an unfinished GO or GD command. This command is executed immediately upon receipt. All status and position information is retained. When executed in a multi-axis mode, the ST command is equivalent to the SA command.

Example: Move the Y axis for a while at 1200 steps/second and then ramp to a stop.

Enter:
```
AY;
JG1200;
(Wait awhile)
ST;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
ST	AX – AS	1*	0*
ST	AA-AM	9*	61*
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 1 to the command queue.
— If an auxiliary output bit settle time has been specified add 2 to the command queue and add 1 to the argument queue.

Related commands: KL, KS, SA, SD, SI, SO

?SV REPORT SERVO VOLTAGE INVERSION STATE

The ?SV command reports the current logical direction for the current servo axis. The state is set with the SVI and SVN commands. The possible responses to the ?SV command are =n for normal and =i for inverted.

Example: Report whether servo voltage is positive for positive moves on axis X

Enter: AX;
?SV

Response: =n <LF> (voltage is normal; i.e. positive for positive moves)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?SV	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?DB, SVI, SVN

SVI INVERT SERVO VOLTAGE

The SVI command inverts the voltage output for the current axis. After receiving this command, the MAXv will produce a negative voltage for positive motion and a positive voltage for negative motion. To cancel this command, issue an SVN command. To make inverted servo outputs the default at power up or reset, use the APP command.

Example: The Y axis encoder is counting opposite the expected direction. Setup the Y axis to produce a negative voltage when moving positive instead of a positive voltage to correct the problem.

Enter: AY;
SVI;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SVI	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: BI, DBI, DBN, ?SV, SVN, UN

SVN NORMALIZE SERVO VOLTAGE

The SVN command normalizes the voltage output for the current axis, negating the effects of the SVI command. After receiving this command, the MAXv will produce a positive voltage for positive motion and a negative voltage for negative motion, the default behavior. To make this the default behavior (if it has been changed via SVI/APP), use the APP command. (SVN is the factory default setting.)

Example: The Y axis encoder was rewired and now counts in the correct direction. Return the Y axis servo output to normal; i.e. output positive voltage for positive motion.

Enter: AY;
SVN;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SVN	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: BI, DBI, DBN, ?SV, SVI, UN

SW SYNC WAIT

The SW command allows synchronization of multi-axis moves or other tasks on one or more MAXv boards by using one of the general purpose input lines. This command causes the MAXv to stop processing new commands until the general purpose input line has been released (allowed to go high) before proceeding with the next command.

The SW command can also be used to cause an axis to wait until the others are finished. To do this, wire-OR the auxiliary lines from several axes together and connect them to a general purpose input line. Use power automatic (PA) mode with the SW command on that line. All commands after that will wait until all axes release their auxiliary lines; i.e. come to a complete stop.

RANGE: 0 ≤ Bit Number ≤ 15

NOTE: The parameter used to specify Bit Number must be configured as an input. See BD, RB, and IO.

Example: The following command sequence will cause the X axis move to wait until the Y axis has finished its move and turned off its auxiliary output which has been wired to the general purpose input 0 line.

Enter:

```
AY;
AN;
MR2000;
GO;
AF;
AX;
SW0;
MR10000;
GO;
```

Response: None.

The SW command provides a way to synchronize moves on two or more controllers. The following example shows one way to do this.

Example: You have 3 four axis controllers, for a total of 12 axes to move together. Call board 1 the “master” and boards 2 and 3 the “slaves”. Wire board 1’s X axis auxiliary line to the two slave boards’ general purpose input 0 line. Send to the master the command “AX PA0”, setting the master’s X axis auxiliary line low until its move starts. This also sets the slaves’ general purpose input 0 line low. Enter the “SW0” command to the two slaves, followed by the move and GO commands. On the master, enter the move command, followed by the GO command. When the master’s move starts, the PA command will set the auxiliary line high releasing the wait on the slave boards. All three boards will start their moves. This provides synchronization to within 480us of each board.

Procedure: Wire board 1’s X axis auxiliary line to board 2’s and board 3’s general purpose input 0 line.

Enter: (Board 1) AX;

```
(Board 1) AX;
          PA0;
(Board 2) AA;
          SW0;
          MR200,200,200,200;
          GO;
(Board 3) AA;
          SW0;
          MR300,300,300,300;
          GO;
(Board 1) AA;
          MR100,100,100,100;
          GO;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
SW#;	AX – AS	1	1
SW#;	AA-AM	1	1
-	AA/CD	Not Valid	

Related commands: BW, WA, WQ, WT

TF TURN OFF SLIP KILL MODE

The TF command disables slip kill mode (enabled with TN.)

Example: Slip kill mode is enabled but a move needs to be performed where slip is likely and not important for this move. Disable slip kill mode.

Enter: TF;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
TF	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ES, IS, RL TN

TL SET SOFTWARE OVERTRAVEL LIMITS

The TL command sets logical limits on the range of travel for an axis. Two parameters must be supplied; one for the upper travel limit and the other for the lower travel limit, both as absolute positions. If the axis reaches either of these logical limits, the MAXv will flag a limit condition just as it would be using the physical limit switch inputs. Move Absolute (MA), Move Relative (MR), and Jog (JF,JG) commands are subject to software travel limits because the MAXv checks an internal absolute position register.

Note: Software overtravel limits and physical limit switch inputs can be enabled at the same time.

RANGE: $\underline{+}$ Position Range

Note: The position range on MAXv is dependent on the adder rate and update rate selection. See Specification section for more detail.

Example: Set logical position limits for the X axis of +/-1,000,000.

Enter: AX;
TL1000000,-1000000;

Response: None.

Note: TL0,0; Turns software limits off.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
TL#,#;	AX – AS	1	2
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: LF, LL, LH, LN, SF, SL, ?TL

?TL REPORT SOFTWARE OVER TRAVEL LIMITS

The ?TL command reports the software travel limits for the current axis assigned via the TL command. The first value returned is the upper (or "positive") limit and the second value is the lower (or "negative") limit. These are not physical limits but rather positional limits that an axis should not exceed. If one of these limits is exceeded, the MAXv will set the current axis' limit flag and notify the host computer of the condition as though the axis encountered a hard limit.

Example:	Find out what the software limits of the Y axis are currently set to.
Enter:	AY; ?TL
Response:	=101000,-1000<LF>
Example:	Find out what the software limits of the T axis are currently set to.
Enter:	AT; ?TL
Response:	=0,0;<LF> (software limits for axis T are currently disabled)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?TL	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: TL

TM TIMED JOG

The TM command performs a jog at the current velocity limits defined for the axis/axes for the specified number of milliseconds. In multi-axis mode, all axes begin moving at the same time and ramp to a stop when their respective jog times have elapsed. The overall jog time will be the parameter passed to the TM command plus deceleration time and acceleration time.

RANGE: 0 ≤ TM ≤ 200,000

Example:	Jog the X axis for 1000 milliseconds.

Enter:	AX;
	TM1000;

Response:	None.
Example:	Jog the X axis for 1000 milliseconds and the Z axis for 2000 milliseconds, starting both at the same time.

Enter:	AA;
	TM1000,,2000;

Response:	None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
TM#;	AX – AS	5*	1*
TM#,#,#,#,#,#,#,#;	AA-AM	5*	1*
-	AA/CD	Not Valid	

— If PA (power automatic) mode is active add 1 to the command queue
— If the axis is a stepper and encoder or servo axis add 1 to the command queue

Related commands: JF, JG, SA, ST

TN TURN ON SLIP KILL MODE

The TN command enables slip kill mode. In this mode, if the motor slips beyond the dead band set by the ES command, the MAXv will kill motion on the axis that slipped as though a KL command had been issued to the axis. This mode can be disabled (default) with the TF command.

Example: X axis is sent on a move. Its encoder cable was not connected to the controller (oops!). The controller issues a KL (Kill) command to the X axis after receiving the slip interrupt. The slip interrupt is generated once the difference between the motor position counts and encoder counts exceed 20.

Enter:

```
AX;
ES20;
TN;
IS:
LP0;
MA30;
GO;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
TN	AX – AS	1	1
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ES, IS, RL, TF

TX TRACK THE X AXIS

Set up the current axis so that it tracks the X axis. The command argument is the tracking ratio specified as a floating point.
*NOTE: If this is a negative, then the axis moves in the opposite direction of the X axis. The command is invalid for the X axis.

Example:	Set the Z axis to track the X axis.

Enter:	AZ;
	TX;

Response:	None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
TX	AY – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related Commands: ET, HF

UF USER UNITS OFF

The UF command turns off user units returning all numeric commands and responses to their factory default raw representations. This command is equivalent to and preferred over UU1; since it turns off the mode thus minimizing unnecessary overhead. See the APP and APB commands to preserve the UF settings to flash memory.

Example: Turn off user unit conversion on the X, Y, and Z axes.

Enter:

```
AX;
UF;
AY;
UF;
AZ;
UF;
```

Or

```
AA;
UF1,1,1;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
UF	AX – AS	Immediate	
UFb,b,b,b;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: ?UU, UU

UN UNIPOLAR

The UN command sets the analog torque outputs of servo axes to unipolar. The analog output will range between 0.0VDC and +10VDC when unipolar is enabled. At maximum positive velocity, the board outputs +10VDC. At maximum negative velocity, the board output approaches 0.0VDC. To maintain position the board outputs 5VDC. This command is valid only in single axis mode.

Example: Set up servo axis X for unipolar operation.

Enter: AX;
UN;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
UN	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: BI, DBI, DBN, ?SO, SVI, SVN

?UR REPORT THE CONTROLLER'S MOTOR UPDATE RATE

This command causes the controller to report its motor control update rate.

Example: Requests the controller's motor update rate.

Enter: ?UR

Response: 1024<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?UR	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: #UR

#UR SET UPDATE RATE

This command sets the update rate of the controller.

#UR#;
# specifies the number of updates per second.

RANGE: 1024, 2048, 4096, 8192

Any other value besides those specified in the range will result in a command error.

Example:	Set the update rate to 2048 times per second.
Enter:	#UR2048;
Response:	None

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
#UR#;	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: ?UR

UU USER UNITS

The UU command converts all move velocities, distances, etc. to user specified units by multiplying by the specified parameter. The UF command is used to terminate this mode. Factory default is with this command off. See the APP or APB commands on to preserve the UU settings to flash memory.

Example: The motor, driver, and gear ratio you are using requires 10,000 steps to move one inch. Set up the X, Y, and Z axes so you can enter move information in inches.

Enter:
```
AX;
UU10000;
AY;
UU10000;
AZ;
UU10000;
Or
AA;
UU10000,10000,10000;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
UU#;	AX – AS	Immediate	
UU#,#,#,#;	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: RU, UF, ?UU

NOTE: While the user units' mode provides a certain level of convenience to the user, it does so at a cost, namely accuracy and control. User unit conversions may cause round off error and may possibly truncate key information.

?UU REPORT AXIS USER UNIT

This command returns the current user units' multiplier as set via the UU command.

Example: Make sure the UU512 command we sent earlier is still current. The command will return the UU value with six digits to the right of the decimal point. If the UU value exceeds six digits for the fractional value, the value will be rounded off to the sixth decimal place.

Enter: ?UU

Response: =512.000000<LF>

If user units are turned off (UF) ?UU returns:

=off<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?UU	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: UF, UU

VB VELOCITY BASE

The VB command allows the acceleration ramp to start off at a specified velocity. This allows faster acceleration and the ability to pass through resonance quickly in some applications. The velocity jumps instantly to the specified velocity, and then accelerates as usual. The deceleration is the same in reverse. This mode is active only for linear ramps; it is ignored for cosine and parabolic ramps but not flagged as a command error. The parameter must be greater than zero and less than the programmed velocity, where the factory default is zero steps per second. This command is not valid with the JG command nor will it work in conjunction with the DC command. See the APP command to preserve the VB settings as the power-up/reset values.

If the VL command is used after the VB command and the velocity value set with VL is less than the previously set VB value, the initial velocity used at the start of a move will be the VL value minus one. This will result in a one-step acceleration ramp and must be taken into consideration in applications making use of the VB command.

RANGE: 0 ≤ VB < VL value

Example: In the single axis mode, set the Y axis velocity base to 200.

Enter: AY;
VB200;

Response: None.

Example: In the AA mode, set the X and Y axes velocity bases to 200.

Enter: AA;
VB200,200;

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
VB#;	AX – AS	1	1
VB#,#,#,#,#,#,#,#;	AA-AM	1	1
-	AA/CD	Not Valid	

Related commands: AC, DC, ?VB, VL

?VB REPORT BASE VELOCITY

The ?VB command returns the base (starting) velocity setting for the current axis as set by the VB command. Note that the base velocity must be lower than the command velocity.

Example: The acceleration ramp should start at 0pps. Make sure we didn't leave it at some other value.

Enter: ?VB

Response: =1500<LF> (Oops! We forgot to set it back to zero)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?VB	AX – AS	Immediate	
	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: VB, ?VL

VL VELOCITY

The VL command sets the maximum velocity register of one or more axes to the operands which follow the command. The factory default is 200,000 steps per second. See the APP command to preserve the VL settings as the power-up/reset values.

RANGE: 1 ≤ VL ≤ 4,000,000

Example: In single axis mode, set the X axis velocity to 10,000 counts per second.

Enter: AX;
VL10000;

Response: None.

Example: In the AA mode, set the peak velocities of the X and T axes to 5,000 and 50,000 respectively. Leave the other axes with their previous values.

Enter: AA;
VL5000,,,50000;

Response: None.

QUEUE REQUIREMENTS

FORMAT	MODE	Axis Ramp Type	COMMAND	ARGUMENT
VL#; or VL#,#,#,#,#,#,#,#;	AX-AS or AA-AM	Linear (LA or PF)	1	1
VL#; or VL#,#,#,#,#,#,#,#;	AX-AS or AA-AM	Short Parabolic (PN0; or PR0:)	2*	26*
VL#; or VL#,#,#,#,#,#,#,#;	AX-AS or AA-AM	All other Parabolic Forms	Not Valid	
VL#; or VL#,#,#,#,#,#,#,#;	AX-AS or AA-AM	Cosine (CN or SC)		
VL#; or VL#,#,#,#,#,#,#,#;	AX-AS or AA-AM	Custom Ramps (SR)		
VL#; or VL#,#,#,#,#,#,#,#;	AX-AS or AA-AM	S-curve (SS)		

Related commands: AC, DC, VB, ?VB, ?VL

?VL REPORT PEAK VELOCITY SETTING

The ?VL command returns the peak velocity setting for the current axis as set by the VL command.

Example:	Make sure our "AX;VL50000;" command worked.
Enter:	?VL
Response:	=150000<LF>

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
?VL	AX – AS	Immediate	
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: VB, ?VB, VL

VS VELOCITY STREAMING

The VS command will generate a pulse train without acceleration or deceleration at the rates specified using the X and Y axes. The parameters are time in 1 update rate second sample intervals, X velocity, and Y velocity. This is a slave mode and cannot be mixed or queued with other commands; i.e. no other motions involving the X and Y axes or AA mode may be currently in execution when this command is issued. AX mode is the only valid mode for use with this command. The VS command does not require a GO command to start the motion; motion begins immediately upon receipt of the complete VS command.

RANGES:
1 ≤ Parameter 1 (Time in 1 update rate of a second) ≤ 200,000
0 ≤ Parameter 2 (X Axis Velocity) ≤ 4,000,000
0 ≤ Parameter 3 (Y Axis Velocity) ≤ 4,000,000

Example: Create a stair step profile on the X and Y axes, with the X axis moving in the negative direction and the Y axis in the positive direction. Make each step last 1 second long and increase velocity by 1,000 steps/second, until a velocity of 3,000 steps/second is reached, then step down to 0 steps/second. (Example assumes an update rate of 1024.)

Enter:
```
AX;
VS1024,-1000,1000;
VS1024,-2000,2000;
VS1024,-3000,3000;
VS1024,-2000,2000;
VS1024,-1000,1000;
VS1,0,0;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
VS#,#,#,#,#,#,#,#;	AX	2	3
-	AA-AM	Not Valid	
-	AA/CD	Not Valid	

Related commands: MV

WA WAIT FOR AXES

The WA command, valid only in AA mode, allows a command to wait until all moves on all axes are finished before executing any further commands. Some commands which can affect a non-moving axis, such as AN, AF and PA, may execute before a previous move on other axes has finished, especially while in a looping (LS-LE, WH-WG) mode. By preceding these commands with a WA, they will not execute until all previously defined moves have finished.

Example: The Z axis auxiliary line controls a laser beam that you only want on while the Z axis moves in a positive direction. The X and Y axes position the laser. You want to repeat the action 10 times.

Enter:

```
AA;
VL1000,1000,1000;
AC10000,10000,10000;
LS10;
MR1000,1000;
GO;
WA;
AN,,1;
MR,,500;
GO;
AF,,1;
MR,,-500;
GO;
LE;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
-	AX – AT	Not Valid	
WA	AA-AM	2*	0*
-	AA/CD	Not Valid	

* This command places entries in all axes' queues.

Related commands: SW, BW, WQ, WT

WD WHILE END

The WD command serves as the loop terminator for the WS command.

Example: (see WS command on page 5-240)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
WD	AX – AS	1	1
WD	AA-AM	1*	1*
-	AA/CD	Not Valid	

* In AA or AM mode, entries are made in all axes' queues.

Related commands: WS

WG WHILE FLAG

The WG command serves as the terminator for the WH command.

Example: (see WH command page 5-237)

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
WG	AX – AS	1	1
WG	AA-AM	2*	1*
-	AA/CD	Not Valid	

- *In AA or AM mode, entries are made in all axes' queues.

Related commands: CW, WH

WH WHILE

The WH command will execute all commands between it and the terminating WG command as a loop until terminated by a CW command. This allows repeated execution of a command sequence which can be terminated by the host. These commands may not be nested but may be executed sequentially.

Example: You have a 3 axis platform that you use to drill holes in the center of a ¼ inch thick sheet of metal. The sheet is 6 inch square. The driver / motor / lead - screw pitch provide 10000 steps per inch. The operator must manually insert and remove the square from the platform. The X and Y axes move a drill into the desired position. The Z axis lifts and lowers the drill. The operator presses a switch which tells the motion controller that the square is in place and ready to be drilled. The operator will continuously remove and replace the squares until ready to take a break. The following is a description of how to set up an OMS board to perform this task.

Procedure: Connect a normally closed momentary switch between user I/O input line 0 and ground. This will be the “Ready to Drill” switch.

Enter:

```
AX;
UU10000;*set up user units so we can reference move to inches
AY;
UU10000;*10000 steps = 1 inch
AZ;
UU10000;              *10000 steps = 1 inch
AX;
VL.1;
AC10;                 *set up X axis homing velocity and acceleration
AY;
VL.1;
AC10;                 *set up Y axis homing velocity and acceleration
AZ;
VL.1;
AC10;                 *set up Z axis homing velocity and acceleration
AX;
HR;
AY;
HR;
AZ;
HR;                   *send each axis to home
AA;
VL3,3,.5;             *set normal move velocity for X, Y and Z axes
WH;                   *start of loop to drill squares indefinitely
SW0;                  *(operator removes/replaces square into platform)
MA3,3;                *wait until operator presses switch
GO;                   *move to center of square
MA,,.5;
GO;                   *move the drill through the square (1/2 inch
                        move on the Z axis drill through the square)
MA,,0;
```

GO;	*lift the drill
MA0,0;	
GO;	*move the platform to home position
WG;	*loop back to starting WH command
(CW;)	*operator wants a break so he/she sends CW from keyboard and presses switch once more (since loop will most likely be waiting for the switch at this point)
	*the loop ends and the following commands execute
MA0,0,0;	
GO;	*move to home position

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
WH	AX – AS	1	1
WH	AA-AM	2*	1*
-	AA/CD	Not Valid	

- In AA or AM mode, entries are made in all axes' queues.
- Related commands: CW, LS, WG, WS

WQ WAIT FOR QUEUE TO EMPTY

The WQ command is a special command that stops the board from processing any new commands until the command queue for the current axis mode is empty; i.e. all previous moves have finished. This command is not valid in looping (LS-LE, WH-WG) modes.

Example: Move the Y axis 1,000 steps and wait until the move is complete before asking for the position.

Enter:

```
AY;
MR1000;
GO;
WQ
RP;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
WQ	AX – AS	Immediate	
WQ	AA-AM	Immediate	
-	AA/CD	Not Valid	

Related commands: BW, SW, WA, WT

WS WHILE SYNC

The WS command will execute the commands between the WS and WD commands as a loop while the specified general purpose input line is true; i.e. low (default). When the line goes high the MAXv will exit the loop and execute the commands which follow the WD command. The test is at the bottom of the loop; i.e. the commands between WS and WD will always be executed at least once.

WD#[,*state*];
If the optional second parameter, *state*, is entered then it specifies the input condition to continue the loop. If the *state* parameter is zero, the loop will continue while the input is low. If the *state* parameter is non-zero, the loop will continue while the input is high. If the *state* parameter is not entered, the default behavior is to loop while the input is low.

If the input line specified is already in the specified state to exit the loop when the WS/WD loop is issued to the MAXv, those commands will be executed only once.

#RANGE: 0 ≤ Parameter ≤ 15

Example: Execute a continuous loop, moving the X axis 10,000 counts and then move the Y axis -1000 counts, until an external device terminates the loop by setting general purpose input 1 high.

Enter:
```
AA;
WS1;
MR10000;
GO;
MR,-1000;
GO;
WD;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
WS#;	AX – AS	1	1
WS#;	AA-AM	1*	1*
-	AA/CD	Not Valid	

* In AA or AM mode, entries are made in all axes' queues

Related commands: LS, WD, WH

WT WAIT

The WT command will wait for a specified number of milliseconds before proceeding with the next command in the queue. In the AA mode, all axes will wait and entries are made in all axis queues. Immediate commands will not wait since they are not queued.

RANGE: 1 ≤ WT ≤ 200,000

Example: You want to produce pulses on the X axis at 5,000 steps/second for 2 seconds, then 10,000 pulses/second for 3 seconds, and then stop.

Enter:

```
AX;
JG5000;
WT2000;
JG10000;
WT3000;
ST;
```

Response: None.

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
WT#;	AX – AS	2	1
WT#;	AA-AM	2	1
-	AA/CD	Not Valid	

Related commands: BW, SW, WA, WQ

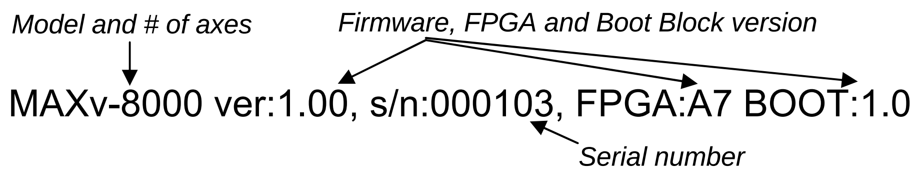
WY WHO ARE YOU

The WY command returns the model type, firmware revision number, and number of controlled axes of the board being addressed.

Example: You want to examine the board identification string.

Enter: WY

Response:



```
MAXv-8000 ver:1.00, s/n:000103, FPGA:A7 BOOT:1.0
```

QUEUE REQUIREMENTS			
FORMAT	MODE	COMMAND	ARGUMENT
WY	AX – AS	Immediate	
WY	AA-AM	Immediate	
-	AA/CD	Not Valid	

This page is intentionally left blank

6. HOST SOFTWARE

6.1 INTRODUCTION TO MAXv SOFTWARE SUPPORT

No software is provided with the MAXv motion controller. The VME user must write their own drivers or adapt their previous VME drivers to the MAXv interface (see Chapter 3 and 4) for samples of flow charts and specific data flow diagrams and details.

This page is intentionally left blank

7. SERVICE

7.1 USER SERVICE

The MAXv family of controllers contains no user serviceable parts. All jumper pins are located on J7, J8, J12, and J13 (See Chapter 2.2 for more details).

7.2 THEORY OF OPERATION

The MAXv controller uses a PowerPC microprocessor for the core of its design. The highest priority process calculates the desired velocity at the selected update rate with a proprietary algorithm (patent number 4,734,847). This frequency is written to logic on board which generates the pulses for stepper motor control and/or the appropriate voltage levels for Servo Control. The velocity profile and synchronization of each axis is also handled by the PowerPC.

The position error is computed and applied to a PID filter if the axis is designated to be a servo axis, to determine the value of the torque output pin or the appropriate servo axis. Synchronization of all axes is performed by the PowerPC.

The commands from the VME computer are temporarily stored in a 1023 character buffer until the MAXv can parse them. The command is then executed immediately or routed to separate command queues for each axis. The command queue contains a list of addresses to execute. The argument queue stores the parameters (as applicable) supplied with each command for the axis. A command from the host may be expanded into several commands to the appropriate axis. The GO command, for example, will expand into start, ramp up, constant velocity and ramp down commands. The LS command will save its parameter in the argument queue, the loop count, on a loop stack along with the address of the LS command to be used by the next LE command as a target for a jump command are stored in the command queue. The LE command will decrement the loop count and jump to the most recent LS command providing the loop count has not reached zero. If the loop count has reached zero and it is not nested inside another loop, the queue space will be flagged as available and the next instruction in the queue will be executed.

The communication interface is performed by the MAXv microprocessor. Interrupts from the MAXv to the VME host are generated by this component. Status of the interrupts and error flags may be read by the host in the status register 4K shared memory section that is designated for the appropriate activity.

This page intentionally left blank

APPENDIX A. LIMITED WARRANTY

The Seller warrants that the articles furnished are free from defect in material and workmanship and perform to applicable, published Pro-Dex, Inc., Oregon Micro Systems specifications for one year from date of shipment. This warranty is in lieu of any other warranty expressed or implied. In no event will Seller be liable for incidental or consequential damages as a result of an alleged breach of the warranty. The liability of Seller hereunder shall be limited to replacing or repairing, at its option, any defective units which are returned f.o.b. Seller's plant. Equipment or parts which have been subject to abuse, misuse, accident, alteration, neglect or unauthorized repair are not covered by warranty. Seller shall have the right of final determination as to the existence and cause of defect. As to items repaired or replaced, the warranty shall continue in effect for the remainder of the warranty period, or for 90 days following date of shipment by Seller of the repaired or replaced part whichever period is longer. No liability is assumed for expendable items such as lamps and fuses. No warranty is made with respect to custom equipment or products produced to Buyer's specifications except as specifically stated in writing by Seller and contained in the contract.

This page intentionally left blank

APPENDIX B.

TECHNICAL SUPPORT

Pro-Dex, Inc - Oregon Micro Systems, Inc. can be reached for technical support by any of the following methods:

1. Internet E-Mail: mailto:Pro-Dex Beaverton Support Group@pro-dex.com

2. World Wide Web: http://www.pro-dex.com

3. Telephone: 8:00 a.m. - 5:00 p.m. Pacific Standard Time

   (503) 629-8081 or (800) 707-8111

4. Facsimile: 24 Hours

   (503) 629-0688 or (877) 629-0688

5. USPS: Pro-Dex, Inc.
   1800 NW 169th Place Building C-100
   Beaverton, OR 97006

RETURN FOR REPAIRS

Call Pro-Dex, Inc. Customer Service at 503-629-8081 or (800) 707-8111 or e-mail to SalesOR@pro-dex.com.
Explain the problem and we may be able to solve it on the phone. If not, we will give you a Return Materials Authorization (RMA) number. Mark the RMA number on the shipping label, packing slip and other paper work accompanying the return. We cannot accept returns without an RMA number. Please be sure to enclose a packing slip with the RMA number, serial number of the equipment, reason for return, and the name and telephone number of the person we should contact if we have further questions. Pack the equipment in a solid cardboard box secured with packing material.

Ship prepaid and insured to:

PRO-DEX, INC.
Twin Oaks Business Center
1800 NW 169th Place, Suite C100
Beaverton, OR 97006

This page intentionally left blank

APPENDIX C SPECIFICATIONS

FEATURES
PID update rate of 122 µs on all 8 axes
Delivers exceptional servo control on multi-axis applications. Identical outcomes when utilizing one or all axes of motion. Configurable PID filter with feed forward coefficients.

VME64 SPECIFICATION
The 160-pin P1/P2 connectors provide high density connectivity on the back plane. VME & VME64 compliant.

266 MHZ, 32-BIT RISC PROCESSOR
Updates all signals and data points providing superior application control.

4K SHARED MEMORY
Permits rapid data transfer to & from controller. Large size accommodates expandability to unique and custom applications.

CONTROLLER CAPABILITIES
6 Channels of general purpose analog input with 16 bit, +/-10 VDC input. 2 Channels of general purpose analog output with 16-bit +/-10 VDC output.

MOTION FEEDBACK
Support Quadrature Encoder Feed back up to 16 MHz on up to 10 encoder inputs.

SOPHISTICATED CONTROL FUNCTIONALITY
16 bit DAC analog resolution. Step pulses from 0 to 4,176,000 steps per second (+/- 0 steps). Backlash compensation. Custom, parabolic, "S"-Curve & Linear trajectory profiles. Real time encoder position capture. S-curve with 4-quadrant jerk parameters.

ADDITIONAL FEATURES

- Consumes a single VME (6U) slot
- Interface port VME P1 and P2 supports both 96 Pin and 160 pin connectors.
- Supports A16, A24 and A32 Addressing modes.
- Non-Volatile Macro Storage
- VME64bus specification ISO/IEC 15776:2001(E).
- Motion parameters continuously available in shared RAM for real time profile status.
- Electronic "mailbox" in shared RAM for priority commands, i.e. abort
- Patented technology to minimize torque ripple and velocity modulation
- Internal Watchdog timer for safety
- Slip & Stall detection with encoder feedback
- Circular interpolation
- Constant Velocity linear interpolation (all axes)
- Axis control signals are also on P1 & P2 connector.
- Output is +/-10V, or Step & Direction per Axis
- Independent home and plus / minus over-travel inputs for each axis
- Commands are intuitive for programming ease.
- Over 150 ASCII character commands, "universal" to current and previous OMS controllers
- Capable of conversion to "user" defined units i.e. inches/ revolutions if desired.
- Person to person toll-free tech support: 800-707-8111
- Easy migration path from earlier VME products

CONTROL SIGNALS

Two 68-pin SCSI and one 50-pin SCSI connectors for high density signal connection on the front panel. 16 “user definable” digital I/O. P2 connector is 160-pins and supports most all the signals available on the front panel.

DESCRIPTION

The MAXv family of Motion Controllers brings the Oregon Micro Systems, Inc. (OMS) intelligent motion control technology to a new level of servo applications as well as stepping motors. A much more powerful 266 MHz 32 Bit RISC processor (PowerPC) provides the capability and power for better and more sophisticated application control. This new generation of motion control products provides up to 8 axes of motion control on a single card to VME bus compatible computers. Each axis can be selected by the user to be an open or closed stepper or a high capability servo axis. In addition, independent analog inputs are provided to enable integration of analog parameters such as temperature, pressure, etc., under the control of the running application. Two additional encoder inputs are available for increased precision and control. Two additional general purpose analog outputs are available.
Outputs are provided for 16 bit analog servo output as well as step and direction for stepper system applications. The servo loop is a PID filter with feed-forward coefficients and an update rate of 122 μs on all 8 axes. Independent plus and minus limits, a home switch input, and an auxiliary output provided for each of the 8 axes so that the state of any of them can be monitored by the system at any time. An additional 16 User definable I/O is available for synchronization and control of other events. The voltage range of limit and home circuits has been extended for operation in the 3 to 30 VDC range. Incremental encoder feedback, differential or single ended, is used for all servo axes and is available for position feedback and may also be used for slip or stall detection. Electronic gearing is also available for tracking with another motor or manual input device, such as an independent encoder tracking.
The bus interface uses Shared Memory technology for communication of commands from the host and feedback of motion control parameters. Commands may be written to this Shared Memory by the host, eliminating the communication bottlenecks of single address port-based communications.
The MAXv uses the PowerPC’s Message unit including the door-bell technology to alert and flag the host or the Controller. Interrupt control and other data are available through reserved storage regions in the common memory area. These include the interrupt vector, interrupt control and status done flag data, over travel and home switch status, Command Error, an ASCII Command and an ASCII Response Ring Buffer, slip flag for each axis as well as the user definable I/O. Some commands may be passed to the MAXv, by passing the communication channel using the mailbox system. These commands cause an immediate interrupt and may be used for critical commands such as abort.
Each axis may perform individual unrelated moves or they can be coordinated as required by the application.
Simple two or three character ASCII commands may be easily sent to the board from any high level language, such as C, C++, or VB. Complex move sequences, time delays, and control of other external events may be programmed through the MAXv interface.
The MAXv controller supports two 68-pin and one 50-pin SCSI type connectors on the front panel as well as a 160-pin connector at P2 for back plane connections. The IOvMAX connection interface module provides an efficient means of connecting the MAXv signals to external devices. It includes two 68-pin connectors and one 50-pin connector, as well as a 100-pin connector that is backwards compatible with the VME58 front panel connector. All signals on this connector module are available on a 180 screw-terminal block.

PROGRAMMING

The MAXv motion controllers are easily programmed with character ASCII commands through an extensive command structure. The commands are combined into character strings to create sophisticated motion profiles, with features such as IO and other functionality. A separate FIFO command queue for each axis is used to store the commands once they are parsed in the MAXv. These commands are then executed sequentially, allowing the host to send a complex command sequence and attend to other tasks, while the MAXv manages the motion process. These command queues can store 800 command values and include a command queue counter that allows multiple execution of any command string.

All commands are sent to the controller as two or three character strings. Some of these commands expect one or more numerical operands to follow. These commands are identified with"#" after the command. The "#" indicates a signed integer input parameter, or a signed fixed point number of the format ##.# when User Units are enabled. User Units define distances, Velocity and acceleration parameters, and represent the input in Inches, millimeters, revolution, etc.

Synchronized moves may be made by entering the AA or AM command mode. This form of the command performs a context switch that allows entering commands in the format MRx#,y#,z#,t#,u#,v#,r#,s#;

Numbers are entered for each axis commanded to move. An axis may be skipped by entering a comma {,} at the appropriate axis position, with no value parameter. The command may be prematurely terminated with a semicolon (;) i.e. a move requiring only the X and Y axes to move would use the command MRx#,y#; followed by the GO command. Each axis programmed to move will start together upon execution of the GO command. The MAXv can be switched back to independent-axis mode by entering the desire single axis command, such as AX.

PROGRAMMING EXAMPLES

In a typical move requirement where it is desired to home the stage and then move to a specified position, the following will demonstrate the programming for a single axis:

Initialize the velocity and acceleration parameters to a suitable value. Set the PID filter gain values. Perform the homing operation initializing the position counter to zero. Perform a motion to the absolute position of 10,000 and set the done flag for that axis when the move is finished.

```
AX;
VL5000;
AC50000;
KP20;
KI1;
KD45;
HN;
HM0;
MA10000;
GO;
ID;
```

In a move requiring a three axis coordinated move to a position, the following could be used:

```
AX;
KP2;
KD6;
HN;
AY;
KP2;
KD6;
HN;
AZ;
KP2;
KD6;
HN;
AM;
VL5000,5000,5000;
AC50000,50000,50000;
MA1000,2000,3000;
GO;
ID;
```

The controller would calculate the relative velocities required to perform a straight line move from the current position to the desired position.

The following demonstrates cutting a hole with a 10,000 count radius using constant velocity contouring and circular interpolation.

The contouring velocity is set to 1000 counts per second. A contour is defined beginning at coordinates 0,0 on the Z and T axes.

Auxiliary output on the X axis is turned on, which could turn on the cutting torch or laser starting the cut at the center of the circle.

A half circle is cut from the center to the outside of the hole positioning the cutting tool at the start of the hole.

The hole is then cut, the torch turned off, the stage stopped and the contour definition completed.

The stage is then positioned and the contour definition executed.

The following would be input from the host computer:

```
CV1000;
CD,,0,0;
AN0;
CR0,5000,3.1415926;
CR0,0,6.2831853;
AF0;
MT-10,10000;
CE;
MT,,-1000,0;
GO;
CX;
```

SPECIFICATIONS

VELOCITY
0 to 4,176,000 pulses per second simultaneous on each axis

ACCELERATION
0 to 8,000,000 pulses per second per second

POSITION RANGE
128,000,000 pulses (±67,000,000)

ACCURACY
Position accuracy and repeatability ±0 counts for point to point moves
Velocity accuracy ±0.01% of peak velocity in jog mode.

POWER
+5VDC +/-5% at 1 amp typical
+12VDC at 0.1 amp typical = +/-5%
-12VDC at 0.1 amp typical = +/-10%

ENVIRONMENTAL
Operating temperature range: 0 to 50 degrees centigrade. Storage temperature range: -20 to 85 degrees centigrade. Humidity: 0 to 90% non-condensing.

DIMENSIONS
6.4" x 9.2" x 0.7"

LIMIT SWITCH INPUTS
Input levels 3-30 VDC Input sense (low or high true) selectable by command input for each axis.

CONNECTOR
Two shielded 68-Pin SCSI3 connectors for all motor control and one 50-Pin SCSI2 connector for I/O signals on front panel, and a 160-pin P2 connector for back plane interconnect.

HOME SWITCH INPUTS

USER DEFINABLE I/O
Up to 16 bits of user definable digital I/O. The 16 bits are user configurable and are configured as 8 inputs and 8 outputs as defaults from the factory.

ANALOG INPUTS
Six independent 16 Bit DAC analog resolution

ANALOG OUTPUTS (SERVO)
+/-10V and 0 to +10V, max. One per axis plus two general purposes.

STEP PULSE OUTPUT
Pulse width 50% duty cycle. Open collector level signal (TTL).

DIRECTION OUTPUT
Open collector level signal (TTL).

ENCODER FEEDBACK
Maximum 16 MHz after
4x quadrature detection. Differential signal.

REFERENCE
VME64bus Specification ISO/IEC 15776:2001(E)
VME64x Specification ANSI/VITA 1.1-1997

SOFTWARE
High level expertise not required.
Over 250 ASCII character commands, expanded from current OMS command set.
Software drivers and .DLLs for Windows®
provided at no additional cost.
User Manual included
Support software available for download at our web-site (www.pro-dex.com)

Front Panel Connectors (SCSI type)

50-Pin General Purpose I/O (J3)					
Pin	FUNCTION	Pin	FUNCTION	Pin	FUNCTION
1	ADC0	18	Phase A9-	35	GND
2	AGND	19	IO5 Phase B9+	36	IO9
3	ADC2	20	Phase B9-	37	GND
4	AGND	21	IO6/ Index 9+	38	IO10
5	ADC4	22	Index 9-	39	GND
6	AGND	23	IO7	40	IO11
7	DAC8	24	GND	41	GND
8	AGND	25	5VDC	42	IO12
9	IO0/ Phase A8+	26	ADC1	43	GND
10	Phase A8-	27	AGND	44	IO13
11	IO1/ Phase B8+	28	ADC3	45	GND
12	Phase B8-	29	AGND	46	IO14
13	IO2/ Index 8+	30	ADC5	47	GND
14	Index 8-	31	AGND	48	IO15
15	IO3	32	DAC9	49	GND
16	GND	33	AGND	50	12VDC
17	IO4/ Phase A9+	34	IO8		

68-Pin 4 Axis List (J5)			
1	X Phase A+	35	X Index +
2	X Phase A -	36	X Index -
3	X Phase B +	37	X STEP
4	X Phase B -	38	GND
5	Y SERVO	39	X SERVO
6	AGND	40	AGND
7	Y Home	41	X Home
8	Y Dir	42	X Dir
9	Y Aux	43	X Aux
10	GND	44	GND
11	Y Pos Limit	45	X Pos Limit
12	Y Neg Limit	46	X Neg Limit
13	Y Phase A +	47	Y Index +
14	Y Phase A -	48	Y Index -
15	Y Phase B +	49	Y STEP
16	Y Phase B -	50	GND
17	+5V	51	V-BIAS
18	GND	52	GND
19	Z Phase A +	53	Z Index +
20	Z Phase A -	54	Z Index -
21	Z Phase B +	55	Z STEP
22	Z Phase B -	56	GND
23	T SERVO	57	Z SERVO
24	AGND	58	AGND
25	T Home	59	Z Home
26	T Dir	60	Z Dir
27	T Aux	61	Z Aux
28	GND	62	GND
29	T Pos Limit	63	Z Pos Limit
30	T Neg Limit	64	Z Neg Limit
31	T Phase A +	65	T Index +
32	T Phase A -	66	T Index -
33	T Phase B +	67	T Step
34	T Phase B -	68	GND

68-Pin 4 Axis List (J4)			
1	U Phase A+	35	U Index+
2	U Phase A-	36	U Index-
3	U Phase B+	37	U STEP
4	U Phase B-	38	GND
5	V SERVO	39	U SERVO
6	AGND	40	AGND
7	V Home	41	U Home
8	V Dir	42	U Dir
9	V Aux	43	U Aux
10	GND	44	GND
11	V Pos Limit	45	U Pos Limit
12	V Neg Limit	46	U Neg Limit
13	V Phase A +	47	V Index +
14	V Phase A -	48	V Index -
15	V Phase B +	49	V STEP
16	V Phase B -	50	GND
17	+5V	51	V-BIAS
18	GND	52	GND
19	R Phase A +	53	R Index +
20	R Phase A -	54	R Index -
21	R Phase B +	55	R STEP
22	R Phase B -	56	GND
23	S SERVO	57	R SERVO
24	AGND	58	AGND
25	S Home	59	R Home
26	S Dir	60	R Dir
27	S Aux	61	R Aux
28	GND	62	GND
29	S Pos Limit	63	R Pos Limit
30	S Neg Limit	64	R Neg Limit
31	S Phase A +	65	S Index +
32	S Phase A -	66	S Index -
33	S Phase B +	67	S Step
34	S Phase B -	68	GND

P2 Connector Pin out at Back plane
(Rows A, B, & C are VME58 compatible)

MAXv Pin Assignment (P2 Connector					
Pin	Row Z	Row A	Row B	Row C	Row D
1	X Phase B -	X Phase B +	+5V	X Index +	X Index -
2	GND	X Step	GND	X Phase A +	X Phase A -
3	ADC0/IO8	X Pos LMT	RSVD	X Dir	X Aux
4	GND	X Neg LMT	A24	X Home	I/O0
5	Y Phase B -	Y Phase B +	A25	Y Index +	Y Index -
6	GND	Y Step	A26	Y Phase A +	Y Phase A -
7	ADC1/IO9	Y Pos LMT	A27	Y Dir	Y Aux
8	GND	Y Neg LMT	A28	Y Home	I/O1
9	Z Phase B -	Z Phase B	A29	Z Index +	Z Index -
10	GND	Z Step	A30	Z Phase A +	Z Phase A -
11	ADC2/IO10	Z Pos LMT	A31	Z Dir	Z Aux
12	GND	Z Neg LMT	GND	Z Home	I/O2
13	T Phase B -	T Phase B	+5V	T Index +	T Index -
14	GND	T Step	D16	T Phase A +	T Phase A -
15	ADC3/IO11	T Pos LMT	D17	T Dir	T Aux
16	GND	T Neg LMT	D18	T Home	I/O3
17	U Phase B -	U Phase B	D19	U Index +	U Index -
18	GND	U Step	D20	U Phase A +	U Phase A -
19	ADC4/IO12	U Pos LMT	D21	U Dir	U Aux
20	GND	U Neg LMT	D22	U Home	I/O4
21	V Phase B -	V Phase B	D23	V Index +	V Index -
22	GND	V Step	GND	V Phase A +	V Phase A -
23	ADC5/IO13	V Pos LMT	D24	V Dir	V Aux
24	GND	V Neg LMT	D25	V Home	I/O5
25	R Phase B -	R Phase B	D26	R Index +	R Index -
26	GND	R Step	D27	R Phase A +	R Phase A -
27	DAC8/IO14	R Pos LMT	D28	R Dir	R Aux
28	GND	R Neg LMT	D29	R Home	I/O6 / 5 AUX
29	S Phase B -	S Phase B	D30	S Index +	S Index -
30	GND	S Step	D31	S Phase A +	S Phase A -
31	DAC9/IO15	S Pos LMT	GND	S Dir	NC
32	GND	S Neg LMT	+5V	S Home	NC

ORDERING INFORMATION				
Model	Computer Interface	Axes	Servo/Stepper	User I/O
MAXv-1000	VME Bus	1	User Definable	25
MAXv-2000		2	User Definable	26
MAXv-3000		3	User Definable	27
MAXv-4000		4	User Definable	28
MAXv-5000		5	User Definable	29
MAXv-6000		6	User Definable	30
MAXv-7000		7	User Definable	31
MAXv-8000		8	User Definable	32

ACCESSORIES	
IOvMAX	I/O Breakout Board for MAXv (without Cable)
CBL50-10	I/O cable for IOvMAX - 10 ft
CBL68-10	10 ft cable w/mating connector, 68-pin

This page is intentionally left blank.

INDEX