USER’S MANUAL

INTELLIGENT MOTION CONTROLLERS

For VME and VME64 bus

MAXv FAMILY

OMS Motion, Inc
15201 NW Greenbrier Parkway
B-1 Ridgeview
BEAVERTON, OR 97006
PHONE 503-629-8081
FAX 503-629-0688
EMAIL: sales@omsmotion.com

WEB SITE: http://www.omsmotion.com/

TRADEMARKS

IBM, IBM PC, IBM PC/XT, IBM PC/AT, IBM PS/2, and IBM PC DOS are registered trademarks of International Business Machines Corporation. CompactCPCI, PICMG-PCI, PICMG are registered trademarks of the PCI Special Interest Group. LabView is a registered trademark of National Instruments. Windows, Windows 2000, Win 95, Win 98, Win XP, Win NT & Vista are registered trademarks of Microsoft Corporation.

DISCLAIMER

OMS Motion, Inc. makes no representations or warranties regarding the contents of this document. We reserve the right to revise this document, or make changes to the specifications of the product described within it at any time without notice and without obligation to notify any person of such revision or change.

3301-2700000

Rev. F

TABLE OF CONTENTS

1. GENERAL DESCRIPTION

1.1. INTRODUCTION

The OMS Motion, Inc. 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.

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, see Figure 2-5. 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 (PS command). 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.)

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 OMS Motion, Inc. 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 2-5 for MAXv component locations.)

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

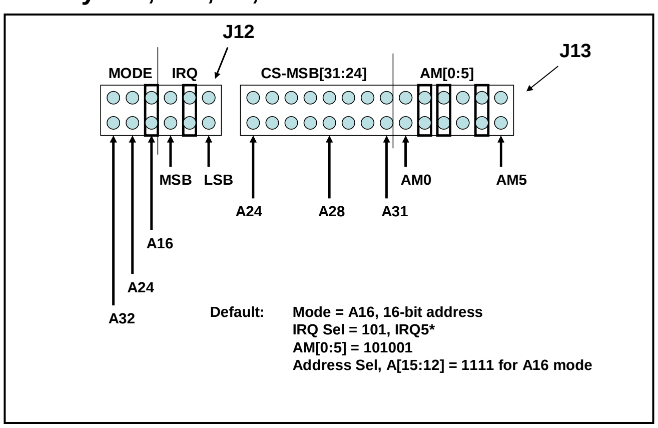


FIGURE 2-1, VME BOARD ADDRESS AND OPERATION OPTION JUMPER SWITCH DIAGRAM

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”.

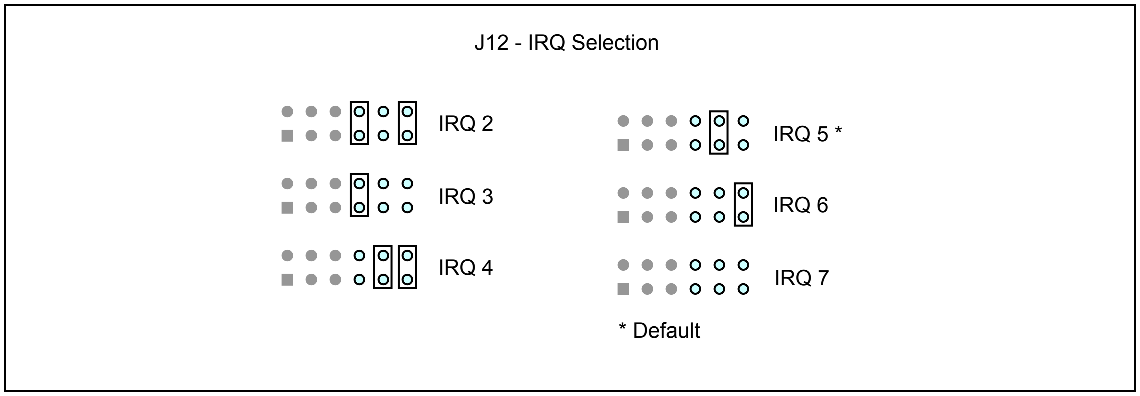


FIGURE 2-2 J12, IRQ SELECTION

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.

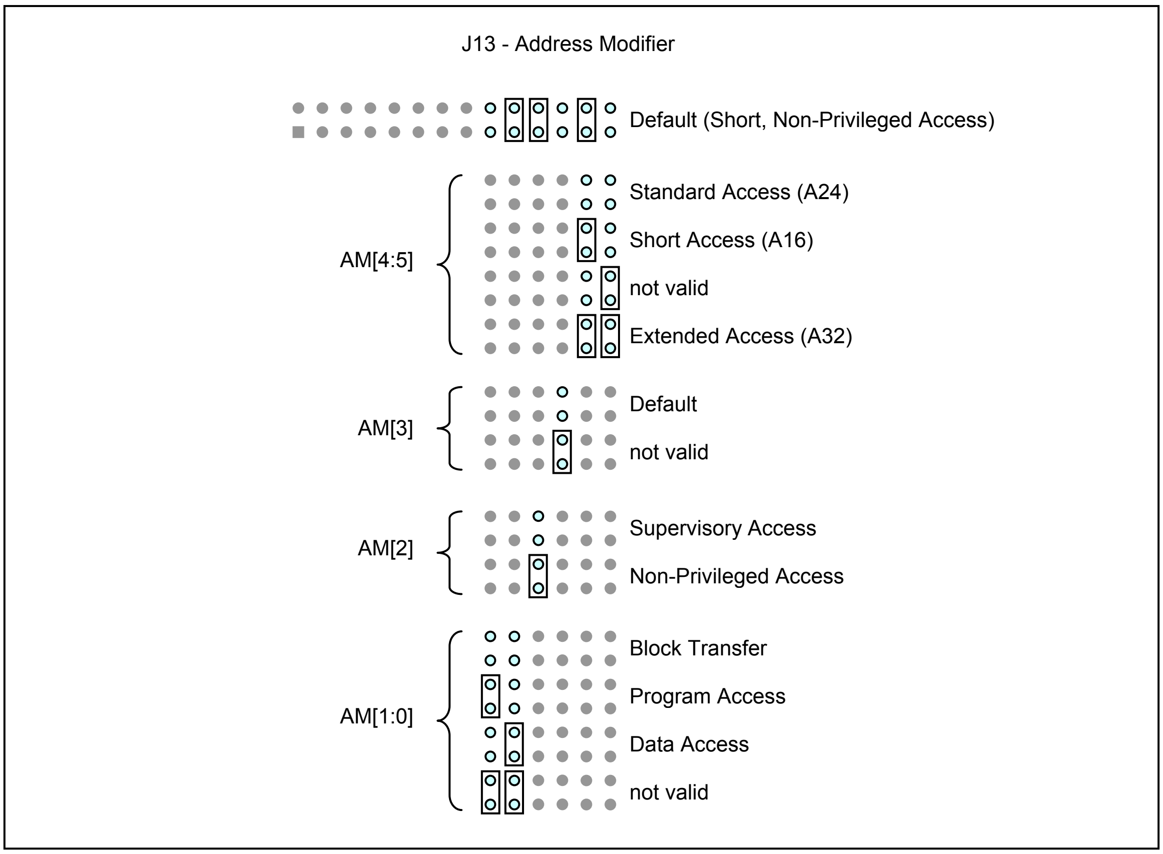
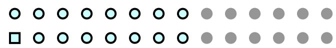


FIGURE 2-3 J13, ADDRESS MODIFIER

J13 Base Address Selection



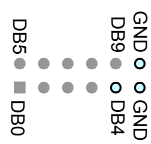
Default (0xF000 in Short mode)

		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 2-4 J13, BASE ADDRESS SELECTION

J11 allows to set the controllers communication to serial upgrade mode. The jumper setting is sampled at startup. This mode allows the controller firmware to upgraded through a regular RS232 interface connected to JP1. During this mode the controller does not process instructions from the VME interface.

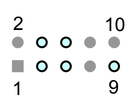
J11 upgrade mode selection



Default (GND-GND)

Jumper setting	*Mode*
GND-GND	*VME mode (default)*
DB4-GND	*Serial upgrade mode using JP1*

JP1 RS232 Connector



Pin	*Description*
3	*RxD*
4	*RTS*
5	*TxD*
6	*CTS*
9	*GND*

FIGURE 2-5 J11 AND JP1, DEBUG MODE SELECTION

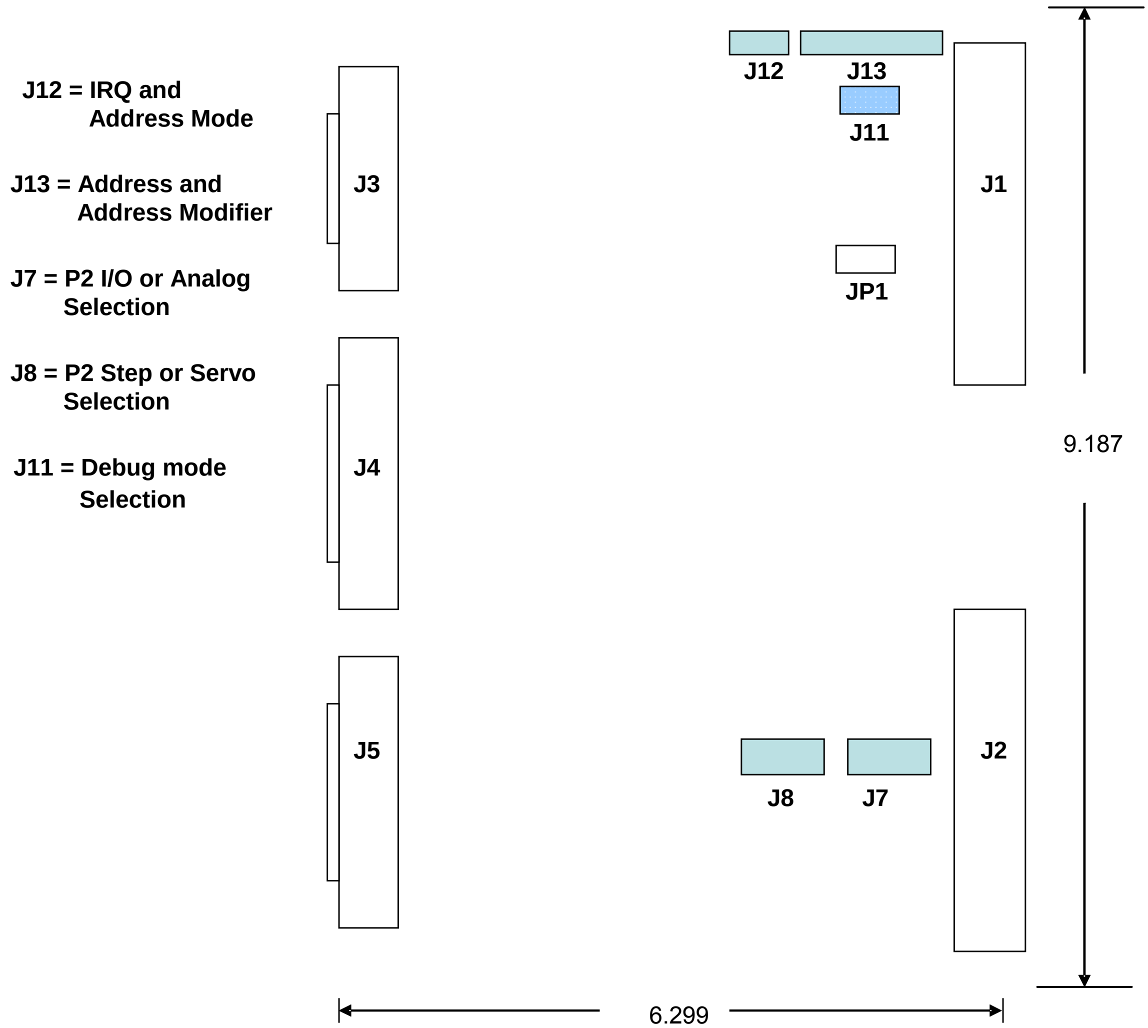


FIGURE 2-6 MAXVPCB DIAGRAM

NOTE: The J2 Backplane connector interface signals are selected on J8 as Shown in Figure 2-6.

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 J7 and J8. (See also Figures 2-5 and 2-6.)

The signals for both the STEP and SERVO signals for each axis are available on the front panel connectors (J4) and (J5), and hence are available at all times. However, only the STEP or the SERVO signal for each axis is available to the backplane via the (J2) connector. Whether the STEP or the SERVO signal is routed to the (J2) connector is determined by the jumper settings on jumper J8 (See Figures 2-5 and 2-6).

The signals for both digital IO channels 8-15 and Analog IO channels (ADC 0-6 and DAC 8-9) are available on the front panel connector (J3), and hence are available at all times. However, only 8 of these 16 signals are available to the backplane via the (J2) connector. Which of the 8 signals are routed to the (J2) connector is determined by the jumper settings on jumper J7 (See Figures 2-5 and 2-6).

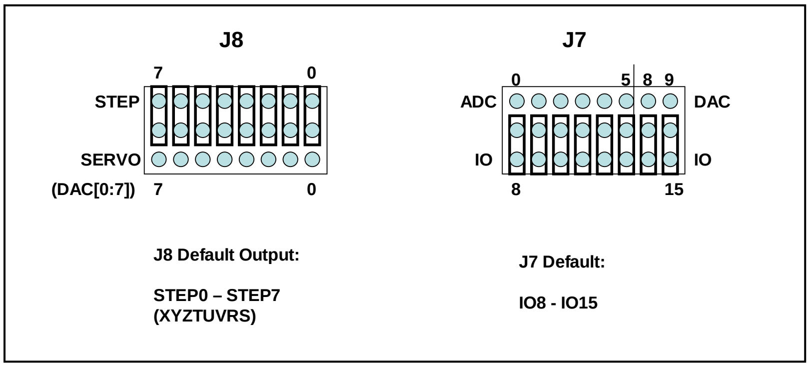


Figure 2-7 STEP/SERVO AND IO/ANALOG JUMPER SWITCH FOR THE BACKPLANE P2 INTERFACE DIAGRAM

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 electrostatic 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.

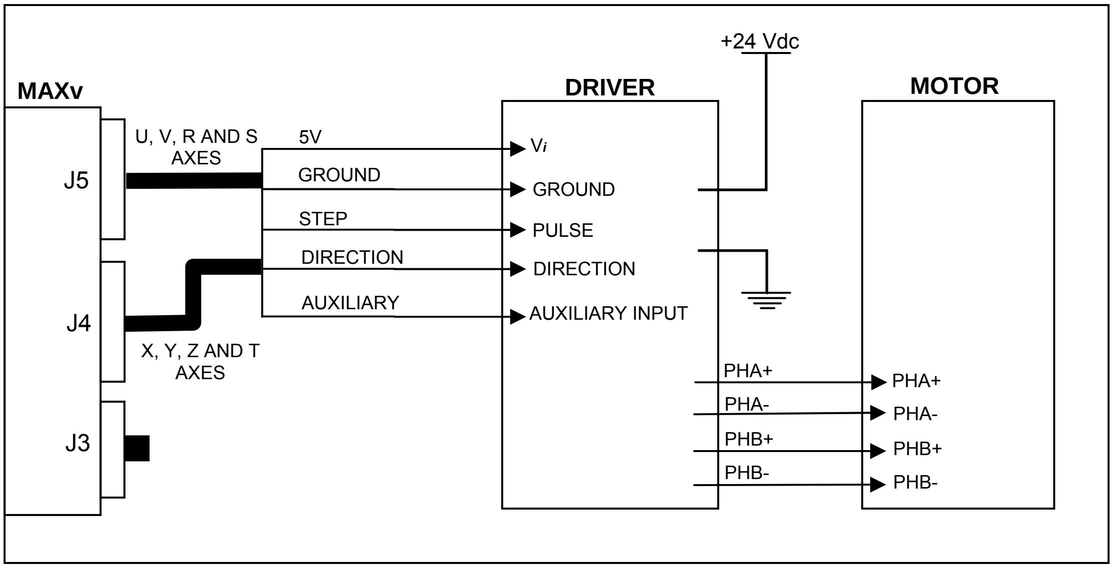


FIGURE 2-8
Example of Wiring Diagram of MAXv Controller Connected to a Stepper Driver / Motor

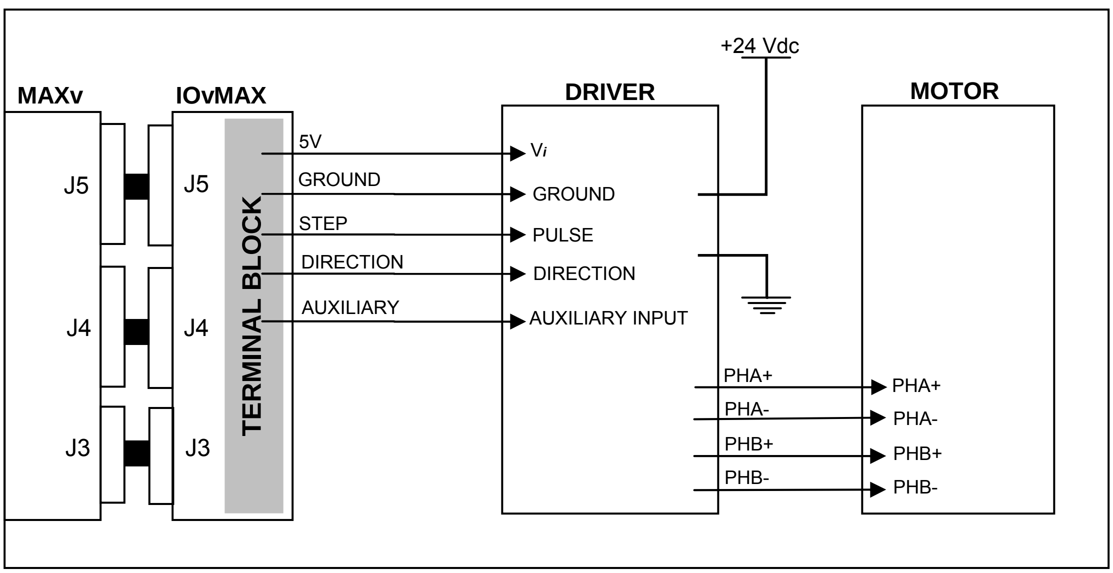


FIGURE 2-9 Example of Wiring Diagram of MAXv Controller via the IOvMAX Interface Module

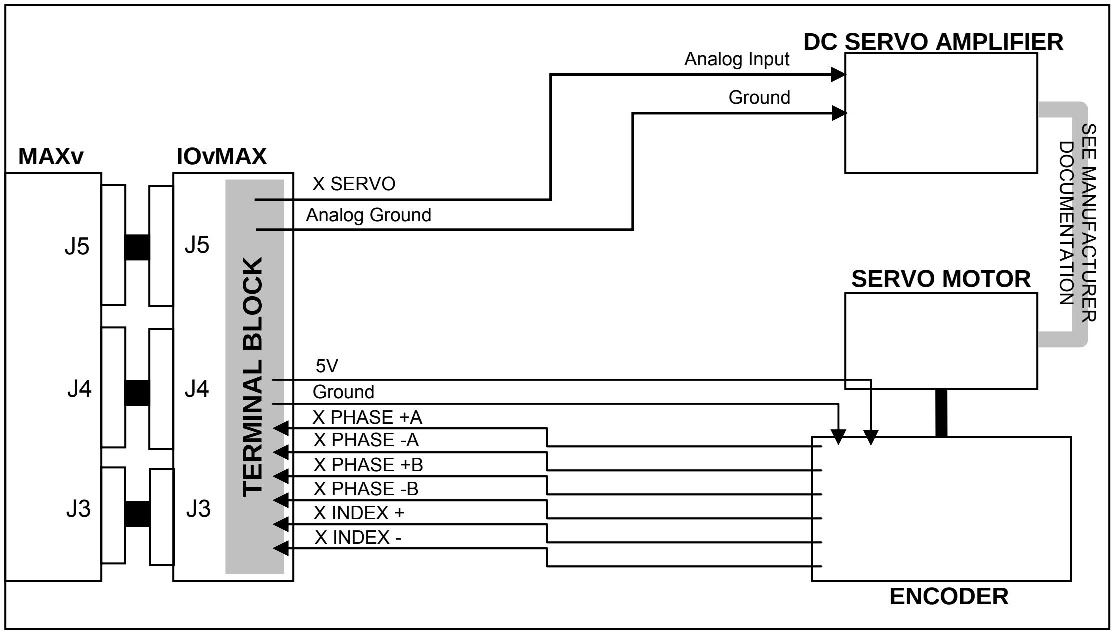


FIGURE 2-10 Example of Wiring Diagram of MAXv Controller via the IOvMAX Interface Module to Servo Motor

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.

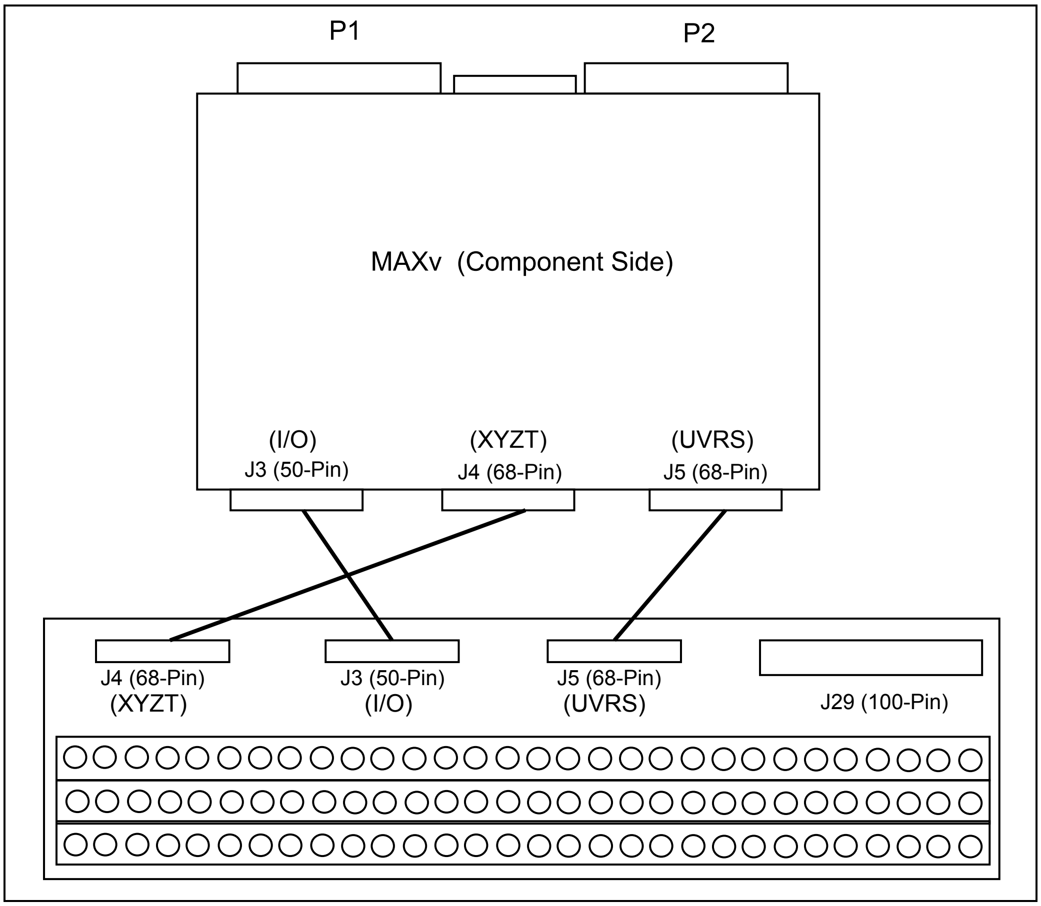


Figure 2-11 IOvMAX Break-Out to MAXv

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.

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.

2.6. 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 MAXnet to the amplifier.
3. Balance your motor:
    a. Configure the axis as a servo axis by sending the "PSM" command.
    b. Using a voltage meter, verify that the command signal from the MAXnet is less than 500mV. If it is not, send the command "KO0;" to the MAXnet and recheck the voltage. If the voltage is still too high, contact OMS Motion, Inc.'s Technical Support department for guidance.
    c. Turn on power to the amplifier and then to the motor.
    d. Adjust the balance setting of your amplifier (if equipped) until the motor stops moving.
    e. If the motor continues to revolve or your amplifier has no balance adjustment:
        i. Send the command "KO100;" to the MAXnet.
        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 MAXnet's DAC (this method works only with incremental encoders, skip it if you use absolute encoder only on that axis):

a. Connect the servo encoder to the MAXnet. (See section 4.4 on incremental encoder feedback)

b. Set the signal/command gain of your amplifier to its minimum setting.

c. Send the "KO3277;" command to the MAXnet and observe the velocity of the motor. The output of MAXnet 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 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 MAXnet 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 to the MAXnet.

5. Verify the direction of your servo encoder:

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

   b. Send the "RE;" command to the MAXnet 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 or analog output must be reversed use one of the methods below.

      i. Use EDI/EDN to invert/normalize encoder direction or

      ii. Use SVP-/SVP+ to invert/normalize PID analog output (inverts values of KO and KOD) or

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

      iv. If your incremental 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 OMS 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.

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.

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

2.7. TUNE THE SYSTEM

2.7.1. INTRODUCTION

The following is an introduction to the basics of tuning a servo motor. 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. 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.

2.7.2. TUNING ASSISTANT

MAXTune.exe is a tuning assistant utility that 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 application and documentation can be found on the CD-ROM supplied with the MAXnet and on OMS' web site found at www.omsmotion.com

2.7.3. MANUAL TUNING

In most 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 rate rpm. If the system requirement is for a velocity higher than 90% of the motors 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 measuring 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 MAXnet can control either current mode or voltage mode amplifiers. The #UR command sets the servo update rate of the MAXnet to one of the following rates: 976.6μs, 488.3μs, 244.1μs, 122.1μs. This affects the responsiveness of the system. High "Following Error" can be compensated for using the feedforward coefficients (KV and KA commands) 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:

KP "Following Error" = (3°/360°) × (counts per revolution)

Voltage mode:

KP "Following Error" = (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:

KI "Following Error" = 0 counts

Voltage Mode:

KI "Following Error" = (80°/360°) × (counts per revolution)

While still in open-loop mode (CL0;) 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 –32767.00 to 32767.00.

Set the previously determined values for maximum velocity, maximum acceleration and the move distance for a trapezoidal profile with at least a 20% flat spot at peak velocity. Use the following formula to determine the move distance:

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

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

Set the KD and KI variables to 0, and the KP variable to 1, and execute the move by sending the move commands to the MAXnet.

Example:
```
MR6000;
GO;
```

Adjust the KP term while repeating the above move command 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;
KP1;
CL1;
MR6000;
GO;
LP0;
KP2;
CL1;
MR6000;
GO;
LP0;
KP4;
HN;
MR6000;
GO;
LP0;
KD10;
CL1;
LP0;
KP8;
CL1;
MR6000;
GO;
LP0;
KD100;
CL1;
```

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.

Once the KP term has been obtained, it can be used to determine the initial value for the KI term. Set the KI and KU variables to 4 times the KP value. The KI term is a gain applied to the accumulated position error over time. The KU variable limits the amount the KI term can contribute to the PID. Continue executing the motion profile and raising the KU 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 will increase the response time of your system. The motion profile should also have a steeper slope as KI increases (see figures 2-9 and 2-10 below). However, as KI increases the system can also become unstable. When the increased KI values cause unacceptable instability, increase the KD parameter. This will increase the dampening on the system’s motion profile; therefore, reducing oscillation or “ringing”. Continue adjusting the KI/KU and KD terms until the proper response time is obtained.

The values for KI range from 0.00 to 32767.00.

The values for KU range from 0.00 to 32767.00.

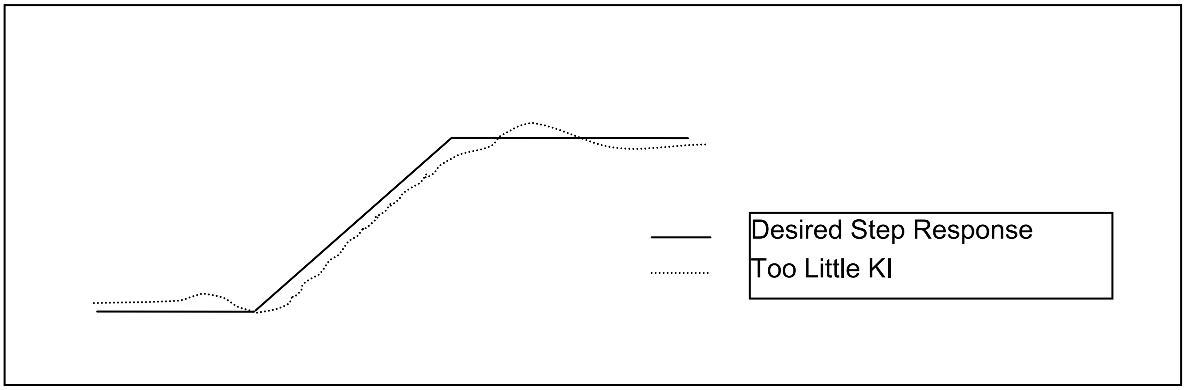


FIGURE 2-12

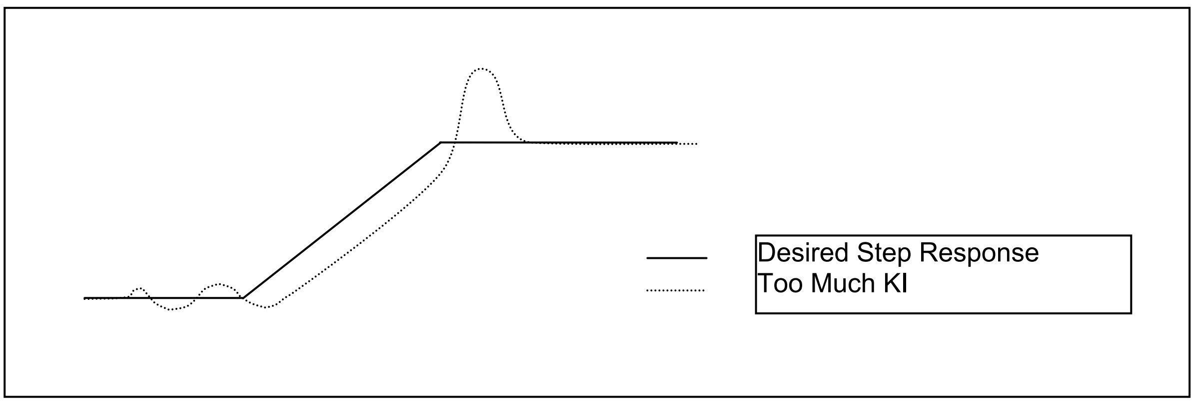


FIGURE 2-13

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.

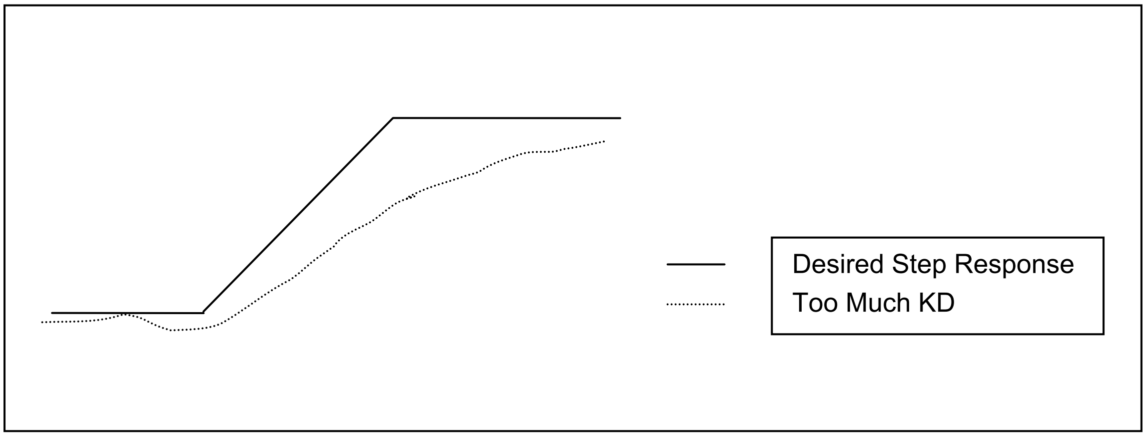


FIGURE 2-14

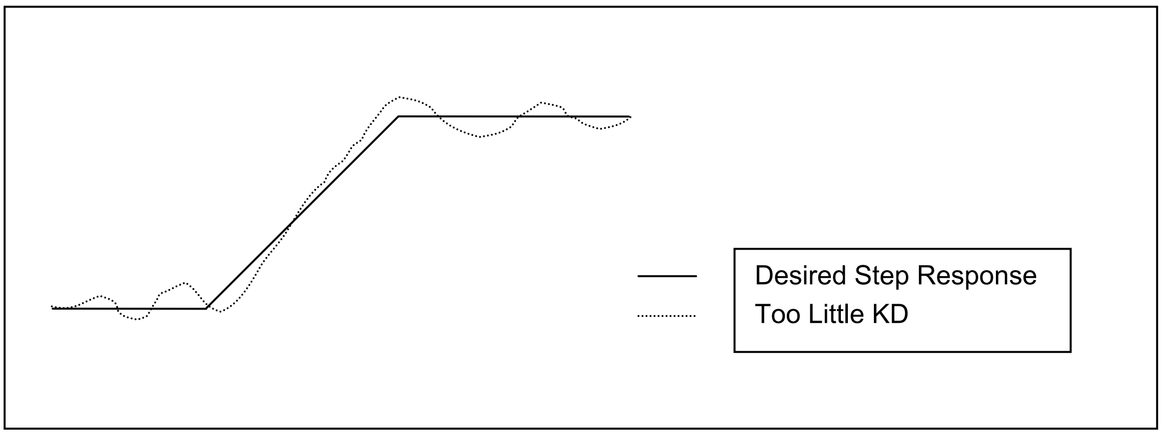


FIGURE 2-15

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, 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.

The KV variable is the velocity feedforward coefficient, and compensates for friction that is proportional to velocity. Unlike KP, KI, and KD, which have to wait for system error before responding, the KV variable has an immediate effect on the commanded move, and is a gain applied to the current velocity. KV makes the system more responsive, and 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.

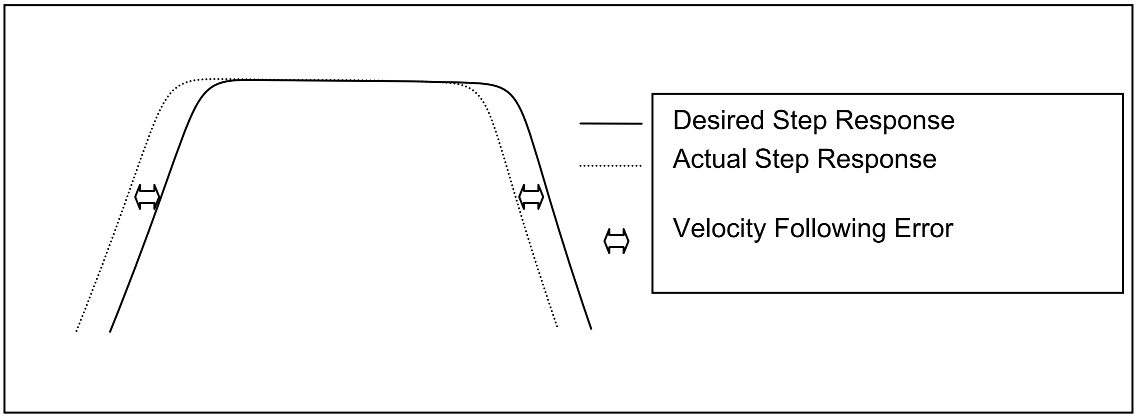


FIGURE 2-16

The KA variable is the acceleration feedforward coefficient, and compensates for inertia. Like KV, the KA variable does not operate on system error, and is applied as a gain to the current acceleration and deceleration. 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, but if KA is too large, instability may occur.

The values for KA range from 0.00 to 32767.00.

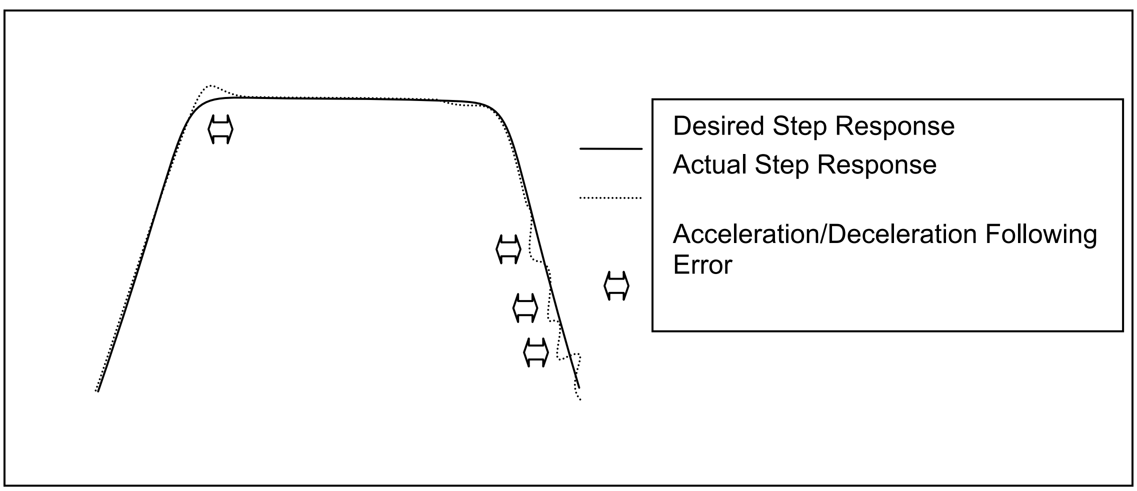


FIGURE 2-17

The KF variable is the friction offset coefficient, and compensates for static friction. The KF variable does not operate on system error, and is applied to all commanded moves. KF increases all portions of the motion profile. If KF is too large, instability may occur.

The values for KF range from 0.00 to 32767.00.

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

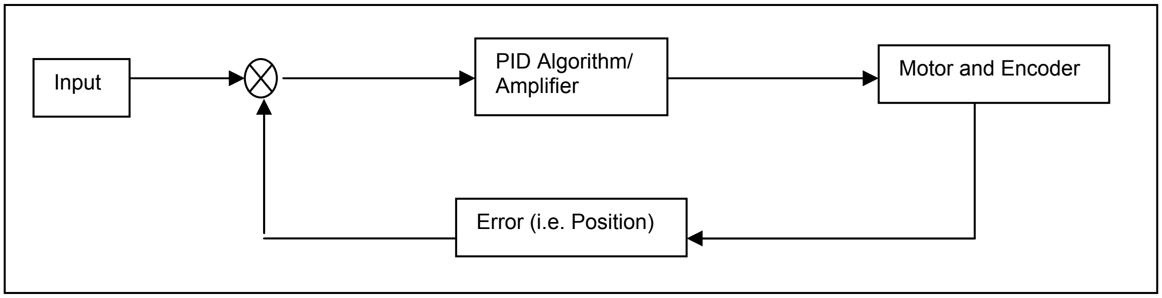


Figure 2-18 FEEDBACK LOOP

To verify that your motor is tuned properly, send the commands LP0;CL1; 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.

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 32768, 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 motor position readings you can fine-tune your PID filter according to the earlier steps.

You may want to save the values for KP, KI, KD, 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. This can be done by using the APP command. These saved parameters will then be used as the power up default set of values.

2.8. 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.

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	LM
Over travel limit (enabled or disabled)	Enabled	LM
Over travel limit polarity (active high or active low)	Active low	LT
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	AJ, RT, 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	AB, 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	HT
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	SV
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
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

2.9. 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.

2.10. POWER-UP/RESET SELF-TEST DIAGNOSTICS

During power-up/reset initialization the MAXv controller performs a Built-in Self-Test diagnostic routine. During this self-test the LEDs D2, D4, and D6 go through a series of states. If the initialization sequence fails in any way, the state of the LEDs will indicate the type of initialization error as follows:

LED	Meaning
GREEN LED (D4) on steady	initialization completed successfully
GREEN LED (D6) on steady	waiting for EPIC RESET completion
GREEN LEDs (D4,D6) on steady	waiting for FPGA ready
GREEN LED (D4) flashing	firmware not resident in flash - waiting for firmware download
RED LED (D2) flashing with GREEN LED (D4) on steady	invalid processor version
RED LED (D2) flashing with GREEN LED (D6)	memory compare error

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.

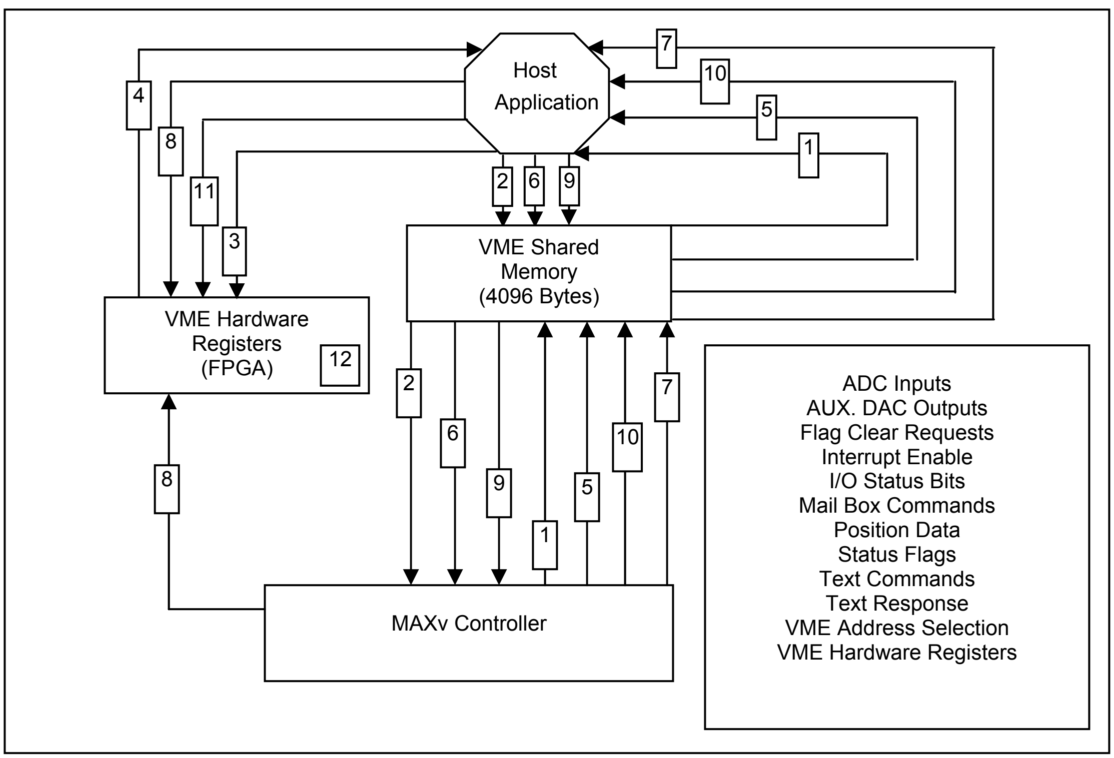


Figure 3-1 Data Flow Diagram

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) 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.

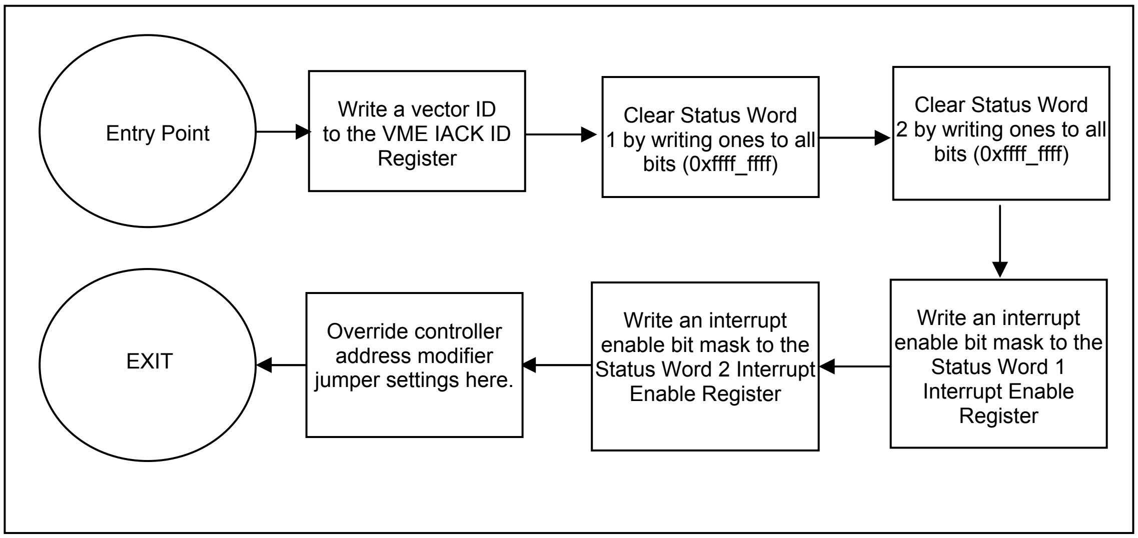
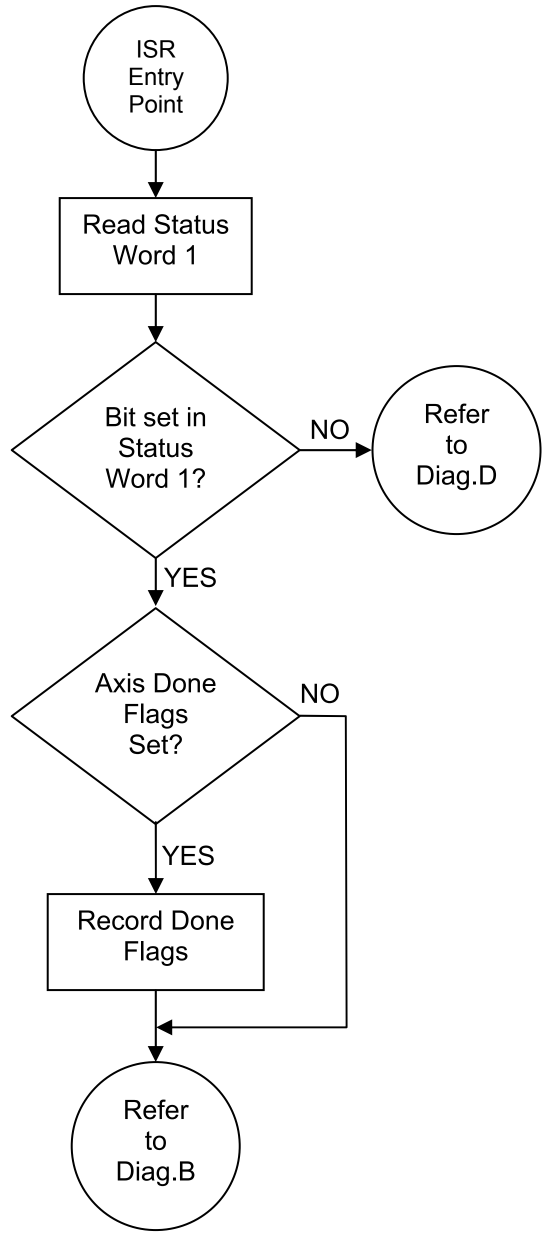


Figure 3-2 Controller Initialization

3.7.1. SAMPLE OF AN INTERRUPT SERVICE ROUTINE

Diagram



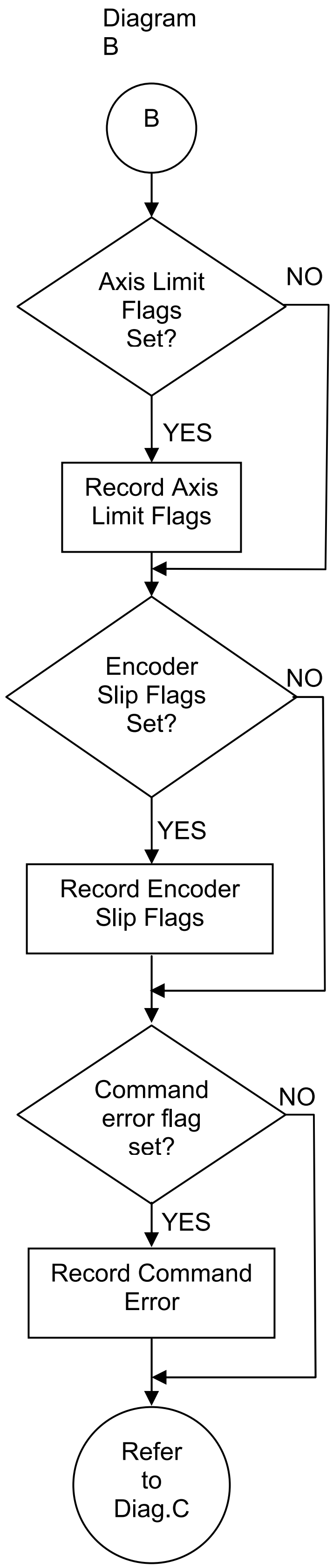


Diagram
B
B
Axis Limit
Flags
Set?
NO
YES
Record Axis
Limit Flags
Encoder
Slip Flags
Set?
NO
YES
Record Encoder
Slip Flags
Command
error flag
set?
NO
YES
Record Command
Error
Refer
to
Diag.C

Diagram C

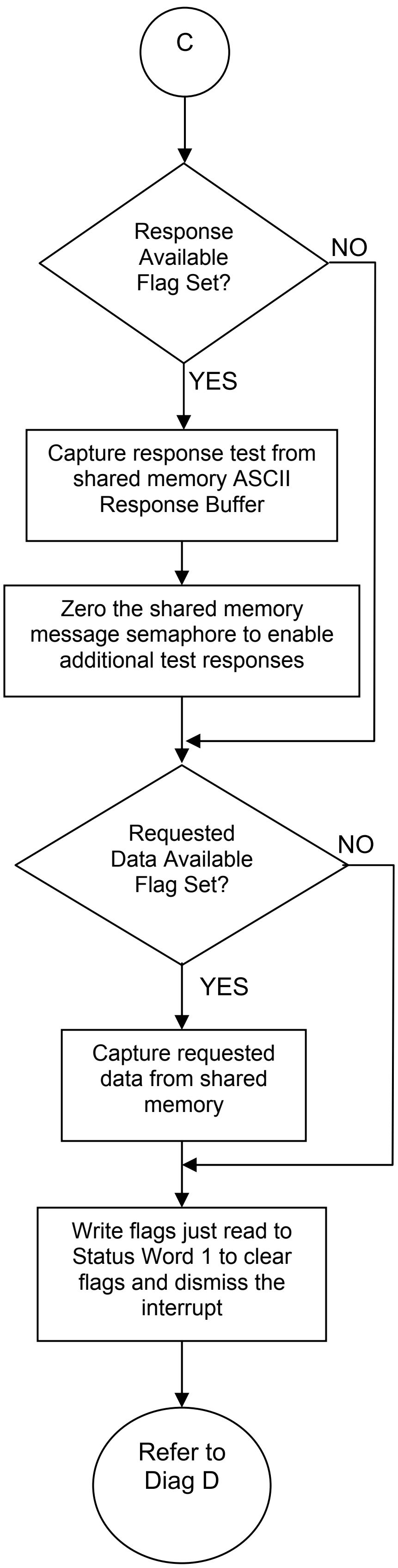
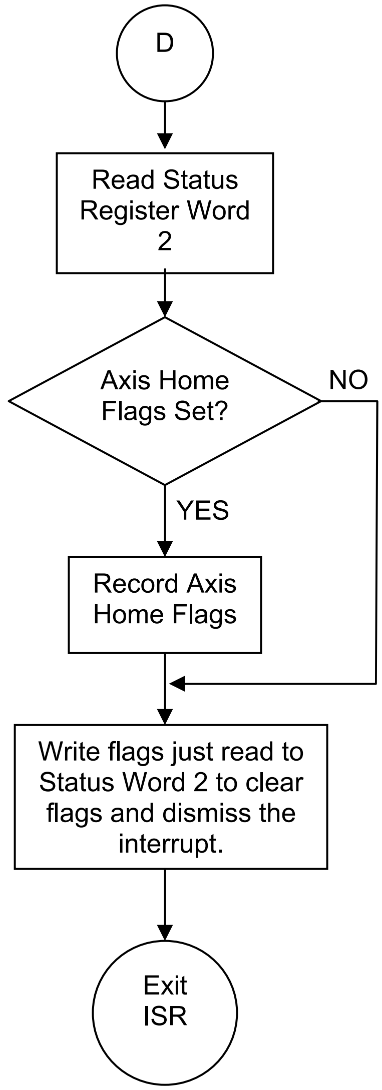
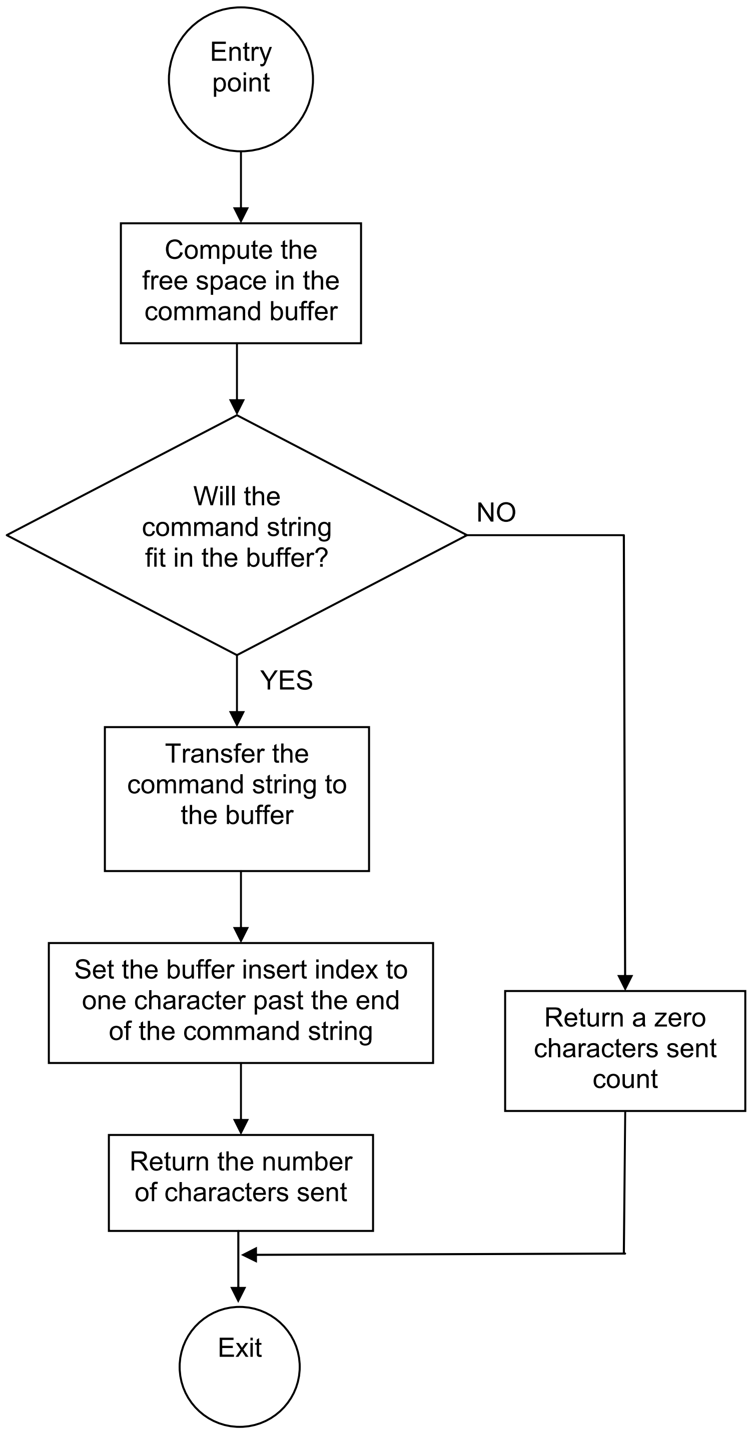

C
Response
Available
Flag Set?
NO
YES
Capture response test from
shared memory ASCII
Response Buffer
Zero the shared memory
message semaphore to enable
additional test responses
Requested
Data Available
Flag Set?
NO
YES
Capture requested
data from shared
memory
Write flags just read to
Status Word 1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

Diagram D


D
Read Status Register 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



3.7.3. SAMPLE OF A SENDANDGETSTRING

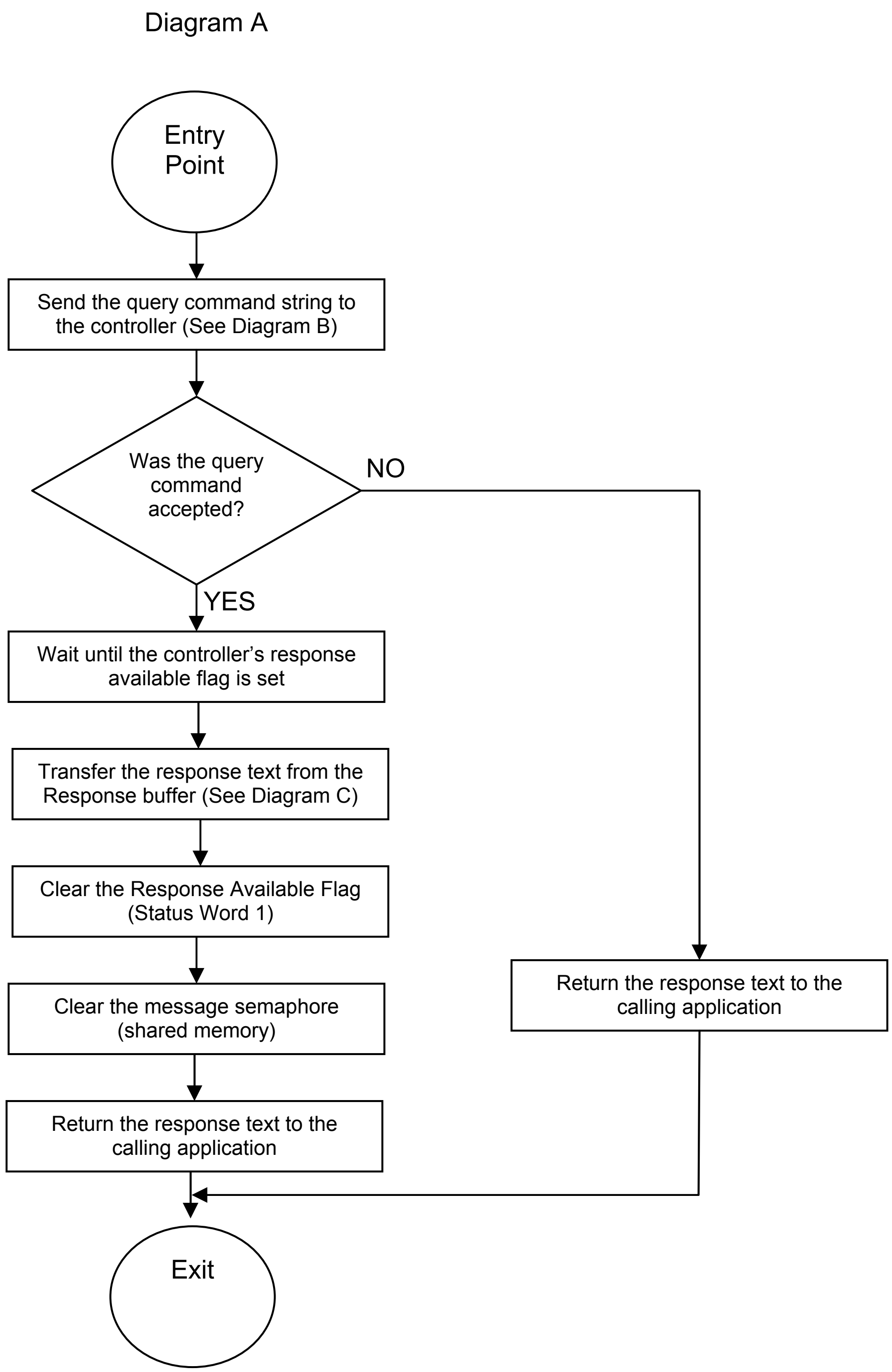


Diagram B

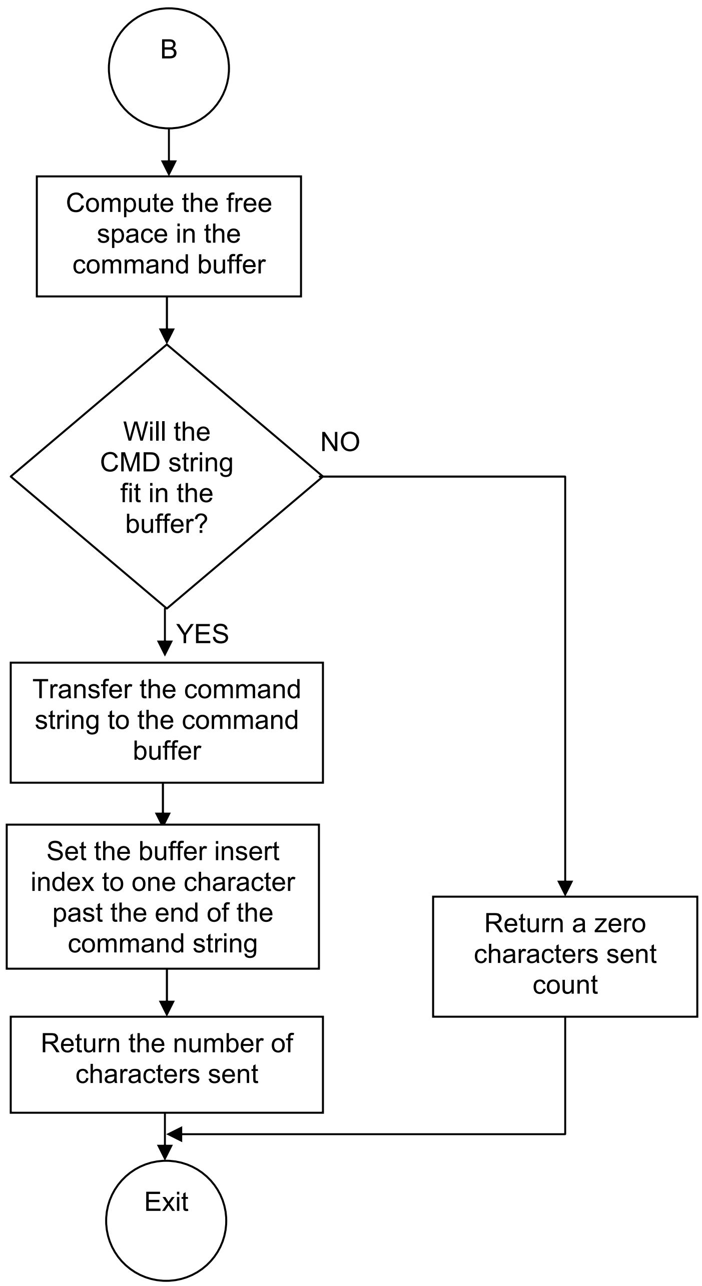
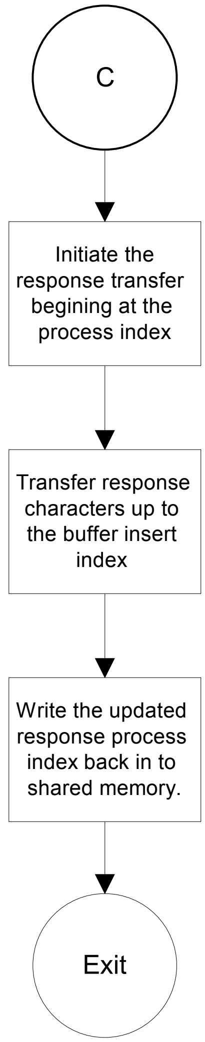

B
Compute the free space in the command buffer
Will the CMD string fit in the buffer?
NO
YES
Transfer the command string to the command buffer
Set the buffer insert index to one character past the end of the command string
Return a zero characters sent count
Return the number of characters sent
Exit

Diagram C



NOTE: The WY (Who Are You) 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
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)

Byte Offset	Byte Offset (Hex)	Byte length	Description
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 words contain the axis absolute encoder positions. It is updated each update cycle.			
168	0xa8	4	X axis absolute encoder position
172	0xac	4	Y axis absolute encoder position
176	0xb0	4	Z axis absolute encoder position
180	0xb4	4	T axis absolute encoder position
184	0xb8	4	U axis absolute encoder position
188	0xbc	4	V axis absolute encoder position
192	0xc0	4	R axis absolute encoder position
196	0xc4	4	S axis absolute encoder position
200	0xc8	36	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	36	Reserved
The following memory region contains analog I/O data.			
3364	0xd24	4	Auxiliary DAC 0
3368	0xd28	4	Auxiliary DAC 1
3372	0xd2C	4	Analog Input 0
3376	0xd30	4	Analog Input 1
3380	0xd34	4	Analog Input 2
3384	0xd38	4	Analog Input 3
3388	0xd3C	4	Analog Input 4
3392	0xd40	4	Analog Input 5
3396	0xd44	8	Reserved
The following memory region contains auxiliary encoder data.			
3404	0xd4C	4	Auxiliary Encoder 0
3408	0xd50	4	Auxiliary Encoder 1
The following memory region contains servo analog output data.			
3412	0xd54	4	X axis servo DAC output
3416	0xd58	4	Y axis servo DAC output
3420	0xd5C	4	Z axis servo DAC output
3424	0xd60	4	T axis servo DAC output

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

3428	0xd64	4	U axis servo DAC output
3432	0xd68	4	V axis servo DAC output
3436	0xd6C	4	R axis servo DAC output
3440	0xd70	4	S axis servo DAC output
3444	0xd74	4	Coherent X axis servo DAC output
3448	0xd78	4	Coherent Y axis servo DAC output
3452	0xd7C	4	Coherent Z axis servo DAC output
3456	0xd80	4	Coherent T axis servo DAC output
3460	0xd84	4	Coherent U axis servo DAC output
3464	0xd88	4	Coherent V axis servo DAC output
3468	0xd8C	4	Coherent R axis servo DAC output
3472	0xd90	4	Coherent S axis servo DAC output
3476	0xd94	552	Real-time position capture data
4028	0xFBC	4	Reserved
The remaining 64 bytes are reserved for the controller's hardware register set:			
4032	0xfc0	4	Status Word 1 flag register
4036	0xfc4	4	Status Word 1 interrupt enable register
4040	0xfc8	4	Status Word 2 flag register
4044	0xfcc	4	Status Word 2 interrupt enable register
4048	0xfd0	4	VME IACK ID Vector
4052	0xfd4	4	Controller configuration switch register
4056	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) STATUS WORD 1 INTERRUPT ENABLES"

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	Real-time position capture data available
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	Real-time position capture data available enable
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
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

Byte Offset (10)	Byte Offset (Hex)	Byte length	Description	Mem. Function
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
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
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

Byte Offset (10)	Byte Offset (Hex)	Byte length	Description	Mem. Function
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

3.10. REAL-TIME POSITION CAPTURE

The position capture commands control the real-time recording of axis position data and the management of the captured position data. The captured position data includes the axis, the positive edge I/O bits, the negative edge I/O bits, the home and encoder home events, and the encoder position of the axis. The position data is captured when the conditions specified for the input bit are met. The capture conditions for the home switch and general purpose input bits can be a rising/positive edge, a falling/negative edge, or the event can be both the rising/positive and the falling/negative edge so data is captured on any transition of the input bit. The real-time position capture feature is only available on an axis with incremental encoders. See the MAX family command reference manual for more details on the real-time position capture feature.

The MAXk controller has a ring buffer in VME shared memory, which is used to transfer the real-time position capture data to the host. When a capture event is recorded by the motor update cycle routine, it transfers the capture table entry to the shared VME memory. The host is signaled that the data is available via bit number 8, or hexadecimal value 0x00000100, in the controller status word 2 register at offset address 0x00000FC8. The data available bit is also available at byte offset 0x00000FCA, and bit number 0 or 0x01. The shared memory for the capture data is implemented as a ring buffer with an insert index that the controller uses to insert data into the shared memory region, and a removal index that the host uses to remove data from shared memory region. The controller places the capture data into the ring buffer at the location specified by the insert index, and advances the insert index. If after being advanced, the insert index equals the removal index, then the controller also advances the removal index. If the controller has to advance the removal index, this means that the host is not removing data fast enough, and capture data was lost by the host. The capture data is available in the shared VME memory at offset addresses 0xD94 through 0xFBB. The format of the capture table data in shared VME memory is defined in table 1-5 below.

TABLE 3-19 Real-Time Position Capture VME Shared Memory (Word Access Offset 0xD94)

Byte Offset	Byte Offset (Hex)	Byte length	Description
3476	0xD94	1	Controller insert index
3477	0xD95	1	Host removal index
3478	0xD96	550	Table entries (10 bytes per entry, and 55 entries)

The number of entries can be greater than one for each axis, if capture events occur on back to back motor update cycles, and if the host does not collect the data fast enough. The format of each table entry is defined in table 1-6 below.

TABLE 3-20 Real-Time Position Capture Table Entry

Byte Offset	Byte Offset (Hex)	Byte length	Description
0	0x00	4	Encoder position, offset 0x00 contains MSBs, offset 0x03 contains LSBs
4	0x04	1	Axis (X = 0, Y = 1, etc.)
5	0x05	1	Home event bits: 0x01 = Positive edge home switch 0x02 = Encoder home event 0x04 = Negative edge home switch
6	0x06	2	Positive edge I/O bits, offset 0x06 contains MSBs, and offset 0x07 contains LSBs
8	0x08	2	Negative edge I/O bits, offset 0x08 contains MSBs, and offset 0x09 contains LSBs

A value of 1 for a given bit indicates that it triggered the capture event. A value of 0 for a given bit means it did not trigger the capture event.

The motion controller contains a PowerPC processor, which writes the data in the shared memory in big endian format. If the host processor is not a big endian processor, then appropriate byte swapping to correct for endian differences must be performed by the host processor when accessing the shared memory data.

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 LT command on an axis by axis basis.

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).

4.3. CONTROL OUTPUT

The MAXv is configured at the factory to control open loop stepper motors. Upon installation, each axis can be configured for servo motors, open loop steppers, stepper motors with encoder feedback or a combination thereof. The servo output may be either unipolar analog (0/+10V) or bipolar analog (-10/+10 V).

Step pulse, direction and auxiliary outputs are open-collector TTL level signals which will wire directly into most driver inputs.

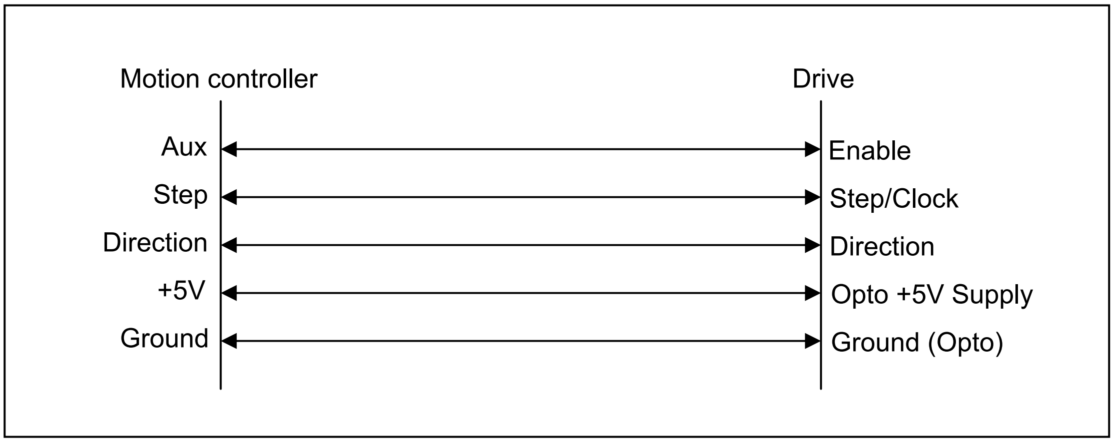


FIGURE 4-1 CONNECTION TO STEP DRIVES WITH INTERNAL PULL-UP RESISTORS AND OPTO-ISOLATION

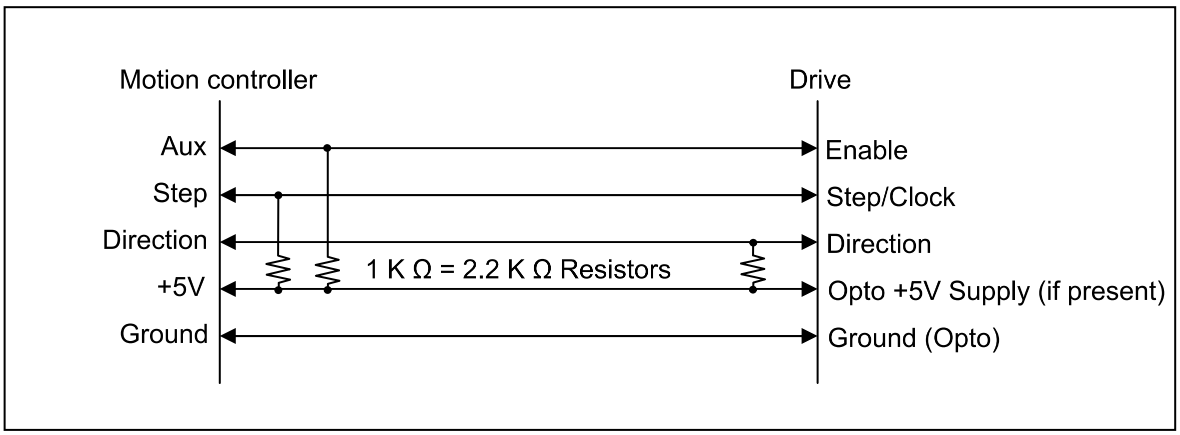


FIGURE 4-2 CONNECTION TO STEP DRIVES WITHOUT PULL-UP RESISTORS

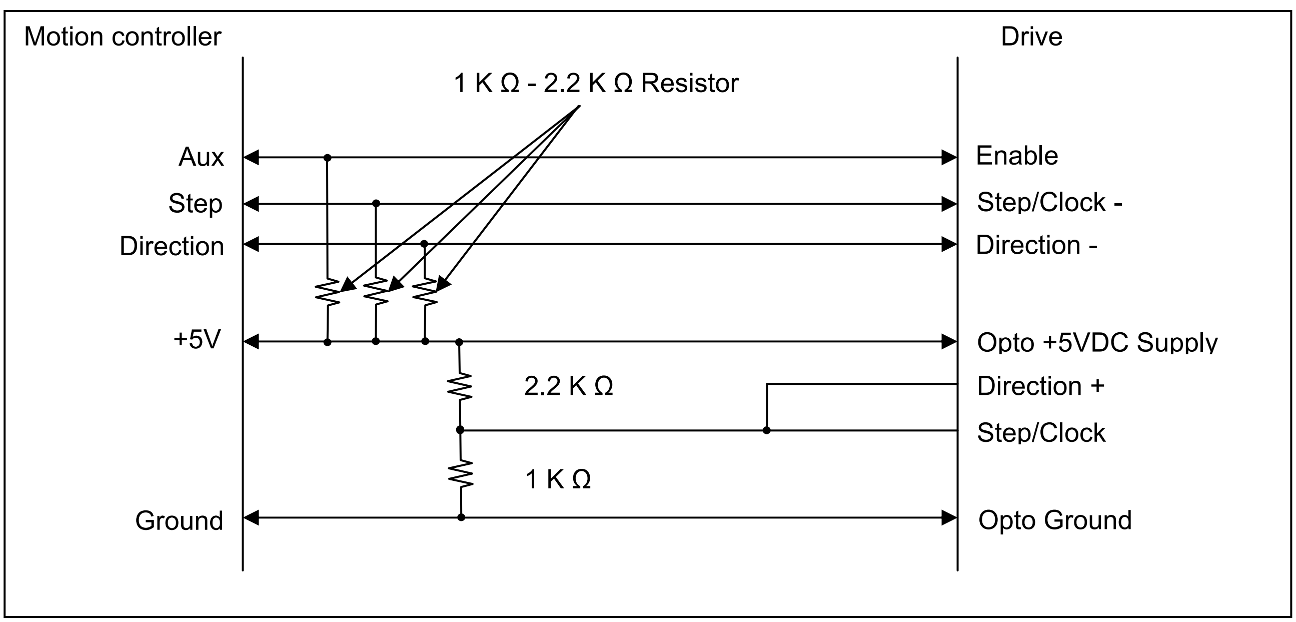


FIGURE 4-3 CONNECTION TO STEP DRIVES WITH DIFFERENTIAL

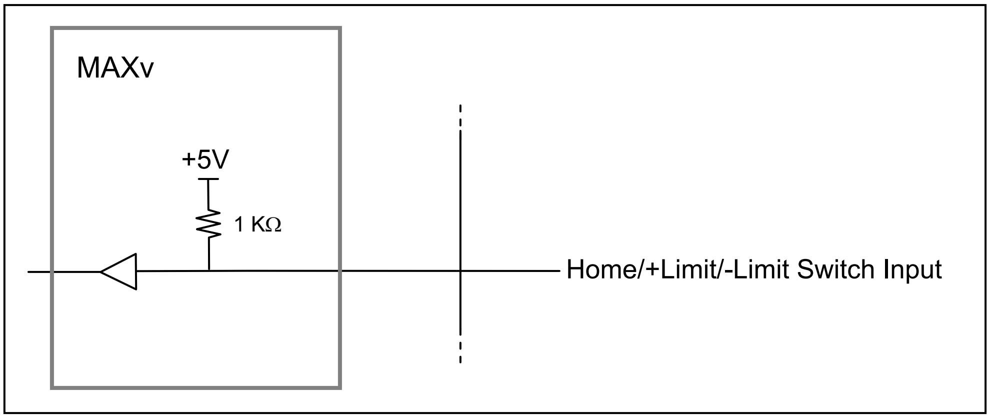


FIGURE 4-4 INPUT WIRING DIAGRAM

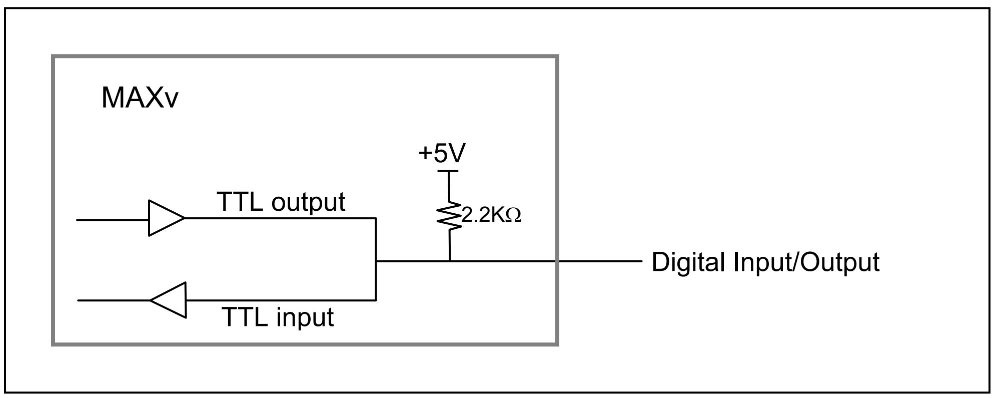


FIGURE 4-5 I/O WIRING DIAGRAM

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.

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 1k 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 Figure 4-6).

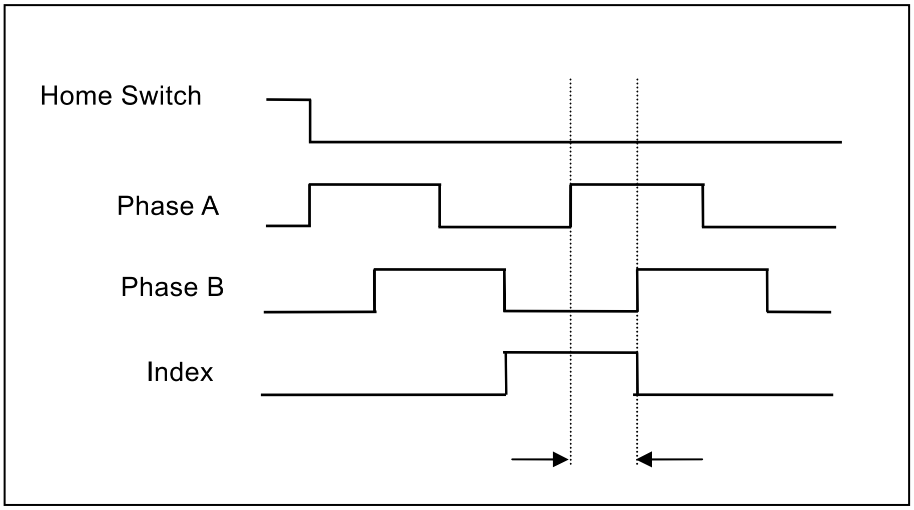


FIGURE 4-6 Encoder Homing State Detection

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. The use of auxiliary encoder channel 8 requires that general purpose input lines 0, 1, and 2 be configured as inputs. The use of auxiliary encoder channel 9 requires that general purpose input lines 4, 5, and 6 be configured as inputs.

Caution

Configure the appropriate general purpose I/O channels as inputs PRIOR to connecting encoder devices to these signals. Not doing so risks damage to the encoder device and/or the MAXv controller.

4.8. ABSOLUTE ENCODERS WITH SSI

The MAXv comes with two axes of configurable absolute encoders with SSI (Synchronous Serial Interface) technology. By default the X and Y axes will have up to 12 bits of resolution of absolute encoding. The MAXv can have up to 8 axes of absolute encoders and up to 32 bits of resolution per axis. The MAXv provides a differential clock output through the I/0 port to deliver clocking to an absolute encoder. The clocking can be configured for the following frequencies: 31,250Hz, 62,500Hz, 125,000Hz, 250,000HGz, 500,000Hz, 1MHz, 2MHz, and 4MHz.

With I/O 0-7 available, typical use of absolute encoders require that clock+ and clock- be configured from the X-Axis through I/O 0-1, Y-Axis through I/O 2-3, Z-Axis through I/O 4-5, and T-Axis through I/O 6-7. For axes U,V,R, and S, clocking would also be configured from clock signals through I/O 0-7. This requires that the clocking be shared between axes if more than four axes of absolute encoding are needed. Absolute encoders sharing the same I/O output clocks have the requirement that the clock frequency is the same and the bits resolution is the same.

4.8.1. CONFIGURATION EXAMPLES

The following are two examples on how to configure the MAXv for absolute encoding. The first case is standard MAXv with two absolute encoders with up to 12 bits resolution. For this example, the X axis is 12 bits resolution with a clock frequency at 125,000Hz, and the Y axis is 9 bits resolution with a clock frequency of 250,000Hz.

```
AX;
PSE;
ECA12,125000;
AY;
PSE;
ECA9,250000;
```

The second example calls for eight absolute encoders, three axes at 16 bits resolution with a clock frequency of 125,000Hz, three axes at 24 bits resolution with a clock frequency of 500,000Hz, and two axes at 32 bits resolution at 250000Hz. This example also shows the use of clock sharing with other absolute encoders with the same clock frequency and bits resolution.

```
AX;
PSE;
ECA16,125000;
AY;
PSE;
ECA16,125000;
AZ;
PSE;
ECA24,500000;
AT;
PSE;
ECA32,250000;
AU;
PSE:
ECA16,125000;     Shares clocks with either X axis  (I/O 0-1)  or Y axis (I/O 2-3)
AV;
PSE:
ECA24,500000;     Shares clocks with Z axis (I/O 4-5)
```

AR;
PSE:
ECA24,500000; Shares clocks with Z axis (I/O 4-5)
AS;
PSE;
ECA32,250000; Shares clocks with T axis (I/O 6-7)

Below is an example of how the absolute encoder can be connected to the MAXv through the front connectors of the board. This utilizes the IOvMAX breakout board for easier connectivity to the absolute encoder environment. Similarly, absolute encoding can be connected through the VME backplane. Table 3-1 provides address mappings for absolute encoder data.

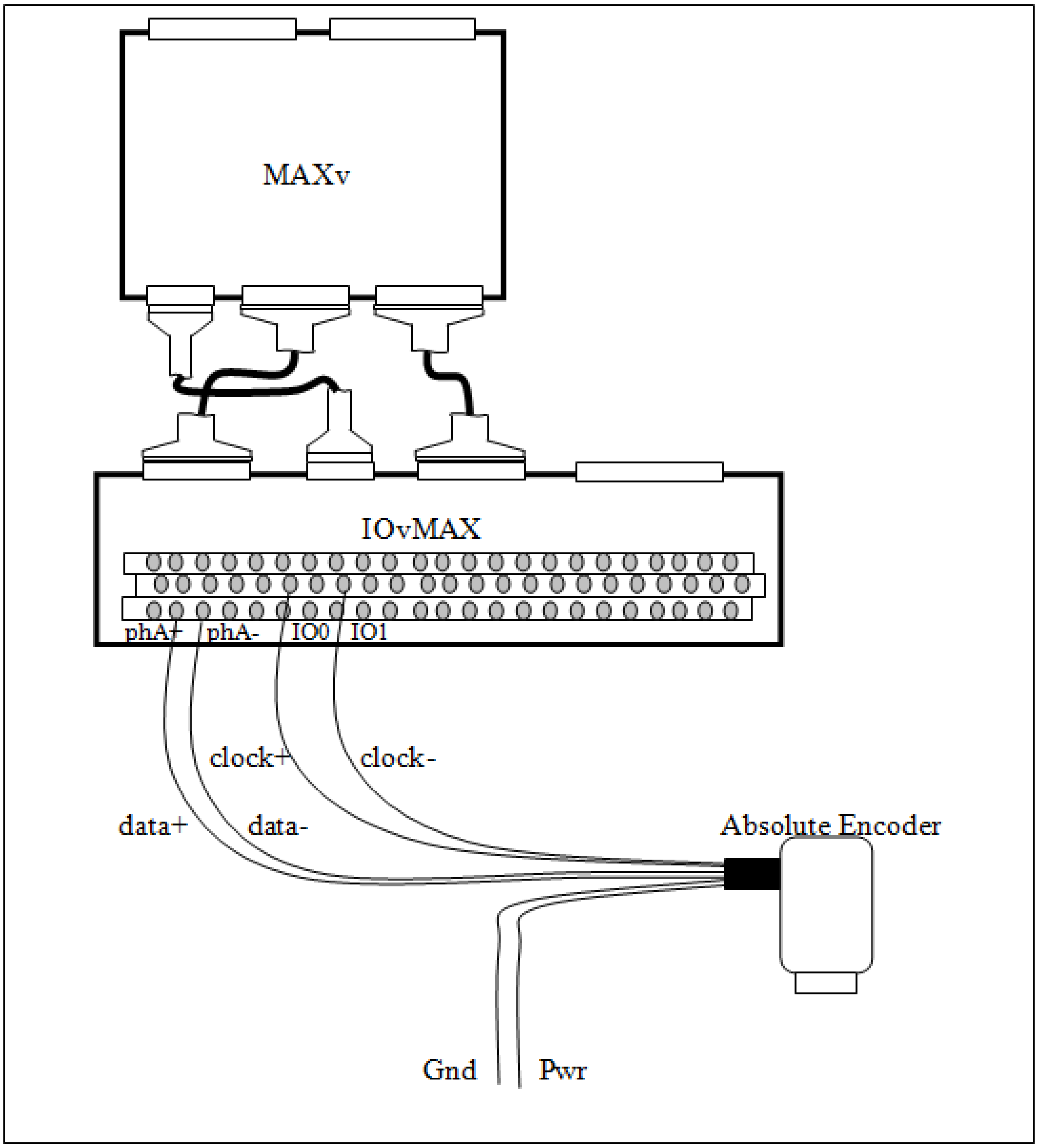


Figure 4-7 Connection of an Absolute Encoder to the MAXv Via the Front Panel and IOvMAX Breakout Board.

4.9. FRONT PANEL CONNECTORS

Table 4-1 Pin Out For Front Panel Connectors

50-Pin General Purpose I/O (J3)

Pin	FUNCTION	Pin	FUNCTION
1	Analog Input 0	26	Analog Input 1
2	GND	27	GND
3	Analog Input 2	28	Analog Input 3
4	GND	29	GND
5	Analog Input 4	30	Analog Input 5
6	GND	31	GND
7	Analog Output 0	32	Analog Output 1
8	GND	33	GND
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

* S Aux, see #UC command in MAX Command Manual

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	GND	40	GND
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	GND	58	GND
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	GND	40	GND
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	GND	58	GND
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

Legend

X axis	V axis
Y axis	R axis
Z axis	S axis
T axis	Ground
U axis	Voltage

CLICK HERE FOR BLACK AND WHITE TABLE VERSION OF THIS TABLE.

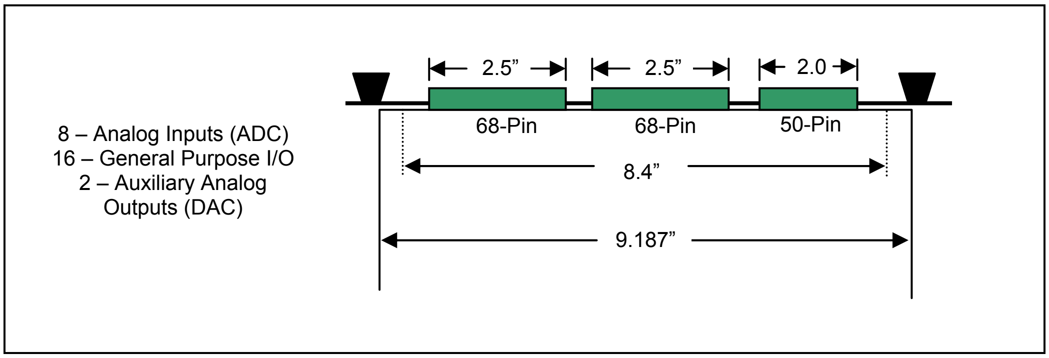


Figure 4-8 MAXv Front Panel Connector Pin Assignment

4.9.1. 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…)

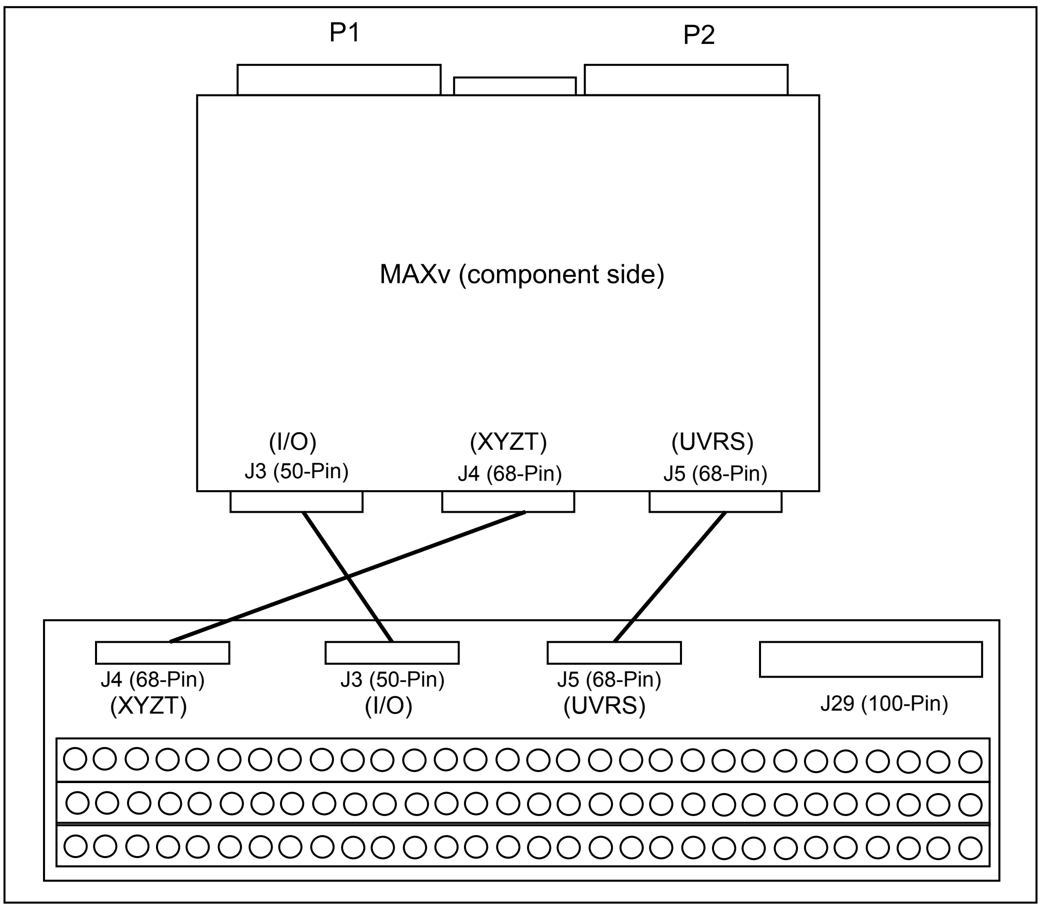


Figure 4-9 IOvMAX Break-Out to MAXv

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 4-2 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\X Servo	GND	X Phase A +	X Phase A -
3	Analog Input 0\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\Y Servo	A26	Y Phase A +	Y Phase A -
7	Analog Input 1\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\Z Servo	A30	Z Phase A +	Z Phase A -
11	Analog Input 2\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\T Servo	D16	T Phase A +	T Phase A -
15	Analog Input 3\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\U Servo	D20	U Phase A +	U Phase A -
19	Analog Input 4\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\V Servo	GND	V Phase A +	V Phase A -
23	Analog Input 5\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\R Servo	D27	R Phase A +	R Phase A -
27	Analog Output 0\IO14	R Pos LMT	D28	R Dir	R Aux
28	GND	R Neg LMT	D29	R Home	S Aux* / I/O6
29	S Phase B -	S Phase B	D30	S Index +	S Index -
30	GND	S Step\S Servo	D31	S Phase A +	S Phase A -
31	Analog Output 1\IO15	S Pos LMT	GND	S Dir	NC
32	GND	S Neg LMT	+5V	S Home	NC

* = See #UC command in MAX Family Command Reference Manual for selecting signal on this pin.

Legend	
X axis	V axis
Y axis	R axis
Z axis	S axis
T axis	Ground
U axis	Voltage

CLICK HERE FOR BLACK AND WHITE TABLE VERSION OF THIS TABLE.

4.10. 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-).

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 4-3 IOvMAX Terminal Block Pin-Out

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin #	Signal Name
1	Bias (1.2V)	37	V Phase A+	73	I/O 2\In8+	109	GND	145	T Step
2	X Phase A+	38	V Phase A-	74	GND	110	I/O 13	146	T Dir
3	X Phase A-	39	V Index A+	75	I/O 3	111	S Pos Limit	147	T Aux
4	X Index A+	40	V Index A-	76	Z Pos Limit	112	S Neg Limit	148	T Home
5	X Index A-	41	V Phase B+	77	Z Neg Limit	113	GND	149	5V
6	X Phase B+	42	V Phase B-	78	GND	114	Analog Output 1	150	U Servo
7	X Phase B-	43	GND	79	Analog Input 2	115	I/O 14	151	GND
8	GND	44	R Phase A+	80	I/O 4\A9+	116	GND	152	U Step
9	Y Phase A+	45	R Phase A-	81	GND	117	I/O 15	153	U Dir
10	Y Phase A-	46	R Index A+	82	I/O 5\B9+	118	indx_n8	154	U Aux
11	Y Index A+	47	R Index A-	83	T Pos Limit	119	GND	155	U Home
12	Y Index A-	48	R Phase B+	84	T Neg Limit	120	indx_n9	156	5V
13	Y Phase B+	49	R Phase B-	85	GND	121	5V	157	V Servo
14	Y Phase B-	50	GND	86	Analog Input 3	122	X Servo	158	GND
15	GND	51	S Phase A+	87	*S Aux\I/O 6\In9+	123	GND	159	V Step
16	Z Phase A+	52	S Phase A-	88	GND	124	X Step	160	V Dir
17	Z Phase A-	53	S Index A+	89	I/O 7	125	X Dir	161	V Aux
18	Z Index A+	54	S Index A-	90	U Pos Limit	126	X Aux	162	V Home
19	Z Index A-	55	S Phase B+	91	U Neg Limit	127	X Home	163	5V
20	Z Phase B+	56	S Phase B-	92	GND	128	5V	164	R Servo
21	Z Phase B-	57	GND	93	Analog Input 4	129	Y Servo	165	GND
22	GND	58	pha_n8	94	I/O 8	130	GND	166	R Step
23	T Phase A+	59	GND	95	GND	131	Y Step	167	R Dir
24	T Phase A-	60	phb_n8	96	I/O 9	132	Y Dir	168	R Aux
25	T Index A+	61	GND	97	V Pos Limit	133	Y Aux	169	R Home
26	T Index A-	62	X Pos Limit	98	V Neg Limit	134	Y Home	170	5V
27	T Phase B+	63	X Neg Limit	99	GND	135	GND	171	S Servo
28	T Phase B-	64	GND	100	Analog Input 5	136	Z Servo	172	GND
29	GND	65	Analog Input 0	101	I/O 10	137	GND	173	S Step
30	U Phase A+	66	I/O 0\A8+	102	GND	138	Z Step	174	S Dir
31	U Phase A-	67	GND	103	I/O 11	139	Z Dir	175	S Aux
32	U Index A+	68	I/O 1\B8+	104	R Pos Limit	140	Z Aux	176	S Home
33	U Index A-	69	Y Pos Limit	105	R Neg Limit	141	Z Home	177	5V
34	U Phase B+	70	Y Neg Limit	106	GND	142	5V	178	pha_n9
35	U Phase B-	71	GND	107	Analog Output 0	143	T Servo	179	GND
36	GND	72	Analog Input 1	108	I/O 12	144	GND	180	phb_n9

*See #UC command in MAX Command Manual

Legend	
X axis	V axis
Y axis	R axis
Z axis	S axis
T axis	Ground
U axis	Voltage

NOTE: Aux Encoder 'A' = 8 and Aux Encoder 'B' = 9 See Table 4-1 for pin-outs of J3(I/O), J4(XYZT), and J5(UVRS) of the MAXv.

CLICK HERE FOR BLACK AND WHITE TABLE VERSION OF THIS TABLE.

TABLE 4-4 VME58 OUTPUT CONNECTOR PIN LIST (J29) IOVMAX

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

Legend	
X axis	V axis
Y axis	R axis
Z axis	S axis
T axis	Ground
U axis	Voltage

CLICK HERE FOR BLACK AND WHITE TABLE VERSION OF THIS TABLE.

This page is intentionally left blank

5. HOST SOFTWARE

5.1. 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

6. SERVICE

6.1. 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).

6.2. 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 OMS Motion, Inc. 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

OMS Motion, Inc. can be reached for technical support by any of the following methods:

1. Internet E-Mail: mailto:support@omsmotion.com
2. World Wide Web: www.omsmotion.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: OMS Motion, Inc.
   15201 NW Greenbrier Parkway
   B-1 Ridgeview
   Beaverton, OR 97006

RETURN FOR REPAIRS

Call OMS Motion, Inc. Customer Service at 503-629-8081 or (800) 707-8111 or e-mail to sales@omsmotion.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:

OMS Motion, Inc.
15201 NW Greenbrier Parkway
B-1 Ridgeview
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 I/O 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.

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 velocity override, 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 15 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 2559 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;
CL0;
HM0;
MA10000;
```

GO;
ID;

In a move requiring a three axis coordinated move to a position, the following could be used:

```
AX;
KP2;
KD6;
CL0;
AY;
KP2;
KD6;
CL0;
AZ;
KP2;
KD6;
CL0;
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:

```
VO1000;
VC,,0,0;
ABH;
VC0,5000,3.1415926;
VC0,0,6.2831853;
ABL;
MT-10,10000;
VE;
MT,,-1000,0;
GO;
VC;
```

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

± 2,147,487,647

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 3-15 VDC Input sense (low or high true) selectable by command input for each axis.
Signal component = DS14C89AM (or equivalent)

CONNECTOR

Two shielded 68-Pin SCSI3 connectors for all motor control, 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

Input levels 3-15 VDC Input sense (low or high true) selectable by command input for each axis.
Signal component = DS14C89AM (or equivalent)

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. NOTE: 6 general purpose inputs are shared with two auxiliary encoder channels. Input signal component = DS26LV32AT (or equivalent for IO 0-7)
and 74LCX244MTC (or equivalent for IO 8-15).
Output signal component = SN74ABT125PWR with 51Ω series resistor (IO 0-7) and PI5C3125 to GND (IO 8-15).

ANALOG INPUTS
Six independent analog inputs, 16 Bit resolution
Signal component = ADS7809 (or equivalent).

ANALOG OUTPUTS (SERVO)
+/-10V and 0 to +10V, max. One per axis plus two general purposes. Signal component = AD621 (or equivalent).

STEP PULSE OUTPUT
Pulse width 50% duty cycle. Open collector level signal (TTL). Signal component = PI5C3125 (or equivalent).

DIRECTION OUTPUT
Open collector level signal (TTL). Signal component = PI5C3125 (or equivalent).

ENCODER FEEDBACK
Maximum 16 MHz after 4x quadrature detection. Differential signal. . Differential signal component = DS26LV32AT (or equivalent).

ABSOLUTE ENCODERS
SSI Technology
X and Y axis up to 12 bits resolution. (default)
Maximum 8 axes of absolute encoders up to 32 bits resolution.

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.omsmotion.com)

TABLE 6-1 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\X Servo	GND	X Phase A +	X Phase A -
3	Analog Input 0\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\Y Servo	A26	Y Phase A +	Y Phase A -
7	Analog Input 1\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\Z Servo	A30	Z Phase A +	Z Phase A -
11	Analog Input 2/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\T Servo	D16	T Phase A +	T Phase A -
15	Analog Input 3\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\U Servo	D20	U Phase A +	U Phase A -
19	Analog Input 4\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\V Servo	GND	V Phase A +	V Phase A -
23	Analog Input 5\1013	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\R Servo	D27	R Phase A +	R Phase A -
27	Analog Output 0\IO14	R Pos LMT	D28	R Dir	R Aux
28	GND	R Neg LMT	D29	R Home	S Aux* I/O6
29	S Phase B -	S Phase B	D30	S Index +	S Index -
30	GND	S Step\S Servo	D31	S Phase A +	S Phase A -
31	Analog Output 1\IO15	S Pos LMT	GND	S Dir	NC
32	GND	S Neg LMT	+5V	S Home	NC

* = See #UC command in MAX Family Command Reference Manual for selecting signal on this pin.

TABLE 6-2 IOvMAX Terminal Block Pin-Out

Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name	Pin	Signal Name
1	Bias (1.2V)	37	V Phase A+	73	I/O 2\In8+	109	GND	145	T Step
2	X Phase A+	38	V Phase A-	74	GND	110	I/O 13	146	T Dir
3	X Phase A-	39	V Index A+	75	I/O 3	111	S Pos Limit	147	T Aux
4	X Index A+	40	V Index A-	76	Z Pos Limit	112	S Neg Limit	148	T Home
5	X Index A-	41	V Phase B+	77	Z Neg Limit	113	GND	149	**5V**
6	X Phase B+	42	V Phase B-	78	GND	114	Analog Output 1	150	U Servo
7	X Phase B-	43	GND	79	Analog Input 2	115	I/O 14	151	GND
8	GND	44	R Phase A+	80	I/O 4\A9+	116	GND	152	U Step
9	Y Phase A+	45	R Phase A-	81	GND	117	I/O 15	153	U Dir
10	Y Phase A-	46	R Index A+	82	I/O 5\B9+	118	indx_n8	154	U Aux
11	Y Index A+	47	R Index A-	83	T Pos Limit	119	GND	155	U Home
12	Y Index A-	48	R Phase B+	84	T Neg Limit	120	indx_n9	156	**5V**
13	Y Phase B+	49	R Phase B-	85	GND	121	**5V**	157	V Servo
14	Y Phase B-	50	GND	86	Analog Input 3	122	X Servo	158	GND
15	GND	51	S Phase A+	87	*S Aux\I/O 6\In9+	123	GND	159	V Step
16	Z Phase A+	52	S Phase A-	88	GND	124	X Step	160	V Dir
17	Z Phase A-	53	S Index A+	89	I/O 7	125	X Dir	161	V Aux
18	Z Index A+	54	S Index A-	90	U Pos Limit	126	X Aux	162	V Home
19	Z Index A-	55	S Phase B+	91	U Neg Limit	127	X Home	163	**5V**
20	Z Phase B+	56	S Phase B-	92	GND	128	**5V**	164	R Servo
21	Z Phase B-	57	GND	93	Analog Input 4	129	Y Servo	165	GND
22	GND	58	pha_n8	94	I/O 8	130	GND	166	R Step
23	T Phase A+	59	GND	95	GND	131	Y Step	167	R Dir
24	T Phase A-	60	phb_n8	96	I/O 9	132	Y Dir	168	R Aux
25	T Index A+	61	GND	97	V Pos Limit	133	Y Aux	169	R Home
26	T Index A-	62	X Pos Limit	98	V Neg Limit	134	Y Home	170	**5V**
27	T Phase B+	63	X Neg Limit	99	GND	135	GND	171	S Servo
28	T Phase B-	64	GND	100	Analog Input 5	136	Z Servo	172	GND
29	GND	65	Analog Input 0	101	I/O 10	137	GND	173	S Step
30	U Phase A+	66	I/O 0\A8+	102	GND	138	Z Step	174	S Dir
31	U Phase A-	67	GND	103	I/O 11	139	Z Dir	175	S Aux
32	U Index A+	68	I/O 1\B8+	104	R Pos Limit	140	Z Aux	176	S Home
33	U Index A-	69	Y Pos Limit	105	R Neg Limit	141	Z Home	177	**5V**
34	U Phase B+	70	Y Neg Limit	106	GND	142	**5V**	178	pha_n9
35	U Phase B-	71	GND	107	Analog Output 0	143	T Servo	179	GND
36	GND	72	Analog Input 1	108	I/O 12	144	GND	180	phb_n9

* See #UC in MAX Command Manual

NOTE: Aux Encoder ‘A’ = 8 and Aux Encoder ‘B’ = 9 See Table 6-3 for pin-outs of J3(I/O), J4(XYZT), and J5(UVRS) of the MAXv.

TABLE 6-3 PIN OUT FOR FRONT PANEL CONNECTORS

50-Pin General Purpose I/O (J3)

Pin	FUNCTION	Pin	FUNCTION
1	Analog Input 0	26	Analog Input 1
2	GND	27	GND
3	Analog Input 2	28	Analog Input 3
4	GND	29	GND
5	Analog Input 4	30	Analog Input 5
6	GND	31	GND
7	Analog Output 0	32	Analog Output 1
8	GND	33	GND
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

* S Aux, see #UC command in MAX Command Manual

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	GND	40	GND
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	GND	58	GND
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	GND	40	GND
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	GND	58	GND
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

TABLE 6-4 VME58 Output Connector Pin List (J29) IOvMAX

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

ORDERING INFORMATION

Model	Computer Interface	Axes	Servo/Stepper	User I/O					
				Limit	Auxiliary	Home	Digital General Purpose	Analog	
								In	Out
MAXv-1000	VME Bus	1	User Definable	2	1	1	16	6	3
MAXv-2000		2	User Definable	4	2	2	16	6	4
MAXv-3000		3	User Definable	6	3	3	16	6	5
MAXv-4000		4	User Definable	8	4	4	16	6	6
MAXv-5000		5	User Definable	10	5	5	16	6	7
MAXv-6000		6	User Definable	12	6	6	16	6	8
MAXv-7000		7	User Definable	14	7	7	16	6	9
MAXv-8000		8	User Definable	16	8	8	16	6	10

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

INDEX

S

U

V

W