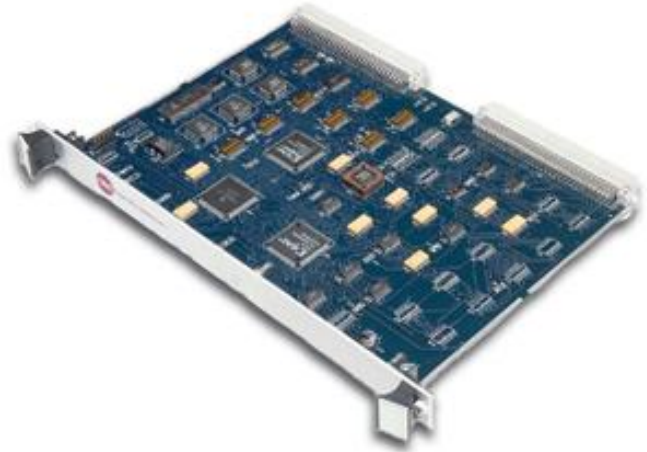




**"The Company in Motion"™**



## VX2

### Intelligent Motion Controller for VME Bus Computers FEATURES

- **Controller capability**
  - Up to six axes of stepper control
  - Up to two axes of encoder feedback
- **Communications**
  - VME bus
  - Interrupt or Polling communication
  - 5 I/O registers for control & status
  - Meets VME bus specification C.1 for D08(O) in short address space
  - Standard address space is also available
- **Sophisticated Control Functionality**
  - Independent and coordinated motion of all axes at the same time
  - Slip & Stall detection with encoder feedback
  - Crystal controlled step pulse from 0 to 522,000 steps per second
  - Circular Interpolation
  - Constant velocity linear interpolation (all axes)
  - Electronic Gearing
- **32 bit processor for extensive co-processing**
  - Does not burden the host with overhead
  - Parabolic, "S"-curve & Linear trajectory profiles
  - Patented technology for minimized torque ripple and velocity modulation
  - Internal Watchdog timer for safety
- **Control signals**
  - 96-pin VME connector
  - Up to 8 user I/O
  - Output is Step & Direction
  - Independent home and plus / minus over-travel inputs
- **Programming**
  - High level programming expertise not required
  - Commands are ASCII characters
  - Automatic conversion to "user" defined units i.e. inches / revolutions
  - Compatible with most operating systems
- **Flash Memory**
  - Non-volatile program storage
  - Non-volatile parameter storage
- **Factory Direct Technical Support**
  - Person to person toll-free tech support, call 800-707-8111
  - Example programs provided
  - All OMS controls are 100% burned-in, tested and quality inspected

## DESCRIPTION

The VX2 intelligent motion control allows control of up to 6 axes of motion control in one slot of a VME bus computer. All axes are controlled through 5 VME short address ports for control, status feedback, data, commands and the VME bus interrupt vector. Each axis has a separate command queue allowing the host computer to transfer a command string then proceed with other tasks, while the VX2 manages the motion process. The computer can be interrupted at any point in the command stream to coordinate the motion process with other activities. Each axis can perform individual unrelated moves or they can be coordinated as required by the application.

The VX2 generates step and direction pulses for control of the most popular step motor drivers. It also supports servo or linear motor controllers which accept step and direction inputs. The VX2 supports high resolution microstepping of 50,000 steps per revolution with a standard 200 step per revolution (1.8 degree per step) stepping motor by developing the high pulse rates required for these applications. This high resolution allows the stepping motor to run smoothly at all speeds and minimizes low speed torque loss due to mechanical resonance effects. The velocity streaming mode allows an arbitrary move contour under control of the VME bus computer. Constant velocity contouring, with linear interpolation on up to six axes and circular interpolation on any two axes, provides constant tool velocity for machining applications.

Simple ASCII commands can be easily sent to the board from any high level language, which allows input and output to an I/O device, for example Basic, Pascal and C. Up to two sync lines are available to synchronize moves to external events and an auxiliary line is associated with each axis, which allows the user to control external events. Complex move sequences, time delays, status checks and control of other external events can be programmed through the VME bus interface.

Incremental encoder feedback is available as an option on up to two axes for those applications requiring precise position feedback and/or correction. The encoder option can correct for position errors, monitor for slip or stall or allow tracking of one motor with another.

## PROGRAMMING

The OMS motion controls are easily programmed with double character ASCII commands through an extensive command structure. These commands are combined into character strings to create sophisticated motion profiles. It includes an 800 command and parameter buffer for each axis and a command loop counter which allows multiple executions of most command strings.

The following commands are available in the VX2 family of motion controls. Some of the commands expect a numerical operand to follow. These commands are identified with a # after the command. The # indicates a signed integer input parameter or a signed fixed point number of the format ##.# when user units are enabled. With user units defined, distances, velocity and acceleration parameters may be input in inches, revolutions, etc.

Synchronized moves may be made by entering the AA command. This command performs a context switch which allows entering commands of the format MRx#, y#, z#, t#, u#, v#;. Numbers are entered for each axis which is to be commanded to move. An axis may be skipped by entering the comma with no parameter. The command may be prematurely terminated with a ";", i.e. a move requiring only the X and Y axes would use the command MRx#, y#; followed by the GO command. Each axis programmed to move will start together upon executing the GO command. The VX2 can be switched back to the unsynchronized mode by entering the desired axis command such as AX.

The AM command mode is available for applications where the host software is multitasking and keeps track of each axis as a task or process. This allows sequenced commands on some axes including linear interpolation while simultaneously performing other unrelated moves on the other axes.

The constant velocity contouring mode allows specification of circular and linear interpolated moves at constant velocity for machining applications. The VX2 will interpret the input sequence and build a queue of moves to later generate a continuous path at constant velocity. The VX2 will generate contours of this type at full speed.

The following summarizes the VX2 command set:

### AXIS SPECIFIC COMMANDS

The following commands specify the axis to which the following commands are to be directed. They remain in effect until replaced by another command of the same type.

#### **AA AXIS ALL**

The AA command will perform a context switch to the synchronized mode.

#### **AM AXIS MULTITASKING**

Commands following the AM will go into the specified axis queues for independent synchronized mode. The AM mode allows several tasks to be managed simultaneously. For instance, a task may be performing coordinated motion on 2 axes, while a second task is performing unrelated but simultaneous motion on another axis.

#### **AX AXIS X**

The AX command directs all the following commands to the X axis.

#### **AY AXIS Y**

The AY command directs all the following commands to the Y axis.

**AZ    **AXIS Z****

The AZ command directs all the following commands to the Z axis.

**AT    **AXIS T****

The AT command directs all the following commands to the T axis.

**AU    **AXIS U****

The AU command directs all the following commands to the U axis.

**AV    **AXIS V****

The AV command directs all the following commands to the V axis.

**SYSTEM CONTROL COMMANDS**

These commands allow control of various system parameters and operating modes to allow the user to optimize the response of the system for his/her application needs.

**EN    **ECHO ON****

The EN command enables echoing from the VX2.

**EF    **ECHO OFF****

The EF command disables echoing from the VX2.

**HH    **HOME HIGH****

The HH command sets the sense of the home switch on the current axis to active high.

**HL    **HOME LOW****

The HL command sets the sense of the home switch on the current axis to active low.

**LF    **LIMITS OFF****

The LF command turns off the limit switch sensing for the addressed axis. This allows the stage to move beyond the limit switch and should be used with caution.

**LN    **LIMITS ON****

The LN command restores the operation of the limit switches for the addressed axis.

**SL    **SOFT LIMIT****

The SL command changes the operation of the limit inputs causing the output train to ramp down instead of terminating immediately. The output queue is not flushed except for the current move.

**SF    **SOFT LIMIT OFF****

The SF command restores the normal operation of the limit switches.

**CN    **COSINE ON****

The CN command enables cosine velocity ramps, i.e. half sinusoid acceleration profiles for all axes.

**PN#   **PARABOLIC ON****

The PN command enables parabolic ramps. The parameter selects the point of truncation.

**PF    **PARABOLIC OFF****

The PF command restores linear acceleration and deceleration ramps.

**RS    **RESET****

The RS command is a software reset which causes the VX2 microprocessor to reset. All programmable values are set to factory defaults. **USER I/O COMMANDS**

The following commands allow manipulation and testing of the user definable I/O.

**AN    **AUXILIARY ON****

The AN command sets the auxiliary output to the high level. The open collector driver is off allowing the output to be pulled high by a pull-up resistor. It may be used to change power level on driver modules so equipped or as a user specified output.

**AF    **AUXILIARY OFF****

The AF command sets the auxiliary output to the low level. The open collector driver is on causing the line to be near ground. It may be used to change power level on driver modules so equipped or as a user specified output.

**PA#   **POWER AUTOMATIC****

The PA command will perform an AN command at the beginning of each move and an AF command after the move. See AN and AF commands.

**SE#   **SETTLING TIME****

The SE command allows specification of a settling time, in milliseconds, to be used before the power is reduced, when using the PA mode.

**MOVE SPECIFICATION COMMANDS**

These commands allow specification of move parameters. They allow move parameters to be tailored to the users system requirements.

**AC#   **ACCELERATION****

The AC command sets the acceleration/deceleration register. This register is used to establish the rate of acceleration and deceleration when a move command is invoked.

**VL#   **VELOCITY****

The VL command sets the maximum velocity register of the axis being programmed. The value in this register is used to establish the maximum number of pulses per second sent to the motor driver from the VX2 board when one of the move execution commands is invoked.

**VB#   **VELOCITY BASE****

The VB command allows the velocity ramp to start at the specified velocity. This allows faster acceleration and the ability to pass through resonance quickly in some applications.

**LP#   **LOAD POSITION****

The LP command will immediately load the position supplied as a parameter into the absolute position register of the axis.

**MA#   **MOVE ABSOLUTE****

The MA command will set up the axis to move to the absolute position supplied as a parameter.

#### **MR# MOVE RELATIVE**

The MR command will set up the axis to move relative from the current position at the time the move is executed.

#### **ML## MOVE LINEAR**

The ML command uses linear interpolation to perform a straight line relative move to the new location. Up to 6 axes may be moved together in the AA mode.

#### **MT## MOVE TO**

The MT command uses linear interpolation to perform a straight line move to the specified absolute position. Up to 6 axes may be moved together in the AA mode.

#### **MO MOVE ONE PULSE**

The MO command will output one step pulse in the current direction without the latency of a ramp up to velocity. This command is not available in models with an encoder option.

#### **RM# REMAINDER**

The RM command will divide the position counter by the parameter supplied and replace the position counter with the resulting remainder as an absolute position. This command is useful in continuously rotating axis applications.

### **MOVE EXECUTION COMMANDS**

These commands allow execution of the moves which have been previously specified.

#### **GO GO**

The GO command will initiate the move which has been previously programmed with such commands as MA, MR, MT, and ML.

#### **GD GO AND RESET DONE FLAG**

The GD command resets the done flags on the active axes then proceeds with the move identical to the GO command.

#### **JG# JOG**

The JG command is a velocity command and will jog the axis at the velocity supplied as a parameter. The velocity may be changed without stopping by entering another JG command.

#### **JF## JOG FRACTIONAL VELOCITIES**

The JF command will jog the current axis at fractional rates.

#### **VS##,##VELOCITY STREAMING**

The VS command will generate a pulse train without acceleration or deceleration at the rates specified for the time specified by the first parameter.

### **MOVE TERMINATION COMMANDS**

The following commands allow termination of move sequences in process.

#### **ST STOP**

The ST command flushes the queue for the currently addressed axis only and causes the axis to decelerate to a stop at the rate previously specified in an AC command.

#### **SA STOP ALL**

The SA command flushes all queues and causes all axes to decelerate to a stop at the rate previously specified in an AC command.

#### **SD STOP AND RESET DONE**

The SD command will stop all axes and clear any done flags.

#### **KL KILL**

The KL command will flush the command queue and terminate pulse generation of all axes immediately.

### **LOOP CONTROL COMMANDS**

These commands allow move sequences to be repeated within loops. Loops can be nested up to four levels deep on each axis.

#### **LS# LOOP START**

The LS command sets the loop counter for the axis being programmed. The parameter specifies the number of times the loop will be executed. Loops may be nested up to 4 levels deep.

#### **LE LOOP END**

The LE command terminates the most recent LS command.

#### **WS# WHILE SYNC**

The WS command will execute the commands between the WS and WD commands as a loop, while the sync line is true, i.e. low. The test is at the bottom of the loop and thus will always be executed at least once.

#### **WD WHILE END**

The WD command serves as the loop terminator for the WS command.

#### **WH WHILE**

The WH command will execute all commands between it and the terminating WG command as a loop until terminated by a CW command. This allows indefinite loops to be terminated by the host computer.

#### **WG WHILE FLAG END**

The WG command serves as the terminator for the WH command.

#### **CW CLEAR WHILE**

The CW command terminates the WH command sequence upon execution of the next WG instruction, thus the loop is always executed at least once.

### **HOME AND INITIALIZATION CONTROL COMMANDS**

These commands allow the coordination of the physical stage home position with the VX2 position register.

#### **HM# HOME COMMAND**

The HM command will find home and initialize the position counter to the position supplied as a parameter.

#### **HR# HOME REVERSE**

The HR command will find home in the reverse direction and initialize the position counter to the position supplied as a parameter.

#### **KM HOME AND KILL**

The KM command will find home and stop generating pulses immediately, i.e. no deceleration ramp will be generated. The position counter is not affected.

#### **KR HOME REVERSE AND KILL**

The KR command will find home in reverse and stop generating pulses immediately, i.e. no deceleration ramp will be generated. The position counter is not affected.

### **MOVE SYNCHRONIZATION COMMANDS**

These commands allow the synchronization of moves with external events or multiple axes sequences.

#### **ID INTERRUPT DONE**

The ID command will return the done flag to the host and interrupt the host if the interrupt has been enabled.

#### **II INTERRUPT INDEPENDENT**

The II command allows each axis to return the done flag to the host when it completes its move independent of the status of the other axes.

#### **IN# INTERRUPT NEARLY DONE**

The IN command allows each axis to return the done flag to the host when the move is nearly complete. The parameter specifies the number of counts left in the move when the interrupt request is generated.

#### **IC INTERRUPT CLEAR**

The IC command will clear the done and error flags.

#### **CA CLEAR AXIS DONE FLAG**

The CA command operates like the IC command, except it clears the done flag for the addressed axis only.

#### **WA WAIT FOR AXES**

The WA command, only valid in the AA mode, allows a command to wait until all moves on all axes are finished before it executes.

#### **WQ WAIT FOR QUEUE TO EMPTY**

The WQ command is a special command that stops the board from processing any new command until the queue for the current axis mode is empty.

#### **SW# SYNC WAIT**

The SW command allows synchronization to external events through use of the sync lines. It causes the axis to wait for completion of other events, i.e. until all devices connected to the sync line have released it. The parameter specifies which sync line is used in boards configured for 4 or 6 axes.

#### **WT# WAIT TIME**

The WT command will wait for the specified number of milliseconds before proceeding with the next command.

### **SYSTEM STATUS REQUEST COMMANDS**

These commands allow the host to request the status of various move parameters, including the status of limit and home switches.

#### **WY WHO ARE YOU**

The WY command returns the model and firmware revision of the board being addressed.

#### **RP RETURN POSITION**

The RP command requests the current position.

#### **RQ RETURN QUEUE STATUS**

The RQ command returns the number of entries available in the command queue.

#### **RA RETURN AXIS INTERRUPT STATUS**

The RA command returns the state of the limit and home switches, and the done and direction flags for the currently addressed axis. The done flag is reset.

#### **RI RETURN INTERRUPT STATUS**

The RI command returns the state of the limit and home switches, and the done and direction flags for all axes. The done flags are reset.

#### **QA QUERY AXIS**

The QA command returns the status of the single addressed axis like the RA command, except the status register and flags are not affected.

#### **QI QUERY INTERRUPT STATUS**

The QI command returns the status of all axes like the RI command, except the status register and flags are not affected.

#### **RC REQUEST ACCELERATION**

The RC command will return the current acceleration or deceleration of the current axis.

#### **RV REQUEST VELOCITY**

The RV command will return the current velocity at which the axis is moving.

#### **RU REPORT POSITION IN USER UNITS**

The RU command returns the current position in user units.

### **CONSTANT VELOCITY CONTOURING COMMANDS**

The contouring command set allows the building of a command sequence which can later be executed at constant velocity for machine tool and other similar applications.

#### **AF#,# AUXILIARY OFF**

The AF command turns off any combination of auxiliary ports, when encountered in the command stream, allowing control of other peripherals such as a laser beam for machining.

**AN#,# AUXILIARY ON**

The AN command turns on any combination of auxiliary ports.

**CD#,#; CONTOUR DEFINE**

The CD command allows entry of a contour definition which will start at the position specified. Any two axes may be used in the contour mode.

**CE CONTOUR END**

The CE command ends the definition of the contour sequence, i.e. terminate the CD mode or ramp to a stop and exit when the contour is executed.

**CK CONTOUR END AND KILL**

The CK command ends the definition of the contour sequence and stops pulse generation immediately without deceleration.

**CR#,#,# CIRCULAR INTERPOLATION**

The CR command causes the two axes defined by the CD command to move in a circular pattern from the entry position. The parameters specify the center of the circle and distance to travel in radians.

**CV# CONTOUR VELOCITY**

The CV command allows the specification of the contouring velocity.

**CX CONTOUR EXECUTE**

The CX command causes the VX2 controller to execute the previously defined contour sequence.

**MT#,# MOVE TO**

The MT command causes the axes defined by the CD command to move to the specified absolute position using linear interpolation.

**RQ REQUEST QUEUE STATUS**

The RQ command returns the number of entries available in the contouring queue.

**USER UNIT COMMANDS**

The following commands allow specification of move parameters in user defined units. The OMS controls will automatically convert all move parameters to these units once they have been initialized.

**UU# USER UNITS**

The UU command converts all move velocities, distances, etc. to user specified units by multiplying by the parameter given in this command.

**UF USER UNITS OFF**

The UF command turns off user units and causes the VX2 board to use its default units.

**ENCODER COMMANDS**

The following are encoder support commands for use with VX2 boards with the encoder option only.

*The following are position maintenance control commands:*

**ER#,# ENCODER RATIO**

The ER command allows specification of encoder ratio by entering encoder counts followed by motor counts, for position maintenance mode.

**HV# HOLD VELOCITY**

The HV command specifies the maximum velocity used for position correction.

**HG# HOLD GAIN**

The HG command specifies the position hold gain parameter. The position error is multiplied by this gain factor in determining the velocity during a position correction.

**HD# HOLD DEADBAND**

The HD command specifies deadband counts for position hold. The VX2 will consider the control in position when the stage is within the specified parameter counts during position correction.

**HF HOLD OFF**

The HF command disables position hold, stall detection, and tracking modes.

**HN HOLD ON**

The HN command enables position correction after a move.

**IP INTERRUPT WHEN IN POSITION**

The IP command operates like the ID command, except the interrupt is deferred until the stage is within the specified deadband.

*The following commands control the slip or stall detection mode.*

**ES# ENCODER SLIP TOLERANCE**

The ES command parameter specifies tolerance before slip or stall is flagged in the status register.

**IS INTERRUPT ON SLIP**

The IS command will enable interrupts to the host when the position error during a move exceeds the parameter specified by an ES command. The interrupt will occur if the done interrupt has been enabled in the control register. A bit in the status register is also set to flag the source of the interrupt.

**RL RETURN SLIP STATUS**

The RL command returns the slip detection status of each axis. An S is returned if slip has occurred for that axis, or else an N is returned.

*The following command controls the tracking mode of the controls.*

**ET ENCODER TRACKING**

The ET command turns on the encoder tracking mode. The axis will track its encoder input, thus allowing one axis to follow the activity of another or a thumbwheel for manual positioning or the movement of another device that produces a signal compatible to the encoder inputs.

*The following commands control the home sequence when used with an encoder.*

## **HE HOME ENCODER**

The HE command enables the encoder index mode, i.e. home is defined as the logical AND of the encoder index, the external home enable and the encoder quadrant.

## **HS HOME SWITCH**

The HS command enables non-encoder (switch only) home mode.

*The following commands return status information about the encoder to the host.*

## **EA ENCODER STATUS**

The EA command returns the encoder status of the currently addressed axis.

## **RE REQUEST ENCODER POSITION**

The RE command returns the current encoder position of the currently addressed axis in encoder counts.

## **VELOCITY STAIRCASE COMMANDS**

The following commands describe the velocity staircase mode. This mode is useful in applications requiring a change in velocity at a prescribed position without stopping.

### **MP MOVE POSITIVE**

The MP command sets the direction logic to move in the positive direction.

### **MM MOVE MINUS**

The MM command sets the direction logic to move in the negative direction.

### **MV#,# MOVE VELOCITY**

The MV command causes the motor to run to the new absolute position at the specified velocity. A velocity staircase may be generated by queuing a sequence of MV commands.

### **SP# STOP AT POSITION**

The SP command will cause the axis to stop at the specified position.

### **FP# FORCE POSITION**

The FP command will flush the command queue and attempt to stop at the specified position.

## **PROGRAMMING EXAMPLES**

In a typical move requirement involving both X and Y axes the following will demonstrate the programming:

- Move a relative distance of 1,000,000 pulses on the X axis with a velocity of 400,000 pulses per second and a peak acceleration of 500,000 pulses per second per second.
- Wait for the X axis to finish its motion, then move the Y axis to an absolute position of 1,000,000 with the same velocity and acceleration parameters.

With these assumed conditions, the following is input from the host computer:

```
AA
VL400000,400000; AC500000,500000;
MR1000000; GO
MA,1000000; GO
```

In a move requiring position correction, the following demonstrates the setup sequence to generate a move and maintain position upon completion (available only with controls that include the encoder option):

- Set up the X axis move parameters of 100,000 steps per second and an acceleration rate of 500,000 steps per second per second.
- Set up the position correction maximum velocity of 50,000 steps per second, a deadband of 10 steps and a correction gain of 2,000, then enable position correction.
- Perform a 200,000 step move. The control will perform the 200,000 step move then maintain position within the 10 step deadband until commanded to a new position.

The following would be input from the host computer:

```
AX VL100000 AC500000
HV50000 HD10 HG2000 HN
MR200000 GO
```

The following demonstrates cutting a hole with a 10,000 count radius using constant velocity contouring and circular interpolation:

- The contouring velocity is set to 1000 pulses per second. A contour is defined beginning at coordinates 0,0 on the Z and T axes.
- Auxiliary output on the X axis is turned on, which could turn on the cutting torch or laser starting the cut at the center of the circle.
- A half circle is cut from the center to the outside of the hole positioning the cutting tool at the start of the hole.
- The hole is then cut, the torch turned off, the stage stopped and the contour definition completed.
- The stage is then positioned and the contour definition executed.

The following would be input from the host computer:

```
CV1000 CD,,0,0;
AN0; CR0,5000,3.1415926
CR0,0,6.2831853
AF0; MT-10,10000
CE
MT,,-1000,0; GO CX
```

In a move requiring a staircase of velocity with the breakpoints of velocity at absolute positions, the following commands would be used:

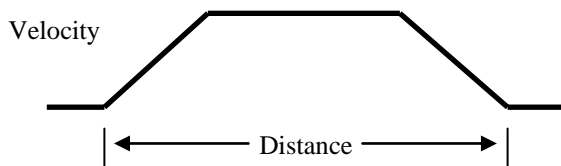
- Define a move in the positive direction.
- Accelerate to 30,000 pulses per second and move to position 10,000.
- Without stopping, accelerate to 50,000 pulses per second and move to absolute position 20,000.
- Without stopping, decelerate to 10,000 pulses per second and move to absolute position 30,000.
- Decelerate to a stop at position 35,000.

The following would be input from the host computer:

```
MP
MV10000,30000
MV20000,50000
MV30000,10000
SP35000
```

## OPERATING PRINCIPLE

The VX2 family of motion controls provides acceleration to an optimum speed followed by a constant velocity and a controlled deceleration to a stop. The actual position can be simultaneously monitored by use of the encoder feedback option. The velocity profile is achieved by calculating the optimum velocity 1024 times each second providing a very smooth acceleration curve. The calculation is used to control a variable frequency pulse train which is derived from a crystal oscillator thus providing very accurate pulse rates. Linear as well as parabolic and cosine velocity ramps can be generated to fit a variety of system requirements. The following figure shows a typical acceleration and constant velocity followed by the deceleration using a



Linear Velocity Profile

linear velocity profile.

Moves may be chained together to provide a more complex pattern. An internal queue can store up to 800 command parameters in an input queue for each axis as well as 124 characters in an input character buffer. This allows a complicated move sequence to be performed without host intervention. A flag can be passed to the host on the completion of a sequence or at any intermediate point in the command stream.

## SPECIFICATIONS

### **Velocity**

0 to 522,000 pulses per second simultaneous on each axis

### **Acceleration**

0 to 8,000,000 pulses per second per second

### **Position range**

134,000,000 pulses ( $\pm 67,000,000$ )

### **Accuracy**

Position accuracy and repeatability  $\pm 0$  counts for point to point moves

Velocity accuracy  $\pm 0.01\%$  of peak velocity in jog mode

### **Environmental**

Operating temperature range 0 to 50 degrees centigrade

Storage temperature range -20 to 85 degrees centigrade

Humidity 0 to 90% non-condensing

### **Power**

+5 volts at 1 amp typical (from VME bus backplane)

### **Dimensions**

10.3 x 7.4 x 0.8 inches high

### **VME bus interface**

Meets VME bus specification C.1 for D08(O) slaves in short address space. (Standard address space is also available on the VX2.)

### **Limit switch inputs**

Opto Isolated TTL input levels with on board 2.2K pull up resistor, requires only external switch closure to ground or TTL level input signal. Input sense (low or high true) selectable by on board switch for each axis.

### **Home switch inputs**

Opto Isolated TTL input levels with on board 2.2K pull up resistor, requires only external switch closure to ground or TTL level input signal. Input sense (low or high true) selectable under software control for each axis.

### **Sync switch inputs**

Opto Isolated TTL input levels with on board 2.2K pull up resistor, requires only external switch closure to ground or TTL level input signal.

### **Step pulse output**

Pulse width 50% duty cycle. Open collector TTL level signal.

### **Direction output**

Same as step pulse output

### **Auxiliary output**

Same as step pulse output

### **Encoder inputs**

Differential line receiver inputs

### **VME bus registers**

The VX2 uses D08(O) addressing within the short or standard address space. The 23 address block is user selectable. The factory default is FF80 hex.



<b>OFFSET</b>	<b>FUNCTION</b>	<b>I/O</b>
1	Data	I/O
3	Done Status	I
5	Control	I/O
7	Status	I/O
9	Vector	I
11	Unused	
13	Unused	
15	Unused	

## STEPPER PIN LIST

PIN	ROW A FUNCTION	ROW C FUNCTION
1	SYNCH 0	SYNCH 1
2	+5VDC	U Step
3	U Direction	U Auxiliary
4	Ground	U Negative Limit
5	U Positive Limit	U Home
6	+5VDC	V Step
7	V Direction	V Auxiliary
8	Ground	V Negative Limit
9	V Positive Limit	V Home
10	+5VDC	X Step
11	X Direction	X Auxiliary
12	Ground	X Negative Limit
13	X Positive Limit	X Home
14	+5VDC	Y Step
15	Y Direction	Y Auxiliary
16	Ground	Y Negative Limit
17	Y Positive Limit	Y Home
18	+5VDC	Z Step
19	Z Direction	Z Auxiliary
20	Ground	Z Negative Limit
21	Z Positive Limit	Z Home
22	+5VDC	T Step
23	T Direction	T Auxiliary
24	Ground	T Negative Limit
25	T Positive Limit	T Home
26	NC*	NC*
27	NC*	NC*
28	NC*	NC*
29	NC*	NC*
30	NC*	NC*
31	NC*	NC*
32	NC*	NC*

## STEPPER WITH ENCODER PIN LIST

PIN	ROW A FUNCTION	ROW C FUNCTION
1	SYNCH 0	SYNCH 1
2	+5VDC	X Index+
3	X Phase A+	X Phase A-
4	Ground	X Index-
5	X Phase B+	X Phase B-
6	+5VDC	Y Index+
7	Y Phase A+	Y Phase A-
8	Ground	Y Index-
9	Y Phase B+	Y Phase B-
10	+5VDC	X Step
11	X Direction	X Auxiliary
12	Ground	X Negative Limit
13	X Positive Limit	X Home
14	+5VDC	Y Step
15	Y Direction	Y Auxiliary
16	Ground	Y Negative Limit
17	Y Positive Limit	Y Home
18	+5VDC	Z Step
19	Z Direction	Z Auxiliary
20	Ground	Z Negative Limit
21	Z Positive Limit	Z Home
22	+5VDC	T Step
23	T Direction	T Auxiliary
24	Ground	T Negative Limit
25	T Positive Limit	T Home
26	NC*	NC*
27	NC*	NC*
28	NC*	NC*
29	NC*	NC*
30	NC*	NC*
31	NC*	NC*
32	NC*	NC*

\*No Connect



**"The Company in Motion"**™

15201 NW Greenbrier Pkwy  
B-1 Ridgeway  
Beaverton, Oregon 97006  
USA

(503) 629-8081, (800) 707-8111  
Fax: (503) 629-0688  
www.OMSmotion.com

## ORDERING INFORMATION

MODEL	STEPPER AXES	
	CLOSED LOOP (ENCODER)	OPEN LOOP
VX2-002		2
VX2-004		4
VX2-006		6
VX2-020	2	
VX2-022	2	2

Part Number: 3701-25000000

Revision B