VXM Program Stored Commands

Motor commands:

ImMx Set steps to incremental Index motor CW (positive), m= motor# (1,2,3,4), x=1 to 16,777,215
ImM-x Set steps to incremental Index motor CCW (negative), m= motor# (1,2,3,4), x=1 to 16,777,215
IAmMx Set Absolute Index distance, m=motor# (1,2,3,4), x= ±1 to ±16,777,215 steps
IAmM0 Index motor to Absolute zero position, m=motor# (1,2,3,4)
IAmM-0 Zero motor position for motor# m, m= 1,2,3,4
ImM0 Index motor until positive limit is encountered, m=motor# (1,2,3,4)
ImM-0 Index motor until negative limit is encountered, m=motor# (1,2,3,4)
(i3,i1...) Combine Index commands to run simultaneously on two VXM controllers connected by VXM bus
SmMx Set Speed of motor (70% power), m= motor# (1,2,3,4), x=1 to 6000 steps/sec.
SAmMx Set Speed of motor (100% power), m= motor# (1,2,3,4), x=1 to 6000 steps/sec.
SmM-x Read and assign analog input value to motor m speed (70% power), x= range
SAmM-x Read and assign analog input value to motor m speed (100% power), x= range
AmMx Acceleration/deceleration, m= motor# (1,2,3,4), x=1 to 127.

Looping/branching commands:

L0 Loop continually from the beginning or Loop-to-marker of the current program
LM0 Sets the Loop-to-marker at the current location in the program
LM-0 Resets the Loop-to-marker to the beginning of the current program
Lx Loop from beginning or Loop-to-marker x-1 times (x=2 to 65,535), when the loop reaches its last count the non-loop command directly preceding will be ignored
L-x Loop from beginning or Loop-to-marker x-1 times, alternating direction of motor 1, when the loop reaches its last count the non-loop command directly preceding will be ignored
LAx Loop Always from beginning or Loop-to-marker x-1 times (x=2 to 65,535)
LA-x Loop Always from beginning or Loop-to-marker x-1 times, alternating direction of motor 1
LM-2 Loop once from beginning or Loop-to-marker reversing index direction of motor 2
LM-3 Loop once from beginning or Loop-to-marker reversing index direction of motor 1 and motor 2
Jx Jump to the beginning of program number x, x= 0 to 4
JMx Jump to the beginning of program number x and come back for More after program x ends, x= 0 to 4
JM-x Similar to JMx except automatically moves back from absolute indexes after program x ends: For pick-and-place within matrix looping patterns

Pausing commands:

Px Pause x tenths of a second, (x=0 to 65,535)
P-x Pause x tenths of a millisecond, (x=1 to 65,535)
PAx Pause x tenths of a second (x=0 to 65,535, 10 μsec pause when x=0) Altering output 1 high for duration of the pause, tenths of a millisecond when x is negative
PA-x Pause x tenths of a millisecond (x=1 to 65,535) Altering output 1 high for duration of the pause

Input/output commands:

U0 Wait for a "low" on user input 1
U1 Wait for a low on user input 1, holding user output 1 high while waiting
U2 Enable Jog mode while waiting for an input
U3 Disable Jog mode while waiting for an input
U4 User output 1 "low" (reset state)
U5 User output 1 high
U6 Send "W" to host and wait for a "G" to continue
U7 Start of Continuous Index with pulse on output 2
U77 Start of Continuous Index with no output
U8 Start of Continuous Index sending "@" to the host
U9 End of Continuous Index with auto-decel to stop
U91 End of Continuous Index with auto-generate a deceleration Index as next command
U92 End of Continuous Index using next Index for deceleration to stop
U99 End of Continuous Index with instantaneous stop
U11 Skip next command if input 1 is high
U21 Skip next command if input 1 is low
U12 Skip next command if input 2 is high
U22 Skip next command if input 2 is low
U13 Wait for a front panel button to jump to a program or continue: "Motor 1 Jog -" button to jump to program #1, "Motor 1 Jog +" button to jump to program #2, "Run" button to proceed in current program.
U14 User output 2 low (reset state)
U15 User output 2 high
U16 Optional User output 3 low (reset state)
U17 Optional User output 3 high
U18 Optional User output 4 low (reset state)
U19 Optional User output 4 high
U23 Wait for a front panel button to jump to a program and come back, or continue: "Motor 1 Jog -" button to jump and return to program #1, "Motor 1 Jog +" button to jump and return to program #2, "Run" button to proceed in current program
U30 Wait for a low to high transition on user input 1
U31 Wait for a low to high transition on user input 1, holding user output 1 high while waiting
U32 Wait for "Motor 1 Jog -" button to be pressed on front panel with debouncing
U33 Wait for "Motor 1 Jog +" button to be pressed on front panel with debouncing
U50 Wait for a low and high on user input 1 with debouncing for a mechanical push-button switch
U51 Wait for a low and high on user input 1 with debouncing for a mechanical push-button switch, holding user output 1 high while waiting
U90 Wait for a low to high on the Run button or connection I/O,4 with debouncing for a mechanical push-button switch

VXM Set Commands

setMmMx Set axis m for motor type/size ***x*** (see table below for value for ***x***)
setDMx Set VXM/VP9000 or NF90 emulation modes, and other operating parameters

setDAx Set Analog Joystick Deadband value
setjmM Set first Jog Speed setting for motor m
setjAmM Set first Analog Joystick range setting for motor m
setJmM Set second Jog Speed setting for motor m
setJAmM Set second Analog Joystick range setting for motor m
setLmMx Set limit switch mode for axis m

setPmMx Set "Pulse Every x # Steps" on output 2 for axis m
setPAx Set Pulse width used by **setPmMx** and **U7**, x=1 to 255 (10 μsec increments)
setIx Set operating mode of inputs
setBx Set RS-232 Baud rate (9=9600, 19=19200, 38=38400)

Bx Backlash compensation, x=0 to 255, 0= off (default), 1= 20 steps
Ox Indicate limit switch Over-travel to host, off when x=0, VXM sends "O" when x=1 and hit limit, x=3 program stops too

PMAx Program Associate program x in Master to program x in Slave (Linked VXMs start the same time) (255= default/disabled)

VXM Immediate Commands

Operation commands:

C Clear all commands from currently selected program
D Decelerate to a stop (interrupts current index/ program in progress)
E Enable On-Line mode with echo "on"
F Enable On-Line mode with echo "off"
G Enable On-Line mode with echo off Grouping a <cr> with "^", ":", "W", "O" responses; Also Go after waiting or holding
H Put Controller on Hold (stop after each command and wait for go)
K Kill operation/program in progress and reset user outputs
N Null (zero) motors 1,2,3,4 absolute position registers
Q Quit On-Line mode (return to Local mode)
R Run currently selected program
! Record motor positions for later recall with "x","y" commands
rsm Run save memory (saves setup/ program values to nonvolatile memory)
res Software reset controller
del Delete last command
[i1,i2...] Send data to Slave through Master (two VXM controllers connected by VXM bus)
setD0 Set VXM back to factory defaults (All programs, settings, motor selections will be erased)
PMx Select Program number x, x= 0 to 4
PM-x Select and clear all commands from Program number x, x= 0 to 4

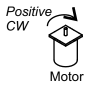
Status request commands:

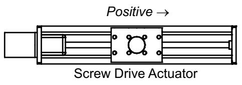
V Verify Controller's status, VXM sends "B" to host if busy, "R" if ready, "J" if in the Jog/slew mode, or "b" if Jog/slewing
X Send current position of motor 1 to host (Motor can be in motion)
Y Send current position of motor 2 to host (Motor can be in motion)
Z Send current position of motor 3 to host (Motor must be stationary)
T Send current position of motor 4 to host (Motor must be stationary)
M Request Memory available for currently selected program
lst List current program to host (ASCII text)

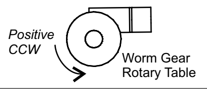
x Send last 4 positions of motor 1 to host that were captured by the "!" command or Input 4 trigger
y Send last 4 positions of motor 2 to host that were captured by the "!" command or Input 4 trigger
Request the number of the currently selected motor
***** Request the position when the last motor started decelerating (shows position when "D" command or Stop/User input 4 used)
? Read state of limit switch inputs for motor 1 and 2 (8 bit binary value)
~ Read state of User Inputs, Motor 1 and 2 Jog Inputs (8 bit binary value)
$ Read state of User Outputs (8 bit binary value)
@ Read user analog input value
B Read Backlash compensation setting
O Read Indicate limit switch setting
D Read/Digitize motor position (Jog Mode)

PM Request the number of the current Program
PMA Request the current program associate number (255= default/disabled)

getMmM Read motor type/size selected for axis m
getDM Read operating mode of VXM
getD0 Gets the VXM's firmware version in the format X.XX
getD1 Gets the VXM's firmware date code in the format XX-XX-XX (month,day,year)
getD2 Returns 2 if system is a single VXM, returns 4 if VXM is a Master
getDA Read Analog Joystick Deadband setting
getjmM Read first Jog Speed setting for motor m
getjAmM Read first Analog Joystick range setting for motor m
getJmM Read second Jog Speed setting for motor m
getJAmM Read second Analog Joystick range setting for motor m
getLmM Read mode of limits for motor m
getPmM Read "Pulse Every x # Steps" value for axis m
getPA Read Pulse width used by **setPmMx** and **U7**
getI Read operating mode of user inputs







Lead Screw Models UniSlide*		BiSlide**	Advance per turn Units	Advance per step Units	Speed @ 1000 SPS (2.5 rev/sec) Units
C	P40	E25	0.025 inch	0.0000625 inch	0.0625 inch/sec
B	P20	E50	0.05 inch	0.0001250 inch	0.125 inch/sec
W1	P10	E01	0.1 inch	0.0002500 inch	0.25 inch/sec
W2	P5	E02	0.2 inch	0.0005000 inch	0.5 inch/sec
W4	P2.5	E04	0.4 inch	0.0010000 inch	1 inch/sec
K1	Q1	M01	1 mm	0.0025 mm	2.5 mm/sec
K2	Q2	M02	2 mm	0.0050 mm	5 mm/sec
Rotary Tables	Gear Ratio				
B4872	72:1		5 degree	0.0125 degree	12.5 degree/sec
B4836	36:1		10 degree	0.0250 degree	25 degree/sec
B4818	18:1		20 degree	0.0500 degree	50 degree/sec
B5990	90:1		4 degree	0.0100 degree	10 degree/sec

* Typical UniSlide model (where x is from above table): MB4024xJ-S4
** Typical BiSlide model (where x is from above table): MN10-0100-x-21

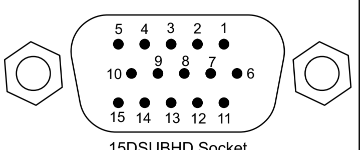
To convert from "real" units to steps, divide the distance desired to move by the Advance per step. (Distance ÷ Advance per step = Steps)

VXM Motor Setting

x	Motor Model (Amps)
0	Default (0.4A to 0.7A)
1	Vexta PK245 (1.2A)
2	Slo-Syn M061 (3.8A)
3	Slo-Syn M062 (4.7A) Vexta PK264 (3A)
4	Slo-Syn M063 (4.6A) Vexta PK266 (3A)
5	Slo-Syn M091 (4.7A) Vexta PK268 (3A)
6	Slo-Syn M092 (4.6A)

Pin#	Name
1	0V
2	+5V
3	Ain
4	Run
5	I1
6	I2
7	I3
8	I4
9	0V
10	J1-
11	J1+
12	J2-
13	J2+
14	O1
15	O2

I/O



15DSUBHD Socket

Phone: 585-657-6151 and 800-642-6446
Fax: 585-657-6153
Email: info@velmex.com
Internet: www.velmex.com and www.bislide.com
mail: 7550 State Route 5 & 20

VELMEX, INC.

Document # VXM-CS 12-29-04
VXM Version 1.30