VXM Command Summary

Most commonly used commands:

F Enable On-Line mode with echo "oFF"
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
R Run currently selected program
C Clear all commands from currently selected program

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)
SmMx Set Speed of motor (70% power), m= motor# (1,2,3,4), x=1 to 6000 steps/sec. (SAmMx is 100% power)
SmM-x Read and assign analog input value to motor m speed (70% power), x=speed range (**SAmM-x** is 100% power)
AmMx Acceleration/deceleration, m= motor# (1,2,3,4), x=1 to 127.

Program management commands:

PMx Select Program number x, x= 0 to 4
PM-x Select and clear all commands from Program number x, x= 0 to 4
PM Request the number of the current Program
PMAx Program Associate program x in Master to program x in Slave (Linked VXMs start the same time) -x or x =255 is disable
PMA Request the current program associate number

Special 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

Pausing and input output commands:

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

Operation commands:

Q Quit On-Line mode (return to Local mode)
R Run currently selected program
N Null (zero) motors 1,2,3,4 absolute position registers
K Kill operation/program in progress and reset user outputs
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)
! Record motor positions for later recall with "x","y" commands
res Software reset controller
del Delete last command

***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)
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
M Request Memory available for currently selected program
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 (8 bit binary value)
~ Read state of User Inputs, Motor 1 and 2 Jog Inputs (8 bit binary value)
@ Read user analog input value
B Read Backlash compensation setting
O Read Indicate limit switch setting
getDx Read mode/version
getDA Read Joystick Deadband setting
getjmM Read first range Jog Speed for motor m. **getjAmM** for Joystick range setting
getJmM Read second range Jog Speed for motor m. **getJAmM** for Joystick range setting
getLmM Read mode of limits for motor m
getMmM Read motor type/size selected for axis m
getPmM Read "Pulse Every x # Steps" value for axis m
getI Read operating mode of user inputs
lst List current program to host (ASCII text)

Commands for two controls connected by VXM bus:

(i3,i1...) Combine Index commands to run simultaneously on two VXM controllers connected by VXM bus
[i1,i2...] Send data to Slave through Master

***Jog mode commands*:**

D Read motor position (Digitize)

Special function and setup commands:

Bx Backlash compensation, on when x=1, off when x=0
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
setDMx Set VXM/VP9000 or NF90 emulation modes, and other operating parameters
setDAx Set Joystick Deadband value
setjmM Set first range Jog Speed for motor m. **setjAmM** for Joystick range setting
setJmM Set second range Jog Speed for motor m. **setJAmM** for Joystick range setting
setLmMx Set limit switch mode for axis m
setMmMx Set axis m for motor type/size x. Also sets default (jog/joystick) motor power to 70%. **setMAmMx** is 100% power
setPmMx Set "Pulse Every x # Steps" on output 2 for axis m
setIx Set operating mode of inputs
setBx Set RS-232 Baud rate (9=9600, 19=19200, 38=38400)

Memory save commands

rsm Run save memory (saves setup/ program values to nonvolatile memory)

Legend: **New Commands for VXM not available on VP9000 or NF90, Different input/output/range/additional values from VP9000, Different command/ function for NF90 mode**

Nf90 emulation mode:

ImM0 Index motor m to absolute zero position
L-0 Sets the Loop-to-marker at the current location in the program
U2 Disable user output when pausing
U3 Enable output when pausing (reset state)

Acceleration/deceleration values will be internally doubled to match NF90's 2x ramp rate
VP9000 Commands not supported by VXM: **U10, U11, U12, U22, U40, U41, U60, U61, U70, U71, U72, U73, { }, %, &**