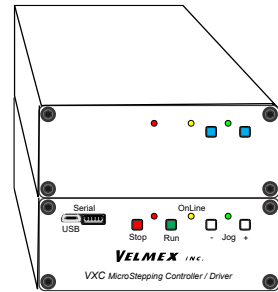


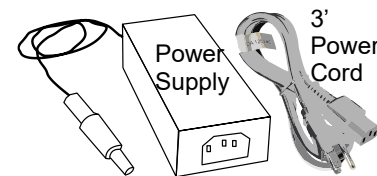
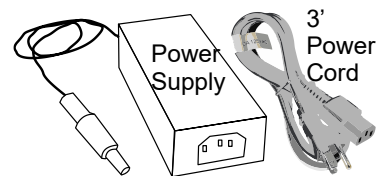
Model **VXC-2** _____

Ser #: _____ CO #: _____

VXC-2, USB Cable, 2 Power Supplies, 2 Power cords Customer: _____ Date: _____



USB Serial Cable



Notes:

Other: _____

* **CAUTION:** Refer to the VXC User's Manual for Precautions & Additional Information
 Visit VelmexControls.com to download the VXC User's Manual

VXC START GUIDE*

Connect:

- A. AC Cords to DC Power Supplies
- B. DC Power Supplies to VXC's
- C. Motor Cables to Motors
- D. Limit Cables to Limit Switches
- E. AC Power Cords to AC
- F. USB Cable Computer to VXC

1 per each axis

NOTE: Always plug VXC power supplies into a common power strip that has an on/off switch

Checked= VXC has been configured for Device

Write Device ID #s here

Axis #1 Device ID # _____

Axis #2 Device ID # _____

- G. Write down Device ID # from tag on each motor & Power-up VXC
- H. Download & Run the VXC Utility App from VelmexControls.com

Cable Color Code & Axis Name

Axis #1	White	X
Axis #2	Blue	Y

IMPORTANT: On multi-axis units, tie cables together around connectors or to the base/ motor to reduce stress on connection

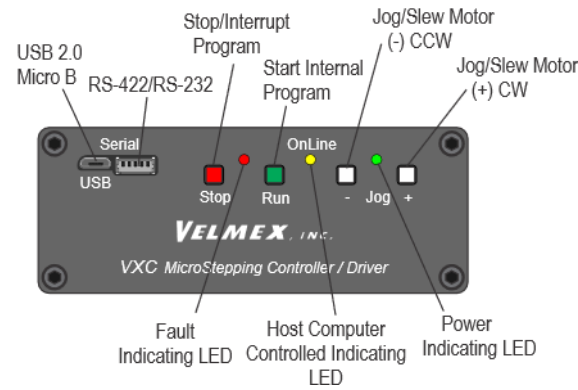


- I. Follow setup prompts in the VXC Utility App. If the VXC Utility App connected to & configured the VXC proceed to step J.

- a. If the VXC Utility App is not compatible with your computer, there are other free terminal apps for serial port communication such as TeraTerm (Win PC), CoolTerm (PC,Mac,Linux), Minicom (Mac,Linux). Set these other terminal apps to 57600 baud, 8 data, parity None, 1 Stop, No flow control, Emulate CR+LF. Select the Serial Port the VXC is connected to. To determine which port the VXC is using, disconnect the USB cable from computer, restart the Terminal App and observe which port is now not showing on the list of available ports. When reconnected to the correct Port, test the communication by typing "E" in the terminal (send) window to put the VXC OnLine (yellow LED on VXC will light.) To go Offline type "Q"
- b. If VXC was preconfigured (check in box G above) by Velmex for axes skip step c.
- c. While the VXC is OnLine set each axis to the Device ID of the attached motor(s) with the "setMLH" command.
setMLHmM=Device ID # m= Axis/motor# (1,2,3,4)
 Example: If Device ID # for Axis/motor# 1= C,1,0,
 Enter: **setMLH1M=C,1,0,**

- J. From the VXC Utility App Terminal you can type and send commands directly to the VXC. Type "E" in the Terminal (send) window to put the VXC OnLine (yellow LED on VXC will light.) Type "Q" to take VXC Offline (back to default Jog Mode.) Type "Help" for VXC built-in help menu. Refer the next page of this document for more examples of VXC commands.

Front (Model VXC-1)



Jog Mode

When the OnLine (yellow) light is off, the VXC is in the Local/Jog mode. Using the front panel jog buttons, each motor can be jogged a single step or slewed to 2000 sps (5 revs/sec.) in either direction.

When a Jog button is pressed the motor moves 1 step (1/400 rev.) If the button is held for >0.3 second the motor will accelerate to 2000 sps. Pressing Stop or the other Jog button while Jogging will hold the current speed (minimum of 63 sps.)

From the VXC Utility App Terminal you can type and send commands directly to the VXC.

These commands are the most common.

ImMx, Set steps to incremental Index motor CW (- is CCW) $m = \text{motor\# (1,2,3,4)}$, $x = \pm 1$ to $\pm 16,777,215$

SmMx, Set Speed of motor (70% power), $m = \text{motor\# (1,2,3,4)}$, $x = 1$ to 6000 steps/sec. Default=2000

C Clear all commands from currently selected program (program 0 is default)

E Enable OnLine mode with echo "on" Use "F" for echo off

Q Quit OnLine mode (return to Local/Jog mode)

R Run currently selected program (program 0 is default)

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

Typical commands to enable OnLine, Clear, & Run motor 1 one revolution:

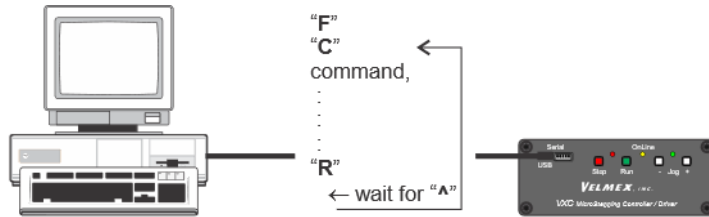
E C I1M400, R

To Clear the previous index from the VXC and move the motor back 2 revolutions:

C I1M-800, R

Common method to send commands is with commercially available languages and Apps such as BASIC, C, LabVIEW, MATLAB, Python, etc

BASIC, C, Python
LabVIEW,
MATLAB, etc.



CAUTION Refer to the VXC User's Manual for Complete Information

Visit VelmexControls.com to download the VXC User's Manual

Have a Question? Email us at:
Support@Velmex.com

Visit VelmexControls.com for examples in the different program languages/ Apps

Common Command Summary*

Motion Commands

ImMx Set steps to incremental Index motor CW (positive), $m = \text{motor\# (1,2,3,4)}$, $x = 1$ to 16,777,215

ImM-x Set steps to incremental Index motor CCW (negative), $m = \text{motor\# (1,2,3,4)}$, $x = 1$ to 16,777,215

IAMMx Set Absolute Index distance, $m = \text{motor\# (1,2,3,4)}$, $x = \pm 1$ to $\pm 16,777,215$ steps

IAMM0 Index motor to Absolute zero position, $m = \text{motor\# (1,2,3,4)}$

IAMM-0 Zero motor position for motor# m , $m = 1,2,3,4$

ImM0 Index motor until home or positive limit is encountered, $m = \text{motor\# (1,2,3,4)}$

ImM-0 Index motor until home or negative limit is encountered, $m = \text{motor\# (1,2,3,4)}$

SmMx Set Speed of motor (70% power), $m = \text{motor\# (1,2,3,4)}$, $x = 1$ to 6000 steps/sec.

AmMx Acceleration/deceleration, $m = \text{motor\# (1,2,3,4)}$, $x = 1$ to 127.

(ImMx,I1Mx,) Run Index moves for axes simultaneously, $m = 2,3,4$ Axis (Axis 1 must be last)

Operation Commands

E Enable On-Line mode with echo "on"

F Enable On-Line mode with echo "off"

C Clear all commands from current program

Q Quit On-Line mode (return to Local/Jog mode)

R Run currently selected program

N Null (zero) motors 1,2,3,4 absolute position registers

Status Request Commands

Help Display context sensitive help screen (for use with terminal program interfacing)

getFac Get all the logged faults with complete descriptions

X Send current position of motor 1 to host (Motor can be in motion)

Y Send current position of motor 2 to host (Motor must be stationary)

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)

Program Management Commands

PMx Select Program number x , $x = 0$ to 12

PM-x Select and clear all commands from Program number x , $x = 0$ to 12

PM Request the number of the current Program

Mem Request Memory available for currently selected program

lst List the current program to host (ASCII text)

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

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)

L-x Loop from beginning or Loop-to-marker $x-1$ times, alternating direction of motor 1

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

Jx Jump to the beginning of program number x , $x = 0$ to 12

JMx Jump to the beginning of program number x and come back after program x ends, $x = 0$ to 12

Pausing, Input/Output Commands

Px Pause x seconds, ($x = 0.0001$ to 5.9999 & 6.0 to 6553.5 sec.)

PAX Pause x seconds ($x = 0.0001$ to 5.9999 & 6.0 to 6553.5 sec, 10 μsec when $x = 0$) Output 1 high for duration of the pause

U0 Wait for a "low" on user input 1

U4 User output 1 "low" (reset state)

U5 User output 1 high

U90 Wait for a low to high on the Run button or connection I/O,4 with debouncing for a mechanical push-button switch