Excitron X Controller User Manual

09/05/2019 v5.25

Introduction and Quick Setup

Excitron's **X Controllers** are integrated with our stepper motors to make your motion control simple and easy. We pack years of design and plenty of power into ultra small new Controllers, loaded with innovative features. They contain all the electronics and power for thousands of motorized applications—and they run right out of the box by simply typing a "**G**" or pushing any Joystick switch.

This manual is for **X** Controllers and all motors. Most commands are capital letters. See Excitron_X_Features.pdf for more details.

You can run standalone and/or send *single letter* serial commands. Serial port communication uses standard 3.3v TTL logic to any smart phone, computer, Arduino, or PLC. Every motor comes with USB-TTL serial port adapter. Various motor size, power, and control functions give you the best selection for your project, and for a broad range of applications.

Our X Controller/motor configuration and part number system is simple—**X86-118** is the number for X Controller assembled with our SM86-118 stepper motor. 86 is the width in mm, 118 is the length in mm.

Stepper motors rotate one small step at a time, and are the best choice for motion, because both position and speed can be precisely controlled, unlike any other type of motor. Our motors can run continuously for 30 years.

We also offer linear slides, belt sliders, actuators, rotary tables from 6" to 39", X-Y tables, and XYZ machines. Visit our website for all documents, prices, and online purchasing.

For a quick start:

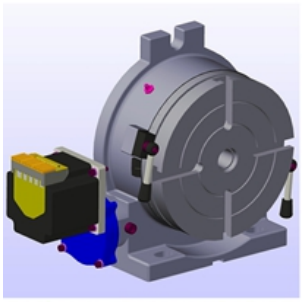
- Make sure the AC-DC power supply is off or unplugged.
- Connect the power supply DC connector, which is a small 1x4 housing, to the **X Controller** 4 pin gold Power header.
- Turn on the power supply, the X Controller LED should be ON.
- Use the built-in Joystick to rotate the motor.
- If you have a USB-TTL adapter, connect your USB cable between your smart phone or computer and the **X Controller** serial 3 pin header. PC USB software drivers must be installed.
- Start a USB smart phone app (Free USB Serial Term) or Realterm (available from our website Serial Programs) or any equivalent program) with **115,200** baud, 8 bits, no hardware handshaking.
- At power-up, the **X Controller** displays:

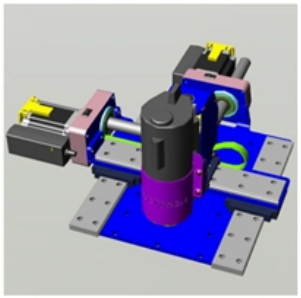
```
Excitron ExROS v5.25 X86-118
14C   24V   Cycles= 0000033
GR= 0486   Steps/rev=0194400
X>
```

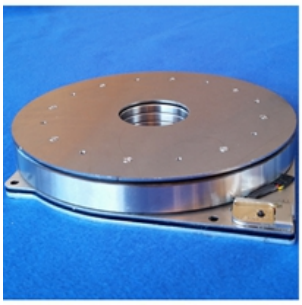
Type **I** to see information and the current motion profile.
Type **G** to run the stepper motor.
Change direction (**C** or **W**), **V**sps, or number of steps **N**, type **G**, and see the difference.
Type **?** To display a brief command help.
It's that simple. You can use inputs for Input Profile, do auto-Home, or run CNC/gcode modes for extra motion control.
Global Controller values are changed in the **c** submenu.


X86-118 with gear


MRC3-57-40M-10 Chuck


8" Rotary Table MRT8-86-80HV


XYZ Robot


MRT8-42-153 Rotary Table


BS80 Belt Slider

What's New

DRO option for displaying key values. Position counter uses Gear Ratio and shows degrees, great for rotary tables.

Optional 12 bit Absolute Encoder, inside the motor, no wires!

Step and Direction on 2 dedicated input pins.

X86 and X110 motors have 10 Joystick switches.

Any Motion Profile can run forever by setting **R** repeat command to 65,535. Joystick center or **O**,**o** stops.

Motion Profile changes are stored immediately, no need to send any commands for storing.

New CNC/gcode Mode--ability to read gcode files, store, view, and execute for easy automation for robots and CNC machines. This saves you time, space, money, and complexity by eliminating external PCs, PLCs, and other expensive CNC controllers. See CNC Gcode section.

Feel free to contact us at info@excitron.com. All available pdfs are in Documentation web link.

Reduction of Hazardous Substances (RoHS) Statement of Compliance.
Excitron is a product manufacturer and does not melt or produce any of the raw materials sold to our customers. Excitron only purchases ROHS parts, does not add or expose these materials to any of the hazardous substances (lead, mercury, cadmium, hexavalent chromium, polybrominated biphenyls, polybrominated diphenyl ether) listed within the EU RoHS Directive, except we use tin-lead solder. As cited in statements from our vendors, the materials sold by Excitron do not exceed allowed levels of these substances because they are exempt, and therefore, the materials are compliant with the EU RoHS directive.

Key Features

The **X Controller** performs all calculations and intelligence to automatically step the motor. It also contains powerful motor power drivers. No additional electronic device is needed to control and drive our stepper motors. See the individual motor **X Controller** data sheets for additional details and connector pins.

All Motion Profiles are pre-loaded with default values. Simple serial port programs like RealTerm or a smart phone USB app, see Serial Programs on website, are used to quickly ***change*** motion parameters.

The parameters for stepping the motor is called a **Motion Profile**. A Motion Profile contains direction, number of steps, accelerating, stepping at a constant maximum Vsps (steps per second), then decelerating to stop precisely at the number of steps you specified. Deceleration is same as acceleration. A time delay before motion occurs gives you precise timing control. Motion Profiles are stored inside the motor, even with power off. See page 8 & 9.

Your goal is stepping your motor from point A to point B quickly and accurately. An Excitron **X Controller** allows you to attain this goal with truly simple commands for friendly use, yet versatile and powerful enough to control a sophisticated robotic automatic assembly machine.

Most step motor documents state that bipolar motors produce more torque than unipolar motors--that is not true with our motors and **X Controllers**. Our proprietary circuitry was first developed and patented in **1976.** Improvements continue steadily to maximize motor and **X Controller** efficiency, which maximizes torque.

Five modes of operation are available: serial commands, Home, Driver, Joystick, and I Input Profile. Encoders are optional. See individual motor drawings for Joystick details.

The **X Controller** uses unique current limiting circuitry for high efficiency and performance. By adjusting speed Vsps you can minimize the time required for a particular motion. Vsps can be changed on the fly while the motor is stepping by pushing the Joystick, see the motor drawings.

Input devices, such as limit switches, may be connected to the TTL/CMOS +3.3 volt logic input pins. Order our handy SW1-2D Switch Assembly. In2 and In3 input pins accept analog 0-3.3v and converts it to a digital value. The ON LED provides visual feedback that power is applied, and it also indicates motor steps or serial communications by brightness.

X Controller uses extremely low supply power for the electronics. With a 19.5 or 24v power supply, the **X Controller** only dissipates 0.025 watt, which is insignificant. All internal voltages are self-generated and thus require only a single voltage supply. Built-in temperature sensing on X Controllers safeguards the electronics and motor from thermal burn out, the limit is 170 ºF (77ºC).

Excitron **X Controllers** drive any size unipolar stepper motor, even our 110-180—a 25-pound Super-size NEMA 42! Many options are available. All **X Controllers** are upgrade-able at our factory, via firmware updates, thus protecting your investment. Custom firmware applications are available.

Sending Serial Commands

The format of **X Controller** commands is simple--send the single letter of the command. Important--capital and small letters are different. If user input is expected after the command is typed, the **X Controller** responds with '=', the existing value, a colon and a space. This existing value also shows the exact quantity of numbers to send. Then it waits until you send the required quantity of numbers. After all input is received, the **X Controller** will send a LF (line feed), CR (carriage return), Axis address, and a prompt '>'. The '>' provides feedback to host computers that the command is complete.

Do not type "Enter" key after typing your command, type only the command and any characters as specified by the each command. All ASCII command numbers are fixed field format. You must type the exact required quantity of numbers. For example, to enter a step quantity of 188 steps, type:

N (**X Controller** sends "=0033000: ") then **0000188**

All 7 digits must be entered. If you enter fewer digits, the command is left waiting for completion. All input values are validated to be within the range specified for each command. Any input value outside a valid range or invalid characters results in '?' being displayed and the default or existing value being used.

At start of any command, commas, spaces, carriage returns (Enter or CR), and line feeds (LF) are ignored. Serial input characters are *not* buffered.

At power up, the **X Controller** is ready to accept serial commands. Commands may be typed in any sequence and the **X Controller** stores your changes immediately.

Warning! Use caution when operating, severe injury can result from the motor rotating. Long wires act like antennas and may cause erratic dangerous motion. Static can reset the Controller, causing motion if auto-Home is set.

Disclaimer—all values and statements contained herein are subject to change without written notice. **No products manufactured by Excitron may be used for life support equipment, full notice on our website. Use extreme caution when operating--severe injury or death can occur because motors move and rotate.**

Serial Port Setup and Power

Excitron motor **X Controllers** can operate **standalone**, and in various modes. Two on-board USART serial ports are 3.3v TTL, use either one. The CNC port is for Gcode and CNC2 is used for Slave communication. Every motor kit includes our USB-TTL adapter, AC power cord, AC/DC switching regulated power supply (100-240VAC 50/60 Hz), with sufficient amperage to drive your step motor. The power supply voltage range for X86 and X110 large motors is 9.2 to 28V, while all others operate from 3.3v to 28v. These and other accessories are available on our web site. See each motor and controller drawing for interface pins, more details in Excitron_SW1-2D.pdf.

The power supply wires should be at least 22 AWG stranded, 18 AWG is best, and as short as possible, 2 to 8 inches. The amount of current into the motor is NOT the power supply current. A rough guideline is the power supply watts out to the motor equals the motor watts (watts = volts x amperes). You will need to budget at least twice the motor's watts. Power (in watts) is also defined as amperes x amperes x resistance.

If the power supply amperage is inadequate, the **X Controller** may pull the power supply voltage low enough to cause a reset of the **X Controller.** This stops your motion. Only use Excitron's power supplies!

A smart phone with USB serial app, or a PC running RealTerm, Hyperterminal, or any serial port program is required to make changes. Each serial port operates with 8 bits, no parity, 1 stop bit, and 115,200 baud. Select the correct PC Port number. Protocol is none, no hardware handshaking, and just ASCII alphanumerics. The **X Controller** will echo all characters received on the USB-TTL CNC port, & echo on the CNC2 port if in Motion Profile mode. Turn off Echo in your communication program. Once setup in various modes, you may not need any serial ports. Each X Controller has an assignable Axis address A, B, C, D, E, X, Y, Z, the default address is '**X**'.

Changing X Controller Configuration

The little **c** command is for changing the X Controller **global** operation and features. Type **c**, then type a **~** to continue. **c** submenu commands are:

A=X: Axis address **ABCDEXYZ** for CNC/Gcode.
C=0: 0 = do Motion Profile at power-up, 1 = do CNC 1st
E=005: Encoder: 000 = disable, nnn = # of steps before stoppage. If you do not have an encoder, you must set E=000.
F=002: First steps at low speed at the beginning of the Profile.
G=486: Gear ratio, set =001 if no gears. If ratio > 1, then Position counter shows 0 to Gear Ratio X Motor Steps/rev.
H=0: Home = 1 = auto-runs at reset/power, 0 = disable auto-run.
I=+06 +09 +12 +15: **+** is enable, **-** is disable. 12 & 15 only for X86 and X110 motors. See **IPM** section page 5. & page 9.
L=1: LCD DRO = 0 disables, 1 = enable.
M=1: Master if = 1, slave if = 0. Master always sends responses, slaves only send response if **^** (up caret) command is issued once after power-up. Underscore **_** command turns off slave responses.
P=001 First Profile to be current at startup, max is 037.
O=1: Output motor step pulses while the motor is stepping: 0 = off, 1 = on. Each step transmits **}** (CW) or **{** (CCW) to CNC port, and a negative going pulse to CNC2 Tx pin. Great for external monitoring of position and speed real-time, or driving another Excitron motor in sync via the serial port CNC2 Tx pin. Note: character overrun occurs > 10,472 Vsps, limited by 96 us/char.
T=1: Torque = 1 is high, 0 = 85% of high (runs quieter). If T is changed, Q quit the **c** menu. You must reset by sending @.
Q Quit and return to previous menu, any changes are saved.

Changing Motion Profile Parameters

These commands are for viewing and selecting Motion Profiles:

P=nn Display the motion profile specified by **nn**. Changes can be made to this current Profile.

v View all Motion Profiles. A header line is sent, followed by all the Motion Profiles.

Q Quit to CNC Mode.

All motor parameters required for sophisticated motion control of your motor are described here in alphabetical order. You type only the characters shown in **bold**. The **X Controller** then displays equals and the existing value (from 1 to 7 numbers), colon and a space. Then type new numbers. See page 8 also. Changes are saved automatically.

A=0: **n** Acceleration, 0 for half accel, 1 is faster.

B Brake continuously, when not stepping.

C CW clockwise motor direction, from the motor face.

K=100: **nnn** Analog value for checking input pins, see Input Pins and Modes page 5.

p=0: **n** connector pin number for pin Modes.

M=0: **n** Mode number, 00-31, for specified input pin.

N=0012000: **nnnnnnn** **N**umber of motor halfsteps.

R=00000: **nnnnn** **R**epeat quantity, from 00000 to 65534. Repeat forever if 65,535. Joystick center or **O**,**o** stops.

t=00400: **nnnnn** **t**ime delay before the motor motion, in milliseconds.

V=04600: **nnnnn** Velocity--maximum half-steps per second, 64 min and 25,000 max.

W CCW Counter-clockwise motor direction.

^ up-caret character—sending to a slave enables its serial port responses, only used in CNC/gcode, see ExROS.

_ Underscore character—sending to a slave disables all of its serial port output, except **>** for completion.

After a certain Vsps, the **X Controller** determines that current limiting is no longer needed, and the motor is "fully on". Torque is then limited by the motor's reluctance. The power supply voltage is the largest factor for torque. Motor current is rated based on the amount of current that causes the motor to reach 85 °C in 10 minutes. If mounted to a relatively large metal plate to dissipate the heat, this current rating will increase significantly. For intermittent operation, which is most stepper motor applications, the motor may be over-driven, limited mostly on temperature.

You can change motion profiles for designing, trouble-shooting, demonstrating, or marketing purposes. For example, most mechanisms operate at blinding fast speed—the motion profiles can be altered to slow down an interesting movement so that fine details can be observed. Since most applications only use less than 5 profiles, you can create an entire second range of motion profiles that operate at slow

speed. You can even offer this feature to your customers.

Stepping the Motor

In Motion Profile mode, you step the motor with the **G**, **g**, **S**, **)** and **(** commands:

G Go step the motor for the specified **N** number of steps.
g Reciprocating motion--same as **G**, and at its completion, the motor reverses direction and performs exactly the same motion profile.
S Step the motor forever. This command ignores the step number. Only a stop command **O** or **o,** or a pin/mode, or a center Joystick push will stop the motor.
) Single step CW.
(Single step CCW. Both **)** & **(** are useful for a serial port step and direction mode.

After you type **G**, **g**, or **S,** you can send **U, D, O,** or **o:**

U Speed up the motor on-the-fly, until ~5,000 sps.

D Slow down the motor on-the-fly, until ~5,000 sps.

O or **o** Starts deceleration and the motor slows down and stops. Pushing J5 or J10 also stops the motor.

Use the built-in Joystick (up/down arrows) to change speed while running. Push J2 to save this new ***global*** Vsps value.

If *you* stop the motor, the remaining Motion Profiles do not run. Example: In2 goes low to run Profiles 06, 07, 08. You send 'o' which stops Profile 06, then Profiles 07 & 08 do not run.

After running, the motor coils are energized for 2 seconds to electronically dampen and hold the motor shaft for ultimate step response. Then the Brake value, if non-zero, is applied to the motor coils. You may immediately start another run during this damping. This feature is especially useful for vertical drives.

Warning—stepper motors require amps of current, and overheating can occur in minutes. Maximum operating temperature is hot to the touch, and may burn you.

If the temperature limit is reached, the motor ceases running. The Controller's address is sent. No operations occur until an @ reset character is received or power is cycled.

Braking

When the motor is not stepping, motor current (amps) may be applied to create a braking (holding) torque. WARNING--high temperatures may occur rapidly with high brake values over 090. Values from 010 to 090 produce little braking, 060 to 094 are typical. Always increase braking values slowly and check heat after 5 minutes. Don't turn on braking unless you need it! Specify Braking motor current by typing:

B=000: **nnn** Braking (holding) current limiting parameter. Range of values is 000 to 100.

Power saved by minimizing braking results in greater torque available for motor stepping. Higher current results in higher temperatures. **B**=000 also reduces the **X Controller** power consumption. Any reset uses your brake for Profile #01, suggest keeping B=000 for #01.

Home Command

You can enable the X Controller to "auto-home" at start-up by setting **H** (in the little **c** menu) to 1:

c>H=0: **1** 0 = disable, 1 = enable run once at power-up

c>R=0: **1** 0 = disable, 1 = auto-repeat forever at power-up

When enabled, Motion Profiles 02-05 run to completion, even without a sensor in your system. You have complete control of the values in these 4 profiles, including direction, distance, and using any input pins. Suggest setting Profile 03 for backing up a small amount of steps at a lower speed, to increase position accuracy. If you don't need any Profile, then set **N**=0000000, **t**=00000, **p**=00. Four @ in a row will clear this H value.

You can run Motion Profiles 02- 05 at any time by sending **H**.

Command Examples and Notes

Example: you wish to rotate the motor for 8,000 steps clockwise, and a maximum Vsps value of 1,800. Send:

N=0000400: **0008000**
C
V=04200: **01800**
G

Note that you do not type '>', '=' nor "enter" with any command, the **X Controller** provides it. The '**>**' prompt is not shown here for clarity .The order of the **N, C,** and **V** commands is not important, but the **G** command is, since it starts motor motion. The **N** and **V** commands must receive their required **exact** digit quantity. To step the motor again, with the same parameters except CCW (counter clockwise), type:

W
G

Note that the other commands do not have to be re-entered. The **X Controller** temporarily stores the last values of all parameters and commands. Save permanently by using the **v** (view all) command.

Be aware that the motor and **X Controller** will heat up faster while running at slower speeds. At Vsps over 3,000, depending on power supply voltage, the motor's inductance, the motor self-limits the power, and should run cooler. A temperatures over 50 °C may feel hot to human touch, but is ok for motor and **X Controller** operation (up to the high temperature limit).

The Controller automatically reduces acceleration as speed increases. This optimizes the stepper motor output torque curve, stronger and better than a linear speed curve.

Many interesting combinations exist to create the motion you need for sophisticated control of automatic assembly machines or for motorized products. The time command **t,** can be coupled with various input mode commands to produce intelligence and control for almost any application.

All stepper motors have a natural resonant frequency around 260 to 1,200 Vsps. If stepped at constant Vsps in this speed range, the motor may vibrate and lose steps. It is best to accelerate through this range. The controller always runs using half-steps.

Power-ON Reset

The **X Controller** has sophisticated power-on reset and brownout

voltage protection circuitry. This protects from power supply glitches, and increases the X Controller's robustness and integrity. After any reset, Brake is 000 for Profile #01 in Command Mode only--the saved eeprom value is unchanged. This protects against unplanned heat build-up. The first motor step after power-up or reset is always the same step, and 1-3 steps may be needed to sync the **X Controller** with the motor's last rotor position. The high value filter capacitors in the power supply provide considerable power supply voltage spike attenuation, and a 1-4 second hold-up time. When switching off, always wait briefly before unplugging the 4 pin power connector. The power supply voltage is read, and is displayed with the **I** command.

Software Reset and @ Commands

Software reset by sending **@**, which hardware restarts the X Controller. Do not use @ command to stop the motor, unless in an emergency.

Other special **@** commands are:

@ sends =Axis address, use when you forget the address.

@@@@ (4 **@**'s in a row, ignore reset output characters)--clears auto-Home & all **I** Input Profile modes. If **H** or **I** is needed, you must set these again.

Input Pins and Modes (while running)

When the motor is running, the Input Pins and Modes are used to give you control for stopping the motor. Note that IPM uses the input pins for starting the motor.

You can create very interesting and versatile stepper motions by using the input pins. Some control features are: stop and go control, logic only, limit switches, wait for two inputs to occur; sequence external events, extended time delays; or other ingenious reasons.

All pins are digital TTL/CMOS compatible (0 to +3.3v) and are also +3.8 volt tolerant. They have Schmidt triggers for noise immunity, 100 ohm resistors, and +-30kV static protection. You read your X Controller pin values and the Input Profile Modes with the **J** command:

```
In2 In3 In5 In6 Shft Enc
 1   1   1   1  220 2770
+06 +09 +12 +15          ← shows which 3 Profiles run,
                           & + enable or - disable
```

The In# is the Input connector pin number; X86 and X110 have In5 and In6. Each input pin is normally high (+3.3v) and can be connected to switches (optical, mechanical, etc.) or to monitor signals from other electronic devices. The correct connection is for one terminal of the switch/control connected to ground, and the other terminal connected to the input pin. Each input pin has a 24K (+- 5%) pull-up resistor to +3.3 volts. **Shft** shows motor shaft step number. **Enc** shows the analog value, & **?** if an encoder error occurred, even if you do not have an Excitron Encoder installed.

You specify the **pin** number (little **p**) in each Motion Profile:

p=3: is the connector pin number.

M =12 is the Mode you wish to assign to your chosen pin number, ranging from 00 to 31.

A pin value of 0 will ignore any pin and its mode in that Motion Profile. Valid Pin Modes are:

00 – wait until pin is low, then step
01 – wait until pin is high, then step
12 – step until pin goes high, then stop smoothly
13 – step until pin goes low, then stop smoothly

An output pin is a custom Excitron firmware request:
28 – output pin is high at start, low after motion profile
29 – output pin is low at start, high after motion profile

Input Profile Mode (IPM) starts running

The **c submenu I** command enables the input pins to randomly run a set of 3 motion profiles. This gives you great standalone operation, and can be used in all other modes. X86 & X110 motors have 4 input pins, all others have 2. For example, 4 push buttons can select any of 4 positions for .01, .10, .50, and 1.00 inches of travel (you set the N value for each). The **I** values are independent of each other, and these are the allowable values:

I=+06	runs Profiles 06, 07, & 08	(In2)
I=+09	runs Profiles 09, 10, & 11	(In3)
I=+12	runs Profiles 12, 13, & 14	(In5)
I=+15	runs Profiles 15, 16, & 17	(In6)

When you type **I**, the Controller sends 2 or 4 (X86 or X110) current values, and waits for you to type **+** (enable) or – (disable) followed by one of the valid numbers, and your new value is auto-saved. IPM Input pins are active ***low***. Once triggered, the 3 profiles run to completion, even if the input pin is raised high (unless you specify Input pin Mode = 12), so use caution. To run only 1 profile when the input goes low, set the remaining 2 profiles with **N**=0000000 and **t**=00000 so that they are ignored. You may also use an Input pin and Mode inside the Motion Profile, after starting with IPM. See last page for details.

Step and Direction Input Pins

You can step the motor by using 2 external input signals Step and Direction:

STEP	negative or positive going pulse, > 5 usec
DIR	0 is CW, logic 1 is CCW

Step/direction input pins are dedicated, always ON. The **X Controller** is CNC-ready, and runs with almost any CNC PC, arduino, or PLC program and **+3.3 volt logic**. The CNC *serial port* can also be used for stepping with **(** & **)** commands. Do not wire more than +3.8v directly to the input pins. Step and Direction ignores Brake, time delay (set time to 0000), and pin/Mode values. An enable signal is not needed because the X Controller will automatically stop when the STEP pulses cease.

Use your favorite CNC program, which send step and direction pulses from a PC parallel port, or other devices like Haas. Step and Direction is disabled in CNC mode. **Important**--if the motor is running using step/direction, do not run **G** or any other command until Driver steps are complete.

Help, Hints and Tips

? Displays the simple command help in each menu.

The Excitron motors and **X Controllers** definitely have the torque range specified, and torque decreases as speed increases. Small burrs, chips, non-flatness, or dents on your moving parts can result in the motor stalling. The faster you step, the worse the effect of the small burr. Also be aware that even small machine screws can exert a thousand pounds of force, bend heavy steel parts, and thus bind up moving parts. Tightening a small timing belt can bend steel motor shafts easily, due to the lever action. See SW2-1D.pdf for more details.

Stepper motors run faster and stronger with direct shaft attachment. Best is to slide our motor shaft into a precision bored hole in your part, and clamp using a C collar clamp. Note that if you're bored hole is off even by .003", your top speed will be limited. Non-rigid "rubber" couplers introduce vibration and phase lags, which can cause motor stalling. Best is to shock mount the motor, not the drive parts.

If your motor just hums without stepping, stop it, and check the drive parts and your **Vsps** values. If your **X Controller** has a Joystick or manual buttons, push one to see correct rotation. Once you verify correct running, then increase **V** to suit your purpose. Generally, the larger the motor, the lower the maximum speed.

Use multiple Motion Profiles for added control and longer time delays. If 1, 2, or 3 Motion Profiles suit your operation without auto-repeating, then use the Joystick switches or **I (Input Profile Mode)**, which trigger ***once*** each time that particular input pin goes low. If you need 4 profiles, and automatically repeating 4 Profiles forever is desired, then use **Home Mode**.

Once a Joystick or IPM motion is triggered, an input pin can control motion, including stopping the motor. See page 9.

You can enable the X Controller to auto-home at start-up by setting **H** (under the little **c** menu) to 1, see Home Command section.

The motor stepping parameters may be loaded by sending an ASCII text file. The format of this file may be any of 3 formats: csv (comma separated values), space separated values, or nothing between values. An example of a csv format, created from MS Excel or Notepad, is:

V04400,B000,N0080000,G or V04400B000N0080000G

Note that the order is not important, except the **G** command is last, which runs the profile. You may send CR and/or LF after each input command, but these are ignored. Use up to a 30 millisecond char. delay. Add up to 120 msec line delay to allow the **X Controller** to finish writing its internal values. Most commands take far less time, experiment.

The X Controller does not buffer input commands. Windows Notepad or similar text editors can be used to capture information via cut and paste, or to set parameters.

The life expectancy for re-writing each EE memory is 100K to 1.4M times (temperature dependent), do not exceed this number. Only changed values are re-written. A Master in CNC Gcode mode has no limits. If you find any Profile will not store new values, then set a different Profile number by using the **c** menu command. Then this number will be the startup Profile and thus used for CNC Gcode.

4 Channel Storage Scope

Type **#** to use Scope. One of the 4 channels is read as 12 bit analog values and displayed, pipelined at 500 Khz:

C Channel, 0 = In2, 1= In3, 2 = In5 for X86 & X110 (others are +Vs), 3 = In6 for X86 & X110 (others are Encoder).

N Number of samples, 0 or 1 =32, 2=64, ..., 126 max=4,048.

R Read, store, and display values simply.

G Graph the stored values, *after* Reading.

S Send the stored values, *after* Reading.

T Time delay reading the data, 0 to 255, 0 or 1 = ~1us, each unit is .25us, 254 = 64us.

? for brief help.

Q Quit, and a hardware reset occurs.

The measurable analog input voltage range is .050v to 2.7v. 0 to .05v read as 0212+-, 2.71v and higher read as 4095. Do not apply any negative voltage, or any voltage higher than 3.6v! Damage will occur.

DRO (Digital Readout)

The wonderful DRO option shows you most key values:

```
Excitron ExROS v5.25 X86-118
21C   24V   Cycles= 0000777
GR=0486  Steps/rev=0194400
P= 0048600   Degrees= 090.0000
P# D G A Brk  Number Repeat
27 C g 1 000 0004800 00000
 Vsps tmsec pin Mode Na  RPM
03600 00300  0   12  J5 000.49
```

Now you can see the rotary table degrees immediately, resolution is +_.0005°! Values update after motor stops running. Additional values will be added in future.

Position Step Counter & Shaft Details

The X Controller maintains a Position step counter, at each motor step, up to +-9999999 **half** steps. The Shaft **half** steps value is also constantly updated real time. Neither are stored after power is off or after a reset. Commands for Position are:

X=+0002200: Change Position **half** steps of this motor, send +- then 7 digits. Shaft is not changeable.

x=+0000333 080 0777 send Position, Shaft halfsteps, & Encoder value, & '?' if an encoder error occurred. If CNC mode, all motors wired in Simple Serial Bus are sent. Response also tells you which motors are wired.

If your Gear Ratio is not G=001, the +- does not apply. Then Position value ranges from 0 to Gear Ratio x Motor Steps/rev, 0 to 0194400 for example (MRT rotary table with 486:1).

ExROS—Excitron Robotics Operating System

New robot technology! You can have total control for 1 to 8 axis robots, products, and CNC machines—without an expensive big external controller!

ExROS is a robotics operating system for easy motor motion, using popular CNC Gcode, while maintaining the Motion Profiles command system described previously. Excitron **Controller/motors** can execute Gcode directly from its own memory, thus eliminating an external PC, PLC, or controller. Saves you money, time, and a lot of space.

Enter the **ExROS** CNC/Gcode mode by typing 'Q', if in Motion Profile Mode, or you can set ExROS to run at power up via c menu. The prompt "X 00001>" indicates **ExROS** CNC mode. You can download your gcode text files into the master Controller, then view, save, retrieve, and execute gcode files.

Unique feature: each axis has its own distance and Feedrate (Vsps speed), better than other gcode controllers. Set only one Controller as a Master, and the other Excitron Controllers in your system as *unique* slaves. You set Master with the **c>M** command. Each X Controller has its own Axis address, and choices are A, B, C, D, E, X, Y, Z. Set the Axis address using the **c>A** command (c submenu). Step and Direction is disabled.

ExROS gives you enough power to sequence over 24,000 motions (gcode lines). Gcode files are read into SRAM memory, or loaded from flash, and then can be stored into flash for permanent storage. **ExROS** commands are:

A **A**xis direction-- enter an Axis letter, then 0 for normal, 1=reverse gcode direction. Changes are stored immediately.
F **F**ile directory in flash storage, shows God & Gcode files.
G **G**o execute the file in RAM. Gcode commands are sent to the Master and slave Excitron controllers. Each Slave uses its current Motion Profile. Underscore commands (do not send any characters, except for the > completion) are automatically sent to slaves.
g same as **G** but is a dry run—motors do not run.
I **I**nformation.
L **L**oad a file from flash to the SRAM.
R **R**ead an external serial port file into RAM memory. Enter a filename after typing **R**. Filename must be at least 2 characters, letters and numbers and some symbols only. First letter of file name must be a letter. You can type gcode manually, and you need to type ctrl-J (LF) at the end of each line.
S **S**ave the loaded SRAM file to flash, type **~** to confirm.
V **V**iew the SRAM file, line numbers are added.
! Erase all flash files, must type **~** twice to be sure, then **G** to erase the God files or **C** for all of the gcode code.
Q **Q**uit and return to Motion Profiles.
? help list of commands.

The input gcode file is a simple text file with industry standard general gcode structure. See ? (help) inside motor, also note:

- Distance value is in motor half-steps, not inches, and is incremental. Limit is ±262,142 (6 digits).
- + is CW, - is CCW, while nothing is assumed as CW.
- Leading zeros on Axis or Feed rate **is not ok**.
- Feedrate is in half-steps/second, and cannot be the 1st line in the file.
- Line numbers and comments (;) are ignored, spaces are allowed.

Example of an input file:

```
Z-2000
F1200 ;sent to all 8 axis
X42000 F2100 Y+02000  ;feed rate sent only to Y Axis
Z2000
M30                   ;end of program, this is mandatory
```

M06 is a general pause or tool change, resume after **~**. **~** is used as a general purpose resume/continue.

Excitron uses this Gcode to machine our own parts. One X86 motor can store over 500 parts!

Simple Serial Bus connects multiple **X Controllers** with one host/PC serial port. One Master and many slaves. For talking to the slave you type: **STX** then **address** then commands, ending with **ETX**. STX is ctrl-B and ETX is ctrl-C. A slave responds to commands only if a STX and matching address is received. You wire the host's transmit to the Receive of one **X Controller**, its Transmit to the next **X Controller** Receive, and so on, and the last **X Controller** Transmit is wired to the Receive of the host. See **Excitron_ExROS_65_Diagram.pdf**. A Simple Serial Bus 3.3v TTL cable is required if more than 1 motor. Keep the shielded cable length short between motors, 6 feet maximum.

Absolute Encoder

If you have the optional 12 bit Absolute Encoder installed, it will give you absolute shaft position and closed loop control. The Position Step Counter and Shaft Step Counter gives you the exact position of the motor, and the encoder verifies the motor steps. Encoder value is displayed after each motor run, and with **J** and **x** commands, 0620 to 3920 is typical.

You can measure a full revolution of encoder values by: N=0000001, V=00200, R=00400, then **G**. Copy/paste to spreadsheet for analysis.

If a stoppage occurs, a **?** error output message is appended to the Encoder value on the CNC serial port, and the motor motion stops.

Any motion may be harmful to people or your mechanism, so use care in all situations.

Motion Profile and Command Details

The **I** command provides important information:

```
                    version     motor
Excitron ExROS v5.25 X86-118

14C   24V   Cycles= 0000033          (temperature, power supply voltage,  cycle count)
GR= 0486   Steps/rev=0194400        (Gear Ratio, Steps/rev) these values are for rotary table with 486:1)

P# D G A Brk  Number Repeat Vsps tmsec pin Mode Na      2 letters at end of each Profile indicate Name (where used)
27 C g 1 000 0004800 00000 03600 00300  0   12  J5      ← current Motion Profile values
```

These two lines are displayed after **I** (Information), **P** or **v** commands. It also shows the quantity of input numbers for each parameter. The following table describes each of the column headings, and defines the minimum, maximum, or allowable values for the parameters. Actual single letter commands are bold. Remember to type **?** for built-in help.

Heading	Cmd	Description of Motion Profile values, changes are automatically saved immediately.
P#	**P**	Profile number, 01 to 32
D	**C** or **W**	Direction of rotation, C (clockwise) or W (counter clockwise).
G	**G**,**g**,or **S**	Go—run this motion—**G** (single), **g** (reciprocal), **S** (continuous).
A	**A**	Acceleration: 0 medium or 1 high, decreases as speed increases.
Brk	**B**	Brake, 000 to 100—use care with high braking values—temperature rises—100 is full on!
Number	**N**	Number of steps, 0000000 to 9999999 (for larger values, use **S** command for continuous stepping).
Repeat	**R**	Repeat quantity, 0 is none (run once) to 65534, each run shows the remaining number. 65535 is repeat forever.
Vsps	**V**	Velocity maximum steps per second, 00064 to 25,000, max depends on motor size.
tmsec	**t**	time delay in milliseconds before running the motion profile, 00000 to 65535.
pin	**p**	Input pin to control this motion, 0 means to ignore any pin *and* pin Mode in this Profile.
Mode	**M**	enter a Mode number, 00-31, for input pins, not validated so enter only valid values, see page 5.
Na	--	Name associated with this Profile

Options

DRO option for displaying key values.
Switch Assembly for remote control.
Firmware modifications--to reduce or eliminate your external electronics, to save you money.
RS232-TTL external adapters for serial communications.

Errata and notes

09/06/10 v5.25 New X controller, Step and Direction always ON, deleted Driver Mode.
03/26/19 v5.21 New Gear Ratio command, Position Counter also shows degrees for rotary tables.
02/19/19 v5.20 DRO available. All analog measurements are 12 bit, **K** command deleted. Better Scope commands.
11/18/17 v5.18 Driver Mode step and direction works great. Modes 12 or 13 do not end motion, fixed in v5.20.
11/11/17 v5.17 Encoder value output on **x** command, and while motor running if E=254, for calibration. Fixed CNC Save command, now saves.
08/26/17 v5.16 Joystick J5 & J10 stop motor if running. Output motor char is { and }.
01/05/17 v5.15 X86 and X110 motors have 10 Joystick switches, each runs 2 Profiles. Shaft Position is now halfsteps. Profile 2 letter name shown at end of line. Startup Profile number can be set.
12/06/16 v5.14 Torque is either high (normal) or medium (70%).
09/04/16 v5.12 Save changed Motion Profile values immediately, including **G**,**g**,or **S**, not with **v** command. Deleted volts display. Time delay is now only at start of run. Deleted Torque command, all motors are factory set for 75% amperage level. If in CNC Gcode, the Cycle counter is updated for each Master motor run, but is not saved to eeprom. If you turn off power, this counter will resume the last saved value.
04/10/16 v5.09 Joystick Center runs Profile 30, 31, & 32. Deleted Acceleration command.
03/20/16 v5.08 Torque increased, especially at > 16,000 hsps. Vsps down to 64. 4 @s clears Home and **I** values. Each @ sends axis letter. Better speed up/down using D/U serial or by Joystick. 'O' or 'o' now stops time delay in Motion Profile.

Handy reference for using Joysticks or Input Profile Modes with pin Modes

You can achieve remarkable and powerful motion by combining **I** Profile Modes with pin/Modes. The following example is how we may setup the Motion Profiles for use with our Switch Assembly, yours may vary. It explains the simple difference between **I Input Profile Modes** and **Input pin & Modes** on page 5. Please read those sections. Extra line spacing is shown here for clarity. The following view is an X86-118, by sending the **I** and **v** commands, then copy/paste here:

```
X> I
Excitron ExROS v5.25 X86-118
14C   24V   Cycles= 0000033
GR= 0486   Steps/rev=0194400
P# D G A Brk   Number Repeat Vsps  msec pin Mode        2 letters at end of each Profile indicate where used
05 C G 0 000 0004800 00000 03200 00300   0   00 H5    ← current Motion Profile, #05 for example
X>v
P# D G A Brk   Number Repeat Vsps tmsec pin Mode
01 C G 0 000 0000400 00000 03200 00300   0   00 01    ← general use
02 C G 0 000 0000400 00000 03200 00300   0   12 H2    ← Home2 values
03 C G 0 000 0001600 00000 03200 00300   0   12 H3    ← Home3 values
04 W G 0 000 0003000 00000 03200 00300   0   12 H4    ← Home4 values
05 C G 0 000 0000020 00000 03200 00300   0   00 H5    ← Home5 values
06 C G 0 000 9999999 00000 03200 00300   2   12 N2    ← In2, IPM runs 6, 7, 8
07 C G 0 000 0000000 00000 03200 00300   0   00 N2
08 C G 0 000 0000000 00000 03200 00300   0   00 N2
09 W S 0 000 0004000 00000 03200 00300   3   12 N3    ← In3, IPM runs 9, 10, 11
10 C G 0 000 0000100 00000 02200 00300   0   00 N3
11 W G 0 000 0000080 00000 00800 00300   0   00 N3
12 C G 0 000 9999999 00000 03200 00300   5   13 N5    ← In5, IPM runs 12, 13, 14 (X86/X110 only)
13 W G 0 000 0000000 00000 03200 00300   0   00 N5
14 W G 0 000 0000000 00000 03200 00300   0   00 N5
15 W G 0 000 9999999 00000 03200 00300   6   13 N6    ← In6, IPM runs 15, 16, 17 (X86/X110 only)
16 W G 0 000 0000000 00000 03200 00300   0   00 N6
17 W G 0 000 0000000 00000 03200 00300   0   00 N6
18 C G 0 000 0000040 00000 01200 00300   0   00 J1    ← Joystick 1 (45°), runs Profiles 18, 19
19 C G 0 000 0000000 00000 02000 00300   0   00 J1      If running, this switch speeds up the motor.
20 C G 0 000 0000000 00000 02000 00300   0   00 J2    ← Joystick 2 (135°), runs Profiles 20, 21
21 W G 0 000 0000040 00000 01200 00300   0   00 J2      If running, this switch saves the current speed as global.
22 W G 0 000 0000000 00000 02000 00300   0   00 J3    ← Joystick 3 (225°), runs Profiles 22, 23
23 W G 0 000 0000000 00000 02000 00300   0   00 J3
24 C G 0 000 0000800 00000 01200 00300   0   00 J4    ← Joystick 4 (315°), runs Profiles 24, 25
25 C G 0 000 0000000 00000 02000 00300   0   00 J4      If running, this switch slows down the motor.
26 C G 0 000 0000000 00000 02000 00300   0   00 J5    ← Joystick 5 Center runs Profiles 26, 27 when released
27 W G 0 000 0000800 00000 01200 00300   0   00 J5      If running, this switch stops the motor.
28 W G 0 000 0000000 00000 02000 00300   0   00 J6    ← Joystick 6 (45°), runs Profiles 28-29, only if 2nd Joystick
29 W G 0 000 0000000 00000 02000 00300   0   00 J6          (only X86 & X110, after May 2017)
30 W G 0 000 0008000 00000 01600 00050   0   00 J7    ← Joystick 7 (135°), runs Profiles 30-31, only if 2nd Joystick
31 C G 0 000 0000800 00000 01600 00050   0   00 J7
32 C G 0 000 0000000 00000 01600 00050   0   00 J8    ← Joystick 8 (225°), runs Profiles 32-33, only if 2nd Joystick
33 W G 0 000 0000100 00000 01600 00020   0   00 J8
34 C G 0 000 0000800 00000 01200 00300   0   00 J9    ← Joystick 9 (315°), runs Profiles 34-35, only if 2nd Joystick
35 C G 0 000 0000000 00000 02000 00300   0   00 J9
36 C G 0 000 0000000 00000 02000 00300   0   00 J0    ← Joystick 10 Center runs Profiles 36-37, only if 2nd Joystick
37 W G 0 000 0000800 00000 01200 00300   0   00 J0
```

Example: If **IPM** is enabled for In3 (I=+09 in c menu) and you push a switch (low=ON) on In3, then Profiles 09 runs. If S command, Profile 09 runs forever. When you flip the switch to high (OFF), the motor gracefully stops because of pin 3 mode 12. Same pin used to start and to stop. Then Profiles 10 & 11 run. If you don't want any motion in any Profile, set N=0000000, t=+00000.

How does this work? Profile 09 starts and runs forever because you pushed the In3 switch (IPM). While running, the same pin is then used for an Input Pin. When you release In3 (pin is high), the motor stops, because of pin = In3 and Mode = 12 (run until this pin goes high). Note that the time delay gives you a little time to let go of the switch before repeating.

If you want to run a fixed distance when you push In9, then make **N** = your step quantity and pin Mode = 0 00, then when you flip the switch up and down, the motor runs the number of steps you specified. See Excitron_SW1-2D.pdf for more details.