

Introduction and Quick Setup

Excitron's **X Controllers** are fully 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 the Joystick.

This manual is for **X Controllers** and all motors. Most details are the same as for our recent Au controller. Most commands are now 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. Our USB-TTL and RS232-TTL serial port adapters are available. Various motor size, power, and control functions give you the best selection for your project.

Our X Controller/motor configuration and part number system is simple—**X57-76M** is the number for X Controller assembled with our SM57-76M stepper motor. 57 is the width in mm, 76 is the length in mm.

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

Our experts are happy to assist you in selecting the best X Controller, stepper motor, and accessories for your project. We also offer linear slides, belt sliders, actuators, rotary tables, X-Y tables, XYZ machines, and heavy duty 3-axis milling 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 app or Realterm (available from our website) or any equivalent program) with **115,200** baud, 8 bits, no hardware handshaking.
- At power-up, the **X Controller** displays:

Excitron ExROS v5.07 X86-118 400 000000 +16°C 24v

X>

Type **G** to run the stepper motor.

Type **I** to see information and the current motion profile. 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.

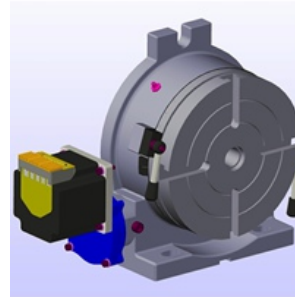
It's that simple. You can use inputs for Input Profile, do auto-Home, or run CNC/gcode modes for extra motion control. Global Controller values are changed in the **c** menu.



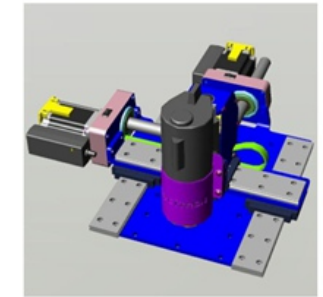
X86-118 with gear



MRC3-57-40M-10 Chuck



8" Rotary Table MRT8-86-80HV



XYZ Robot



MRT8-42-153 Rotary Table



BS80 Belt Slider

What's new

New CNC/gcode Mode--ability to read gcode files, store, view, and execute for easy automation for robots and CNC machines. Enhanced Master/Slave features--one master Excitron X Controller sends serial commands to other Excitron X Controllers via our Simple Serial Bus. This saves you time, space, money, and complexity by eliminating external PCs, PLCs, and other expensive CNC controllers. See CNC Gcode section.

Electronic damping on last step for 2 seconds, then Braking is automatically applied, if enabled.

Command "H" Home allows automatically running Profiles 02-05 at power up, and optional auto-repeat.

Feel free to contact us at info@excitron.com. All available pdfs are in [Documentation](#) weblink. Read Excitron_X_Features.pdf.

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

This document applies to all Excitron **X Controllers**. See the individual motor **X Controller** data sheets for additional details and connector pins.

The **X Controller** performs all calculations and intelligence to automatically step the motor. It also contains powerful motor drivers. No additional electronic device is needed to control and drive our stepper motors.

The controller runs stand-alone. You change the **simple** motion parameters using serial communications and control, thus no programming is required; all you need is a basic serial interface program like RealTerm or a smart phone USB app, see [Serial Programs](#) on website

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.

You can quickly change motion parameters to achieve your goal. You may opt for higher torque and faster movement after finding that the temperature rise is satisfactory. Or you may desire to reduce mechanical noise by varying the acceleration, Vsps, or torque. 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: Command, Home, Driver, Joystick, and **I** Input Profile. Optical Encoders are optional. See individual motor drawings for Joystick details.

A full set of parameters for stepping the motor is called a *Motion Profile*. 30 motion profiles are stored and retained, even with power off. A motion profile consists of acceleration, stepping at a constant maximum Vsps (steps per second), then deceleration to stop precisely at the number of steps you specified. Deceleration is identical to acceleration. A time delay before or after motion occurs gives you precise timing control.

Some key parameters are acceleration, maximum Vsps, number of steps, direction, and torque control. The **X Controller** uses unique current limiting circuitry for high efficiency and performance. By adjusting the acceleration, top Vsps, and other factors, 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.

Various input devices, such as limit switches and potentiometers, 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 LED provides visual feedback that power is applied, and it also indicates motor steps or serial communications by being brighter.

One key unique feature is the extremely low supply power for the **X Controller** electronics. With a 19v 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 °F (77°C).

Excitron **X Controllers** drives any size unipolar stepper motor, even our 110-180—a 25-pound Super-size NEMA 42! Many options are available. All **X Controllers** are upgradeable at our factory, via firmware updates, thus protecting your investment. Custom firmware applications are also available, just ask the friendly Excitron technical staff.

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. In other modes, the mode character is also sent.

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 this range results in a '?' being displayed and the default or existing value being used. If you send invalid characters, a '?' is displayed and no values change. The exception to this rule is that 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** remembers your last typed values, unless you activate another Motion Profile or use Joystick and Inputs. You save values permanently with the **v** (view) command.

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.**

Excitron motor **X Controllers** can operate standalone, or in various modes. The on-board USART serial port is 3.3v TTL, and 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.

The power supply wires should be at least 22 AWG stranded, 18 AWG is best, and as short as possible, 4 to 12 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.

Excitron's **X Controllers** can operate at 1.5 times the rated value for intermittent operation. 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 is harmless to the **X Controller**, but stops your motion.

A smart phone with USB serial app, or a PC running RealTerm, Hyperterminal, or any serial port program is required to make changes. The serial port operates with 8 bits, no parity, 1 stop bit, and 115,200 baud. Select the correct Port number. Protocol is none, no hardware handshaking, and just ASCII alphanumeric. The **X Controller** will echo all characters received on the USB-TTL port, therefore turn off Echo in your communication program. Once setup in various modes, the serial port is no longer needed. 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 accesses commands for changing the X Controller operation and features. Type **c**, then you must type a ~ to continue. **c** submenu Commands are:

A=X: Axis address **ABCDEFXYZ** for CNC/Gcode.
C=0: 0 = do Motion Profile at power-up, 1 = do CNC 1st
D Driver mode, see Driver Mode section for details (future)
e=0: echo CNC serial port, set = 1 only if standalone motor.
F=002 First steps at low speed at the beginning of the Profile.
G=1: GOD is for future Master of Masters.
H=0: 0 = disable at power-up; 1 = auto-runs at reset/power.
I=+06 +09 +12 +15: + is enable, - is disable. 12 & 15 only for X86 and X110 motors. See IPM section.
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.
O=1: view each motor step pulses while the motor is stepping: 0 = off, 1 = on. If on, each motor step CW is an (, CCW is an (. Use for external monitoring of position and speed real-time, or driving another Excitron motor in sync via the serial port. Character overrun occurs at higher Vsps, so use wisely.
V=0: Voltage setting—future, not used.
Q Quit and return to Motion Profile or CNC menu.
7 035 1st digit is version last digit number, next 3 digits are internal use only.

Setting Motion Parameters

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** displays the equals sign, the existing value (from 1 to 7 numbers), colon and a space (most commands).

A=0: **n** Acceleration, 1, 2, or 3 is allowed, 3 is fastest.
C Clockwise motor direction, as viewed from the motor face.
N=0012000: **nnnnnn** Number of steps.
R=00000: **nnnnn** Repeat quantity, from 00000 to 65534.
t=+-00800: **nnnnn** time delay either before (+) or after (-) the motor motion. Input +- or -, then 5 numbers representing milliseconds, e.g., **t=+00500** defines a 500 millisecond time delay after the motor motion.
V=04600: **nnnnn** Velocity--maximum Vsps. Note that the step quantity and acceleration values may define a motion such that the maximum Vsps will not be achieved.
W Counter-clockwise motor direction.
) Rotate 1 step clockwise.
(Rotate 1 step counter-clockwise.
^ up-caret character—sending to a slave enables all serial port responses.
_ Underscore character—sending to a slave disables all of its serial port output except > for completion.

The Damping delay gives you automatic holding at the end of motor running for 2 seconds, not adjustable.

Setting Motor Stepping Torque

Motor torque is directly proportional to motor current, with some non-linearity. Torque is defined as Force x Distance, and is usually measured in oz-in (ounce-inches). See Excitron_X_Torque_Graphs.pdf in Documentation. The **X Controller** uses unique current limiting circuitry for high efficiency and performance for the user. Step motor current (amps) must be limited, or excessive heat and amperage can destroy motors and **X Controllers**. Excitron stepper motors are rated for 2 to 5 volts; so Excitron's current limiting allows them to be connected to a higher voltage (12, 19, and 24 volt) power supply. You control precisely the amount of torque in the motor with one simple command:

T=060: **nnn** sets the internal % of full motor torque.

A low value for **T**, such as 020, produces low torque. Typically use T=040 to 090. Higher values increase torque. Start at low values for **T** and increase the value until your motion is satisfactory. **Do not** increase **T** without adjusting **V** first. **Do not** increase **T** to make the motor run faster. At higher speeds the **T** is not effective—it is only effective for Vsps < 3,000. After determining the suitable values, adjust **T** up or down for best performance vs. heat buildup. The X Controller measures power supply voltage at start-up, and automatically adjusts the amperage. You do not need to adjust **T** if swapping power supplies, the voltage measurement is automatic.

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 up to 2 times its rated power, depending on many factors.

Warning—stepper motors require amps of current, and overheating and damage can quickly occur. Start with low **T** values, then gradually increase while monitoring temperature via the *i* display. Maximum operating temperature is hot to the touch, and may burn you. Selecting higher torque than allowed may damage the **X Controller**. You are responsible for any damage to the **X Controller** or motor. If the temperature limit is reached, the motor ceases running. Then the LED blinks once per second, and a “s=x Hi temp” message is sent from the serial port, where the “s=x” part of the message is the address of the unit sending this message. No operations occur until an @ reset character is received or power is cycled. Note that this terminates Motion Profile Mode, and the **X Controller** is then in serial Command Mode.

Stepping the Motor

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 continuously. This command ignores the step number entered. Only a motor stop command **O** or **o** will stop the motor.
-)** run one motor step CW, any unsaved changes are discarded.
- (** run one motor step CCW.

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

- U** Speed up the motor on-the-fly.
- D** Slow down the motor on-the-fly.
- O** or **o** Starts deceleration gracefully and the motor slows down and stops. Pushing JCEN also stops the motor.

Use the built-in Joystick to change speed while running. The new global Vsps value is saved by pushing J2.

After running, the motor coils are energized for 2 seconds to electronically dampen the motor shaft for ultimate step response. You may immediately start another run during this damping. This feature is especially useful for vertical drives.

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 110. Values from 040 to 060 produce little braking, and values between 080 and 100 are typical. Always increase braking values slowly and check heat. If you do not need braking

torque, specify zero braking current by typing:

B=000: nnn Braking (holding) current limiting parameter.
Range of values is 000 and 010 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 makes **B=000** for Profile #01 only during Command mode, the saved **B** value is unchanged for all other modes. **P 01** or **v** restores your **B** value, be careful.

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

When enabled, Motion Profiles 02-05 are run to completion, even if you don’t have a sensor in your system. You have complete control of the values in these 4 profiles, including which direction to rotate 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 Profiles 04-05, then set **N=0000000** and **t=00000**.

You can run Motion Profiles 02- 05 at any time by sending **H**. Note that auto-home will auto-repeat forever at power-up if you enable this feature by setting **R=1** (in **c** submenu).

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=0000080: 0010000
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).

Acceleration, maximum Vsps, torque, and other parameters can be optimized for your application. An acceleration value of **1** is a good starting point. The X Controller automatically decreases acceleration as speed increases. This optimally matches the stepper motor’s output torque curve, 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. This condition is worse at higher torque settings. It is best to accelerate through this range. Lower **T** values reduce resonance and miss-stepping. 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. The X Controller waits a few seconds for your power supply to stabilize before allowing any motor motion. The RESET pin is normally high. To reset the **X Controller**, send a logic low-level control to this pin, and release. 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.

Software Reset and @ Commands

Software reset by sending **@**, which restarts the X Controller as if a power-on reset occurred. Use this command to terminate Motion Profile and Driver Modes, and run serial commands. Do not use this reset command more than 100,000 times due to eeprom write limit.

Other special **@** commands are:

@@{ followed by any character (besides STX)--sends Axis address.

@@@ (3 @'s in a row)--clears Home & all **I** Input Profile modes.

Input Pins and Modes

You can create very interesting and versatile control of stepper motions by using the input pins and their pin Modes. Some control features are: stop and go control, logic only; wait for two inputs to occur; sequence external events, extended time delays; or other ingenious reasons. All pins are TTL/CMOS compatible (0 to +3.3v) and are also +5 volt tolerant. They have Schmidt triggers for noise immunity, 200 ohm resistors, and +-100kV static protection. You read your X Controller pin values and the Input Profile Modes with the **J** command:

```
In2 In3 In5 In6
  1   1   1   1
```

The In# is the Input connector pin number; only 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. Input In2 is digital or analog (12-bit resolution), depending on the pin Modes, and the other pins are digital only, and have an 24K (+-5%) pull-up resistor to +3.3 volts. The **J** display is the digital representation of the 8-bit analog value, with a full range from 000 (zero) to 255 (+3.3V). If used for digital switch inputs, then any value near 000 is a digital low while any value close to 180 is a high. Use a 25K ohm potentiometer.

You specify the **pin** number (little **p**):

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
- 09 – at power-up, count pin CTR, using Profile 01 repeat value
- 12 – step until pin goes high, then stop smoothly
- 13 – step until pin goes low, then stop smoothly
- 14 -- step until analog pin In2 > ADC value, then stop smoothly
- 15 -- step until analog pin In2 < ADC value, 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

Driver and Input Profile Modes restrict In2 and In3 as general input. For **repeat** and/or reciprocal motions, an input triggers only 1 motion, not multiple motions.

I Input Profile Mode (IPM)

The **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. The **I** values are set independently of each other, and these are the allowable values:

- I=+06** runs Profiles 06, 07, & 08 (In2, analog only)
- 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 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. The analog **I** Profile Mode for In2 will trigger Profiles 06, 07, & 08 when the analog pin value is **lower** than the **K** ADC value you specify in Profile 06. Example: **I** =+06 and **K** value = 190. Profiles 06, 07, & 08 run when the analog value on In2 pin is lower than 190 (~2.5 volts). Input pins are active **low**. Once triggered, the 3 profiles run to completion, even if the input pin is raised high, 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. Driver Mode disables the corresponding Input Profile Modes. Sending 3 **@**'s in a row

(@@@) clears **I** to 00 and Home=0. See last page for more details.

Motion Profiles

The **X Controller** stores all motor stepping parameters in each of the Motion Profiles. You can view and edit these profiles individually. The motion profiles are stored in EE memory, and thus retain their values when power is off.

The major purpose of motion profiles is to allow a standard programmed **X Controller** to operate as a standalone unit, not needing a PC or external **X Controller**. Once set up, only a power supply is needed. With the **H** command, the Controller runs Profiles 02 through 05, and automatically repeats if c-menu **R** = 1 (enabled).

You can easily modify existing motion profiles to fit your own application by entering values for Acceleration, Vsps, etc. The **X Controller** provides for a broad range of applications. All you do is change the profiles.

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.

Here are the commands for viewing, selecting, and saving Motion Profiles: If you issue a **v** or **Q** command, changes to current parameters are saved automatically.

- P=nn** Display the motion profile specified by **nn**.
- v** View all Motion Profiles. A header line is sent, followed by all the Motion Profiles. Current Profile data is saved.
- Q** Quit to CNC Mode. Current Profile data is saved.

Note that the **last G, g, or S** executed is saved into the current Motion Profile. 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:

```
A0200,V04400,B000,N0080000,v
```

Note that the order is not important, except the **v** command is last, which saves the input values to current Profile number. You may also have CR and/or LF after each input command, but these are ignored. If you send multiple lines, you must insert a 300-millisecond delay after each line to allow the **X Controller** to finish writing its internal values. 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. Note that **eprom** writes require 6ms per character to save data in the **X Controller**. The life expectancy of the EE memory is 100,000 times, do not exceed this number.

Driver Mode (future)

You set Driver Mode for step/dir signals operation by typing little **c** then little **D**=1, and it is saved immediately. Setup your favorite CNC program, which send steps and direction pulses from a PC LPT1 parallel port, or other device. The X Controller steps the motor using 2 external input signals Step and Direction:

DIR 0 is CW, logic 1 is CCW, this is IN2
STEP negative or positive going pulse, > 5 usec, this is IN3

Driver Mode is now integrated with Command serial mode so that you can also run all serial commands, such as **I** or **G**. The only precaution is if the motor is running using step/dir, do not run **G** or any other run command until Driver steps are complete.

The **X Controller** is CNC-ready, and runs with almost any CNC PC or PLC program and **+3.3 volt logic**. An enable signal is not needed because the X Controller will automatically stop when the STEP pulses cease. The X Controller *serial port* can be used for stepping, especially in CNC mode. Do not wire +5v or higher directly to the input pins.

In Driver Mode, internal Torque (amperage), temperature, Full/half step, and Brake values are maintained automatically by the X Controller, using the current Motion Profile. Use caution if you change Profiles.

Switches and Input Profile activations give you easy jogging or other semi-automatic positioning. Since Driver Mode uses two input pins for step/direction, the corresponding Input Profile Modes are disabled. The improved Driver Mode (step/dir) automatically processes active high or low pulses, and with no time restrictions. Since it is possible to run the motor using an input pin or TTL serial characters, do not send CNC step pulses while the motor is already stepping, and vice-versa, or they will be lost.

Help, Hints and Tips

? Displays the simple command help.

The Excitron motors and **X Controllers** definitely have the torque range specified, and generally, 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 leverage action.

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, check your **V**, and **T** values. DO NOT increase **T** while trying to get the motor to spin faster—see other notes about **T**. If your **X Controller** has manual push buttons, push one to see correct rotation. Once you verify correct running, then increase **V** to suit your purpose.

Use multiple Motion Profiles for added control. For instance, for

longer time delays, use multiple Motion Profiles.

If 1, 2, or 3 Motion Profiles suit your operation without auto-repeating, then use **I (Input Profile Mode)**, which trigger **once** each time that particular input pin goes low. And this works well with up to 4 inputs. If you need 4 profiles, and automatically repeating forever is desired, then use **Home Mode**. In either mode, once started, an input pin can control motion. An example is **I=+09**, and Profile #09 has **p=3 mode=00**. Pushing a switch attached to **In3** triggers Profiles #09-12, but motion stops at Profile #10 because **p=3 mode=00**. Pushing your switch again triggers Profile #10 then #11.

CNC/Gcode/Robotics Operating System

New technology! Excitron **X Controller motors** can now execute Gcode directly from its own memory, thus eliminating an external PC, PLC, or controller. Saves you money, time, and a lot of space. 8 axis are standard.

You can download your gcode text files into the **X Controller coder motors**, then view, save, retrieve, and execute gcode files. Unique feature: each axis has its own distance and Feedrate (Vmps speed), unlike all other gcode controllers. Now you can have total control for multi-axis robots and CNC machines. One X Controller operates as a Master, and the other X Controllers in your system are 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.

Excitron provides a mini-operating system for easy motor motion, while maintaining the Motion Profiles command system. It gives you enough power to sequence over 24,000 motions (gcode lines). Gcode files are read into RAM memory, and then can be stored into flash for permanent storage. To use the Gcode/Robot system, type **Q** from the Motion Profiles Motor menu. CNC Commands are:

- A** Axis direction-- enter an Axis letter, then 0 for normal, 1=reverse gcode direction. Changes are stored immediately.
- F** File directory in flash storage, shows God & Gcode files.
- G** Go execute the file in RAM. Gcode commands are sent to the Master and slave Excitron controllers. 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—the motors do not run.
- I** information.
- L** load a file from flash to the SRAM.
- I** line number change, applies only if SRAM file exists. Has no functionality, for future firmware.
- R** Read 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)
- S** Save the file in RAM to flash storage, type ~ to confirm.
- V** view the file in SRAM, 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** Quit and return to Motion Profiles.
- ?** help list of commands.

The input gcode file is a simple text file with general gcode

structure. Distance value is in motor half-steps, not inches, and is incremental. Limit is 5 digits. + is CW, - is CCW, while neither is assumed as CW. Leading zeros on Axis or Feed rate is not ok but do not exceed 5 digit limit. 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
X40000 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 ~.

Excitron uses this Gcode to machine our own parts. One X86 motor can store every part we make, and many more.

Simple Serial Bus connects multiple **X Controllers** with one host/PC serial port. One Master and many slaves. For talking to the slave you must 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 the **Excitron_ExROS_65_Diagram.pdf**. A Simple Serial Bus 3.3v TTL cable is required if more than 1 motor. Keep the cable length short between motors, 5 feet maximum.

Position step counter Details

The X Controller maintains and validates Position step counter, at each motor step, up to +9999999 steps. The Shaft value is also constantly updated real time, stored in volatile memory, but is only a calculated value, not measured. Commands for Position are:

- X** Position value can be changed, +- 7 digits.
- x** send Position and Shaft values.

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:

Excitron ExROS v5.07 X86-118 400 0000000 +16°C 24v (version, controller/motor, halfsteps/rev, cycle count, temp., volts)

```
P# D G A Brk Number Repeat Vsps Trq% +msec pin Mode K
01 C g 2 000 0004800 00000 03600 055 -00300 0 12 180 ← current profile values
```

The last two lines are displayed after **I**, **P** or **v** commands, and are described below. Note that this display 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.

Heading Cmd Description, these values apply to the motion profile, and are all saved when power is off with the U command

P#	P	Profile number, 01 to 30
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: 1 low, 2 medium, 3 high, acceleration, accel. slightly decreases as speed increases.
Brk	B	Brake, 000 to 100—use care with braking values—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 to 65535. While running, the remaining number is output.
Vsps	V	Velocity maximum steps per second, 00256 to 18,000, max depends on motor size.
Trq%	T	Torque percent of full rating, 005 to 150—use with care, increasing T increases the amperage!
msec	t	time delay in milliseconds, before (+) or after (-) 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, not validated so enter only valid values.
K	K	Analog input value 0 to 255, for pin/modes and for I Profile Modes analog compare mode.

Remember to type **?** for built-in command help.

Options

- USB-TTL and RS232-TTL external adapters for USB serial communications.
- Switch Assembly for remote control.
- External Joystick for remote control.
- Firmware modifications--to reduce or eliminate your external electronics, such as PLCs or computers.

Examples of using Joystick or Input Profile Modes (I) with pin Modes

You can achieve remarkable and interesting motion by combining I Profile Modes with a pin/modes. The following example is how we setup the Motion Profiles for use with our Switch Assembly. It explains the simple setup, and to offer further help about the differences between I Input Profile Modes and Input pin & Modes. Please read those sections. Extra line spacing is shown here for clarity. The following view is from an X86-118 by sending the I and v commands, then copy/pasted here:

X> I

Excitron ExROS v5.07 X86-118 400 0000000 +16°C 24v

P#	D	G	A	Brk	Number	Repeat	Vsps	Trq%	+msec	pin	Mode	K
01	C	G	0	000	0004800	00000	03200	055	+00300	0	00	200

>v

P#	D	G	A	Brk	Number	Repeat	Vsps	Trq%	+msec	pin	Mode	K
01	C	G	0	000	0000400	00000	03200	055	+00300	0	00	100
02	C	G	0	000	0000400	00000	03200	055	+00300	0	12	100
03	C	G	0	000	0001600	00000	03200	055	+00300	0	12	100
04	W	G	0	000	0003000	00000	03200	055	+00300	0	12	100
05	C	G	0	000	0000020	00000	03200	055	+00300	0	00	100
06	C	G	0	000	9999999	00000	03200	055	+00300	2	12	100
07	C	G	0	000	0000000	00000	03200	055	+00300	0	12	100
08	C	G	0	000	0000000	00000	03200	055	+00300	0	12	100
09	W	G	0	000	9999999	00000	03200	055	+00300	3	12	100
10	W	G	0	000	0000000	00000	03200	055	+00300	0	12	100
11	W	G	0	000	0000000	00000	03200	055	+00300	0	12	100
12	C	G	0	000	9999999	00000	03200	055	+00300	5	12	100
13	W	G	0	000	0000000	00000	03200	055	+00300	0	12	100
14	W	G	0	000	0000000	00000	03200	055	+00300	0	12	100
15	W	G	0	000	9999999	00000	03200	055	+00300	6	12	100
16	W	G	0	000	0000000	00000	03200	055	+00300	0	12	100
17	W	G	0	000	0000000	00000	03200	055	+00300	0	12	100
18	C	G	0	000	0000040	00000	01200	055	+00300	0	00	100
19	C	G	0	000	0000000	00000	02000	055	+00300	0	00	100
20	C	G	0	000	0000000	00000	02000	055	+00300	0	00	100
21	W	G	0	000	0000040	00000	01200	055	+00300	0	00	100
22	W	G	0	000	0000000	00000	02000	055	+00300	0	00	100
23	W	G	0	000	0000000	00000	02000	055	+00300	0	00	100
24	C	G	0	000	0000800	00000	01200	055	+00300	0	00	100
25	C	G	0	000	0000000	00000	02000	055	+00300	0	00	100
26	C	G	0	000	0000000	00000	02000	055	+00300	0	00	100
27	W	G	0	000	0000800	00000	01200	055	+00300	0	00	100
28	W	G	0	000	0000000	00000	02000	055	+00300	0	00	100
29	W	G	0	000	0000000	00000	02000	055	+00300	0	00	100
30	W	G	0	000	0000000	00000	02000	055	+00300	0	00	100

← CNC Master & Driver Mode uses this Profile

← Home values

← Home values

← Home values

← Home values

← I=06 IN2

← I=09 IN3

← I=12 IN5 (X86/X110 only)

← I=15 IN6 (X86/X110 only)

← Joystick 1 (45°), runs Profiles 18, 19, 20
If the motor is running, this Joystick switch speeds up the motor.

← Joystick 2 (135°), runs Profiles 21, 22, 23
If the motor is running, this Joystick switch saves the current speed as global, to eeprom.

← Joystick 3 (225°), runs Profiles 24, 25, 26

← Joystick 4 (315°), runs Profiles 27, 28, 29
If the motor is running, this Joystick switch slows down the motor.

← Joystick Center runs Profile 30

The far right column shows the Motion Profiles used by pins, Home, or Joystick. Example: Profile 09 has the same input pin In3 and has Mode 12, which is “run until this pin goes high” and this is used to stop motion. Why does this work? First, a switch is wired to In3. Flipping the In3 switch low (ON) triggers running the 3 motion profiles, then when you flip the switch to high (OFF), the motor gracefully stops because of pin 3 mode 12. Same pin, but two different functions. If you want to run a fixed distance, 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 valuable details.

Errata and notes

08/04/15 v5.07 Dir CNC command changed to Files. If Autohome enabled, Home now does not run after every run. Autohome does not clear until @@ is typed. Fixed CNC Go command ignored +/- values.

07/02/15 v5.06 (5.03-5.05 are almost identical to 5.06). Fixed Gcode erase command.

12/09/14: v5.02 corrected many manual details. CNC mode read was 'r', now 'R'. Added Joystick Center runs Profile 30. Acceleration now 1, 2, or 3.