Introduction and Key Features

The *Faster Than Fast*^{R} Excitron **FTFc15** stepper motor controller/drivers make your motion control simple. We pack years of design into a very small case, and fully integrate it with the motor. It contains all the electronics and power for thousands of motorized applications—and they run right out of the box by simply typing "w" then a "G..

This document applies to the **FTFc15** Excitron controllers, which use a standard robust DB15 standard connector. Each motor size has a particular **FTFc15** attached. See the individual motor controller drawing sheets for additional details.

The **FTFc15** will perform all calculations and provide the intelligence to automatically step the motor accordingly. It also contains all output power drivers for the motor. No additional electronic device is needed to control and drive a stepper motor.

You have the freedom to enter ***simple*** motion parameters to create ***ingenious*** stepper motor motion profiles. Serial communications and control use ASCII (alphanumeric) characters, thus no programming is required; all you need is a basic serial interface program like Hyperterminal.

Your biggest goal is stepping the motor from point A to point B in a time period that fulfills your performance requirements. The **FTFc15** quickly 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. Just ten minutes is all it takes.

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, SPS, or torque. Most step motor documents state that bipolar motors produce more torque than unipolar motors--that is not true with our motors and controllers. Our proprietary circuitry was 1st developed and patented in **1976.** Improvements continue steadily to maximize motor and controller efficiency, which maximizes torque.

Three modes of operation are available: Command, Motion Profile, Driver, and Input Profile.

A full set of parameters for stepping the motor is called a motion profile. Twenty-four motion profiles are stored and retained, even after power is removed. A basic motion profile consists of acceleration, stepping at a constant maximum SPS (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. Input pins add versatility.

Some key parameters are acceleration, maximum SPS (steps per second), number of steps, direction, and torque control. The **FTFc15** uses unique current limiting circuitry for high efficiency and performance. By adjusting the acceleration, top SPS, and other factors, you can minimize the time required for a particular motion. A full range of acceleration is provided for any application: dog slow to extremely fast. SPS can be changed on the fly while the motor is stepping.

Various input devices, such as limit switches and potentiometers, may be connected to the five CMOS TTL +5v logic input pins. One input pin accepts analog 0-5 V and converts it to a digital value. The **FTFc15** has 1 TTL/CMOS output that you may use to control relays, LEDs, and equipment (proper interfacing is required). A single LED provides visual feedback that power is applied, and it also indicates motor steps or serial communications.

One key unique feature is the extremely low supply power for the **FTFc15** electronics. With a 24-v power supply, the **FTFc15** only dissipates 0.25 watt, which is insignificant. All internal voltages are self-generated and thus require only a single voltage supply. Built-in temperature sensing on 57, 86, and 110mm series controllers safeguards the **FTFc15** from thermal burn out, the limit is 170 degF. When mounted on or near the motor, this temperature sensing also protects the motor.

Measuring motor amperage is a thing of the past. The **FTFc15** controls and drives any unipolar stepper motor, even our FTFm110-201—a 26-pound Super-size 42! The power supply voltage range is 10 to 30V, a voltage range of 36-48 is optional.

Many options are available. All **FTFc15**'s are upgradeable to new features in the future via firmware updates, thus protecting your investment. Custom firmware applications are also available, just ask the friendly Excitron staff.

What's New

100 amp rated controller for 86 and 110 series.

USB to serial adaptor--auto-connects as a COM PC port.

Greater versatility and intelligent motion with **Input Profile Mode**. Any input pin can individually select 3 motion profiles to run. This "random" motion selection is available in Command, Motion Profile, and Driver modes.

FTFc15 controllers for 57mm and larger motors have two onboard push button switches that run **2** motion profiles each--great for manual rotation or jogging.

With the **f**, **F**, and **m** commands, you create a plateau of stepping at a minimum sps at the beginning, end, or both, of your motion. Also run in a pseudo-sinusoidal motion with the little **s** command.

Some units manufactured since 09/01/05 have a simple built-in help, typing **?** displays serial command details.

Setup, Configuration, and Power

All Excitron motor controllers use TTL (+5v logic) for the UART serial port, and do not use RS232 +-10V levels. This means that a separate RS232-TTL cable adaptor, which contains the electronics to convert RS232 to TTL, must be purchased (see Accessories web page). Damage will result if you plug in any +-10v RS232 cable.

Unless you connect with TTL, you will need our special RS232-TTL serial adaptor. If you order the **FTFc15** and motor with this RS232-TTL adaptor, it is pre-soldered for you. You will also need a DC switching regulated power supply, from +10V to +30v, with sufficient amperage to drive your step motor. These and other accessories are also available on our web site. The TTL serial UART transmit is DB15 pin #1 and receive is DB15 pin #2.

For software, a PC computer running Hyperterminal or an equivalent serial port communications program is required. The +5 volt logic serial port operates with 8 bits, no parity, 1 stop bit, and 57,600 baud. Protocol is none, no hardware handshaking, and just ASCII alphanumerics. The **FTFc15** will echo all characters received, therefore turn off Echo in your communication program. Once setup in standalone mode, the RS232-TTL adaptor is no longer needed, if you like.

The first command after power up or reset must be a **w** (little w). This ***wakes*** up the **FTFc15** and it displays:

```
Excitron Corp  10/25/05 v2.80
```

The **FTFc15** is now ready to accept serial commands. The command prompt is '>'. The most informative command is **i**, which displays all motor parameters and the selected motion profile. Commands may be typed in any sequence and the **FTFc15** remembers your last typed values. See the Motion Profile section for how to save your values.

Use the little **c** command to access these commands:

b=57600: select baud from 19,200, 38,400, or 57,600 (default). Type just the 1st digit of the baud (1, 3, or 5), and immediately the baud is changed.

d Driver mode, see Driver section for details.

e echo received serial port characters: 0 = off, 1 = on

s Simple Serial Bus (little **s**)—assign A-Z address.

q quit and returns to main command menu.

Simple Serial Bus allows connecting multiple **FTFc15** controllers with one host/PC serial line. The simple protocol is: **STX-address byte-data bytes-ETX**. STX is ctrl-B and ETX is ctrl-C. A master (normal) controller address is M. Allowable slave addresses are A-Z except M. A slave responds to this protocol only when a matching address is received. You wire the host's transmit to the Receive of one controller, it's Transmit to the next controller Receive, and so on, and the last controller transmit is wired to the receive of the host. See **FTFc15** Connections sheet.

A high amperage 24" 18 AWG power cable with a power supply filter is now included with every motor/controller. The power supply wires should be minimum 20 AWG stranded, 18 AWG is best, and as short as possible. 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.

Since our stepper motors and controllers can be operated at 2 to 3 times the rated value for intermittent operation, the power supply watts could be increased by the same amount. If the power supply amperage is inadequate, the **FTFc15** may pull the power supply voltage low enough to cause a reset of the **FTFc15.** This is harmless to the controller, but disrupts your motion profile and operation stops. If you operate in Motion Profile mode (described later), then the **FTFc15** will begin anew at the first motion profile selected by the **U** command.

Sending Commands to the FTFc15

Commands are sent to the **FTFc15** using any serial port communications program. This section describes the protocol of sending commands via keyboard or your host computer.

The format of **FTFc15** commands is to simply 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 **FTFc15** 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 sent, the **FTFc15** will send a LF (line feed) & CR (carriage return), a 4.5 msec delay, and a prompt '>'. This provides feedback to host computers that the command is complete.

Do not type "Enter" key (CR) after typing your command, type only the command and any characters as specified by the each command. This technique is similar to web page input and word processing programs where you do not type 'enter" after typing a word or set of numbers. Note that the ASCII numbers are fixed field format. This means the exact required quantity of alphanumerics must be typed. For example, to enter a step quantity of 188 steps, type:

N (**FTFc15** sends "=0033000: ") then **0000188**

All 7 digits must be entered. If you enter less than 7, 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 saved. Default values are not changeable. If you send extraneous alphanumeric 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.

Power-ON Reset

The **FTFc15** has sophisticated power-on reset and brownout protection circuitry, and the user requires no further action. This protects any operations from power supply glitches and brownout voltage conditions. The following information provides more description of the robustness and integrity of the controller.

Pin #4 is a RESET pin, normally high. To reset the **FTFc15**, send a logic low-level control to this pin, and release. After any reset, the Brake value is forced to 000 for Command Mode only, and the saved ee 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 controller with the motor's last rotor position. The high value filter capacitors in the **FTFc15** provide considerable power supply voltage spike attenuation. After power is turned off, they also cause a 1-4 second hold-up time, and the LED is on. Wait briefly before unplugging.

Software Reset

@ software reset, the **FTFc15** starts fresh, as if a power-on reset occurred. Use this command to terminate Motion Profile and Driver Modes. A little **w** command must then be issued to wake up the controller.

Input/Output Pins

The **FTFc15** has 5 inputs and 1 output on the DB15 connector, and all pins are TTL/CMOS compatible at 0 to +5 volts. You read the pin values with the **J** command:

```
 In3 In5 In7 In8 In9  Out6
1018  1   1   1   1     0
```

The IN# matches the DB15 pin number. Each input pin is normally high (+5v) 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 pins have a 200K (+- 30%) pull-up resistor connected to +5 volts, except analog pin # has a pull-up resistor of 3.9K (+- 0.5%). Output pin #6 is normally low (0 volts) and can source 1 ma or sink 10 ma. Do not exceed these current levels, or the controller may reset. Note that the actual level of the output pin is read, and it may be different from your intended value. IN3 measures analog input voltage levels, such as potentiometers, with 10-bit resolution. The display is the digital representation of the analog value. The full range is 0000 (zero) to 1023 (+5V). If used for digital switch inputs, then any value near 0000 is a digital low while any value close to 1024 is a high. Use a 25K ohm potentiometer connected to IN3 and ground.

You specify the pin number *and* the pin mode with the **p** command (little **p**). Two inputs are expected with this command, for example:

p=3: **5** mode=01: **13** 1st value is connector pin number (0 means to ignore any pin and its mode). 2nd value is the mode, ranging from 00 to 31.

Allowable pin numbers are 3, 5, 6, 7, 8, and 9. Modes are:

00 – wait until pin is low, then step
01 – wait until pin is high, then step
04 – IN3 analog input pin changes SPS prior to motion profile, runs when analog value > 0050 (threshold)
07 – joystick, step if IN7 or IN8 is low, stop when high
08 – Simultaneous joystick mode 07 and analog mode 04
12 – step until pin goes high, then stop smoothly
13 – step until pin goes low, then stop smoothly
28 – OUT6 output pin is high at start, low after motion profile
29 – OUT6 output pin is low at start, high after motion profile

Modes other than these shown here are reserved for future use. Some modes and options will obviously eliminate some pins as general input or output. Some very interesting and versatile combinations of stepper motion can result from using these modes and profiles. You can use a motion profile for: logic only; to wait for two inputs to occur; for extended time delays; or other ingenious reasons.

Input Profile Mode

Use the **I** command to enable the input pins to randomly run a set of 3 motion profiles. This gives you great standalone operation. For example, 5 push buttons can select any of 5 positions for .10, .50, 1.00, 2.00, and 5.00 inches of travel. Then you can move a precise distance by any combination of button pushing.

I=12: **nn** enable the 1st motion profile set of 3.

I	runs profiles	Pin #	Enables pins
06	06-08	IN3	3, 5, 7, 8, 9
09	09-11	IN5	5, 7, 8, 9
12	12-14	IN7	7, 8, 9
15	15-17	IN8	8, 9
18	18-20	IN9	9 only

Input pins are active ***low***. Setting **I** = 06 allows all 5 input pins to trigger each associated set of 3 motion profiles. Setting **I**=15 makes motion profiles 15-17 run when IN8 is *low, **and*** enables the higher range of profiles 18-20, which will run if IN9 is low. In this case, IN3, 5, and 7 will not trigger any profiles. Once triggered, the 3 profiles run to completion, even if the input pin is raised high, so use caution.

If you only want to run 1 profile when the input goes low, just set the remaining 2 profiles with **N**=0000000 and **t**=00000 so that they will be ignored. You may also use any input pin with any pin mode, as described above, but some combinations are illogical. Driver Mode uses IN3 and IN9 for step/direction, so do not use Input Profile Mode 06.

Setting Motion Parameters

All motor parameters required for sophisticated motion control of your motor are described here in alphabetical order. Note that you type only the characters shown in **bold**. The **FTFc15** displays the equals sign, the existing value (from 1 to 7 numbers), colon and a space (most commands).

A=0100: **nnnn** First motor step SPS, range of values is very slow (0002) to extremely fast (1000.

a=18: **nn** 2nd acceleration value to fine tune after 1,000 SPS, 00 is slower and 63 is faster acceleration.

C Clockwise motor direction.

M=0: **n** Micro stepping, full step is 0 and half step is 1.

N=0012000: **nnnnnnn** Number of steps. A value of 0000000 is for motion control using inputs, without any time delay or stepping.

r=006: **nn** Repeat quantity, from 001 to 250. The remaining repeat value is output *after* each motion, except the last.

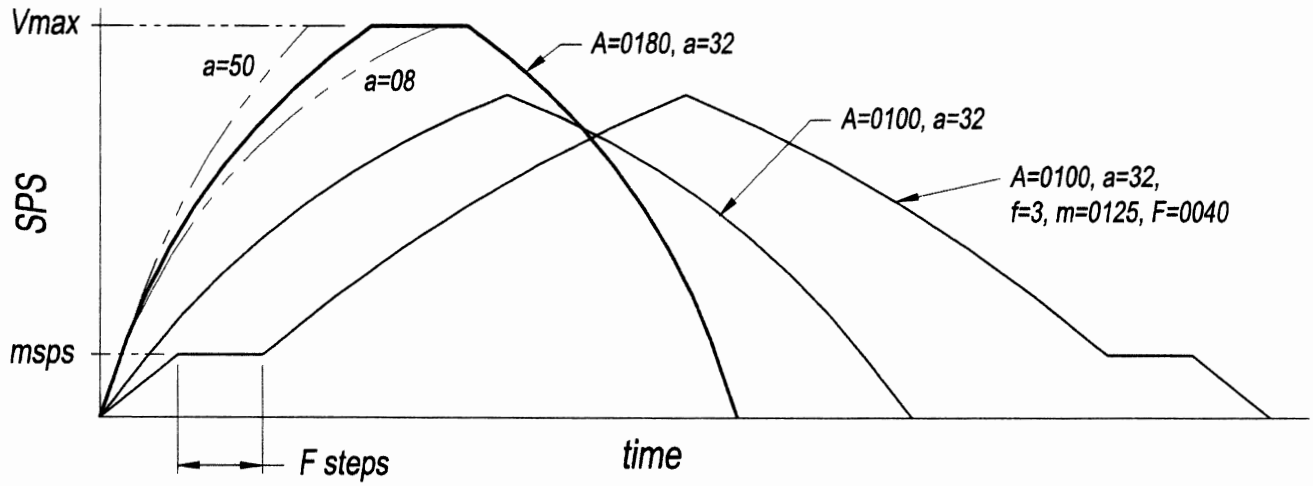
t=-/+00888: **nnnnn** Time delay either before (-) or after (+) the motor motion. Input - or +, then 5 numbers representing milliseconds, e.g., **t=+00500** defines a 500 millisecond time delay after the motor motion.

V=08600: **nnnnn** Maximum SPS. Note that the step quantity and acceleration values may define a motion such that the maximum SPS will not be achieved. Slower **A** command values also self-limit the maximum SPS.

W Counter-clockwise motor direction.

f,F,m 3 input values that set a flat speed portion at the beginning, rear, or both of the speed profile.

The following graph shows examples and variations of acceleration **A** and little **a**, and the use of **f**, **F**, and **m**. Note that deceleration is always a mirror image of acceleration.



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). The **FTFc15** 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 controllers. Most step motors are rated for 2 to 5 volts; so current limiting allows them to be connected to a 10 to 30 volt (24 volt being most popular) power supply. You control precisely the amount of torque in the motor with one simple command:

T=012: **nnn** sets the internal value for motor torque.

A low value for **T**, such as 001-002, produces low torque. 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 **A**, **a**, and **V** first. And ***do not*** increase **T** to make the motor run faster. At higher speeds the T is not effective, yet it is effective as the motor spins up and during deceleration. See the individual motor data sheets for further guidance. After determining the suitable values, adjust **T** up or down for best performance vs. heat build up. The maximum allowable **T** is set for each motor size, and is displayed with the ***i*** command. The **Tmax** value will produce 2X the rated amperage at 24 volt, and 1X the rated amperage at 12 volt.

After a certain SPS, the **FTFc15** 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 degree C in ten 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 3 times its rated power, depending on many factors.

Warning—stepper motors require amps of current, and over heating and damage can quickly occur. Start with low **T** values, then gradually increase while monitoring temperature via the *i* display. Maximum operating temperature is 75 degrees C (centigrade), and is monitored while the motor is running (not on small controllers). Selecting higher torque than allowed may damage the **FTFc15**. You are responsible for any damage to the 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. Note that this terminates Motion Profile or Driver Modes, and the **FTFc15** controller is then in Command Mode.

Stepping the Motor

You step the motor with the **G**, **g**, **S** and **s** commands:

G Go step the motor until the specified number of steps are reached.

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.

s Sinusoidal motion (cyclical)--step continuously only if **f** is 2 or 3. This command ignores the step number entered. Only a motor stop command **O** or **o** will stop the motor.

After you type **G**, **g**, **S** or **s,** the **FTFc15** sends "U,D,O,o: " to indicate your valid command choices.

U Speed up the motor on the fly. The SPS increment is determined by the **V** value. Any U or D command does not update the original SPS value, except during Mode 04.

D Slow down the motor on the fly. The SPS decrement is determined by the **V** value.

O Starts deceleration gracefully and the motor slows down according to the *acceleration* factor you specified.

o Motor stepping is stopped immediately. The motor's inertia may cause erratic motion for a few steps.

Increasing Vsps on-the-fly with **U** may cause lost steps when stopping because there are not enough steps left to decelerate properly.

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 1 to 70 produce little braking. Always increase braking values slowly. If you do not need braking torque, specify zero braking current by typing:

B=000: **nnn** Set the braking (holding) current limiting parameter. Range of values is 000 to 138.

Power saved by minimizing braking results in greater torque available for motor stepping. Higher current results in higher temperatures. **B**=000 also reduces the **FTFc15** 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, careful.

Command Examples and Notes

You wish to rotate the motor for 10,244 steps clockwise, half stepping, and a maximum SPS value of 1,800. Send:

N=0000080: **0010244**
C
M=0: **1**
V=04200: **01800**
G U,D,O,o:

Note that you do not type '=' nor "enter" with any command, the **FTFc15** provides it. The '>' prompt is not shown here for clarity .The order of the **N, C, M,** and **V** commands is not important, the **G** command is, since it starts motor motion. The **M, N** and **V** commands must receive their required exact digit quantity. To step the motor with the same parameters except CCW (counter clockwise), type:

W
G U,D,O,o:

Note that the other commands do not have to be re-entered. The **FTFc15** remembers the last values of all parameters and commands. Be aware that the motor and controller will heat up faster while running at slower speeds. At SPS over 3,000, depending on power supply voltage, the motor, etc, the motor self-limits the power, and should run cooler. A temperatures over 50 degree C may feel hot to human touch, but that is ok for motor and controller operation (up to the 75 degree C limit).

Acceleration, maximum SPS, torque, and other parameters can be optimized for your application. An acceleration value of 0200 is a good starting point. The **FTFc15** 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.

Stepper motors have a natural resonant frequency around 200 to 1200 SPS. If stepped at constant SPS in this speed range, the motor may lose steps. This condition is worse at higher torque settings. It is best to accelerate through this range. A larger **T** value at these speeds increases resonance and mis-stepping. Usually running half-step (**M**=1) helps. When changing from half to full stepping, the controller may shift a half step to always be on a full step.

Warning! Use caution when operating, severe injury can result from the motor rotating. Long wires act like antennas and may cause erratic dangerous motion.
The **FTFc15** stores all motor stepping parameters for each of the 24 motion profiles. You can view and edit these profiles individually, and then set them to automatically execute from memory upon power-up. The motion profiles are stored in EE memory, and thus retain their values when power is off.

Motion Profiles

The major purpose of motion profiles is to allow a standard programmed **FTFc15** to operate as a standalone unit, not needing a PC or external controller. Once set up, only a power source and power wires are needed. With the **Q** command, you specify any of the 24 profiles as the starting profile, and a subsequent profile as the last one, i.g.,profiles #05 through #12. The **FTFc15** will sequentially run these motion profiles, and automatically repeat after the last one is completed

You can easily create your own motion profiles by entering values for acceleration, SPS, etc, or by retrieving an existing motion profile. Thus custom models of **FTFc15** are not necessary.

You can also create 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.

The commands associated with saving, retrieving, and displaying motion profiles are: **P, Q, U,** and **v** (little v). For each of these commands, simply type the letter of the command. The **FTFc15** responds with a '=', then you type the required numbers. If you issue a **P, Q,** or **v** command, changes to current parameters are lost, unless you save them first by using the **U** command.

P=nn Display the motion profile specified by **nn**.

Q=xx yy Quit command mode and start motion profile (standalone) mode. Thereafter, at power up or a reset, the **FTFc15** will automatically run the motion profiles you specify with the **xx** and **yy** numbers.

U=nn Saves all of the current motor parameters to the motion profile specified by **nn**. Example: **U=03** will store the values as motion profile **03**.

v View all 24 motion profiles. The **FTFc15** displays a header line, followed by 24 data lines.

Special note for the **U** command: it stores the ***last*** **G**, **g**, **S** or **s** value executed, into the motion profile you specify. There is no other command to store the **G**, **g**, **S** or **s** value.

With a few simple command settings, you can create some very interesting motion profiles. For instance, if P #01 is CCW and set for mode #13, P #02 is CW and set for mode #12, and you have a switch connected to an input pin, then after you start motion profile mode with the **Q** command, the switch position determines direction of rotation!

You stop the **FTFc15** from executing motion profiles by typing the reset command **@**. Then a little **w** to enter command mode. The **FTFc15** will execute motion profiles when you again issue the **Q** command. In the normal command mode, you can change any motor parameters and then save this modified profile, overwriting any existing motion profile.

The motor stepping parameters may also be downloaded 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, easily created from MS Excel or Notepad, is:

A0200,V04400,B000,N0080000,U12

Note that the order is not important, except the U12 command is last, which saves the input values to Profile number 12. You may also have CR and/or LF after each input command. If you send multiple lines, you must insert a 300 millisecond delay after each line to allow the **FTFc15** to finish writing.

The **FTFc15** 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 **ee** writes require 10ms per character to save data in the controller. When running motion profiles, the controller outputs the profile number and a space, followed by the repeat number(s) (counting down) and a space, if echo is on. The life expectancy of the EE memory is 100,000 write commands (**Q** or **U**). Do not exceed this number.

Driver Mode

You set Driver Mode for CNC operation by typing little **c** then little **d**. Driver mode is when external signals, from a PC parallel port for instance, fully control steps and SPS. The **FTFc15** steps the motor using 2 external input signals:

DIR DB15 pin #3, 0 is CW, logic 1 is CCW
STEP DB15 pin #9, negative going pulse, > 25 usec

The **FTFc15** is CNC-ready, and runs with almost any CNC PC or PLC program. An enable signal is not needed because the **FTFc15** will automatically stop when the STEP pulses cease. The **FTFc15** serial port is not used for STEP and DIR. See the CNC Diagram for more details.

Internal current control, temperature, Full/half step, and Brake functions are maintained automatically by the **FTFc15**, using the values of Motion Profile #01. Only a reset **@** serial character terminates Driver Mode and enables Command mode. The 2 manual switches operate during Driver Mode for ease of jogging.or other manual positioning. Since Driver Mode uses IN3 and IN9 for step/direction, do not use Input Profile Mode 06.

Since it is possible to step the motor using an input pin or the manual switches, do not send CNC step pulses while the motor is already stepping.

Manual Switches

Your **FTFC15** may have two onboard button switches **Sw1** and **Sw2** for manual rotation, and they run 2 Motion Profiles each: #21-22 and #23-24 respectively. You can set the number of steps, direction, time delay, and other parameters for each switch. "Sw1" or "Sw2" is sent to the serial port when pushed, unless echo is off. They are very useful, especially for robotics or CNC machines. For example, you can easily set SW1 for exactly 1 revolution CCW and SW2 for 1 revolution CW.

Pushing a switch continuously results in automatic repeat of its 2 Profiles. When you release the switch, the motor decelerates and stops. Either switch is de-activated if the motor is already running. The repeat value **r** must be 001. Pushing both switches simultaneously causes an emergency stop.

With the 60 or 100 amp option on the 86 or 110 series, a 3rd manual switch **Sw3** is available for custom applications. Standard configuration for **Sw3** is that it runs Motion Profile #03 to completion once triggered.

Unless you saved your changes, pushing any switch will erase unsaved changes.

Help Command

On some **FTFC15** controllers (Excitron's manufacturing discretion), typing **?** displays a simple serial command help.

Additional Command Details

The **i** command provides important information. Here is an example:

```
Excitron Corp.  10/25/05 v2.80                    ← firmware date and revision

s  baud echo In Tmax degC    Vs
M 57600    1 00  028  023 24.14

P# D M G  Acc a2 Brk  Number  Vsps Trq   msec rep pin/mode f Fnum msps
01 C 0 g 0200 50 000 0004800 03600 012 -01000 004   3    01 0 0100 0125
```

The 2nd & 3rd lines show: **s** is M (for master, slaves are A-Z); serial port baud; echo 0 (off) or 1 (on); I is the 1st motion profile set of 3 for Input Profile Mode, **Tmax is the** maximum T for command entry—NOTE: the safe *ALLOWABLE* T value is lower then Tmax--depends on voltage, use caution. Controllers for motors larger than 42mm measure temperature and voltage: **degC=** degree centigrade of the **FTFc15** (and motor if assembled together), **Vs=**power supply voltage.

The last two lines, which are also displayed after the **i, P** or **v** commands, are described below. Note that this display also defines 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	Command	Description
P#	P	profile number, 01 to 24
D	C or W	direction of rotation, C(lockwise) or W(counter clockwise)
M	M	micro stepping, 0 (full) or 1 (half)
G	G,g,S,or s	type of motion—**G** (single), **g** (reciprocal), **S** (continuous), **s** (pseudo-sinusoidal)
Acc	A	1st SPS value, determines acceleration, 0002 to 1000
a2	a	2nd acceleration value , 00 (slowest) to 63 (fastest)
Brk	B	brake factor, 000 to 138—use care with braking values—128 is full on!
Number	N	number of steps, 0000000 to 9999999 (for larger values, use **S** command for continuous stepping)
Vsps	V	maximum steps per second, 00002 to 12000, limited by **A** value
Trq	T	torque value (non-dimensional), 001 to 250 max—use carefully! Motors have different preset maximum.
msec	t	delay time in milliseconds, before (-) or after (+) the motion profile, 00000 to 65000
rep	r	number of times to repeat this motion profile, 01 to 250. While running, the remaining number is output.
pin/mode	p	DB15 pin #3, 5, 7, 8, or 9 to control this motion profile (0 means to ignore). Then enter mode number, 00-31.
f	f	section for running at a flat sps, 0 = ignore, 1 = begin, 2 = end, 3 = both. (for end, Acc < msps)
Fnum	F	number of steps to run at minimum sps, according to "f", add to N to get total steps, minimum is 0002.
msps	m	minimum sps for running at beginning or end, minimum is 0002, cannot be more than V (sps)

Enjoy! Please review other available documentation for additional details. Feel free to contact us at info@excitron.com.

Options

- 60 or 100 amp power for controller in 86 and 110 series.
- RS232-TTL adaptor—for connecting to PCs, PLCs, and other equipment using +-10 volt serial lines.
- USB – add-on module for serial communications
- RS485-TTL adaptor—for connecting to PCs, PLCs, and other equipment using 5 volt differential serial lines.
- Firmware modifications--to reduce or eliminate external electronics, such as PLCs or computers.
- Rear shaft, knob, and optical encoders from 1 to 2000 positions per revolution..
- Wireless link – remote control up to 1,000 feet.

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. **Use caution when operating--severe injury or death can occur.**

v2.74 Tuned sps internal curve for higher sps and smoother running. Typing 'O' will terminate input pin modes. .

v2.72: Changed ***i*** command heading. Added echo command.