

Stepper Motor Driver

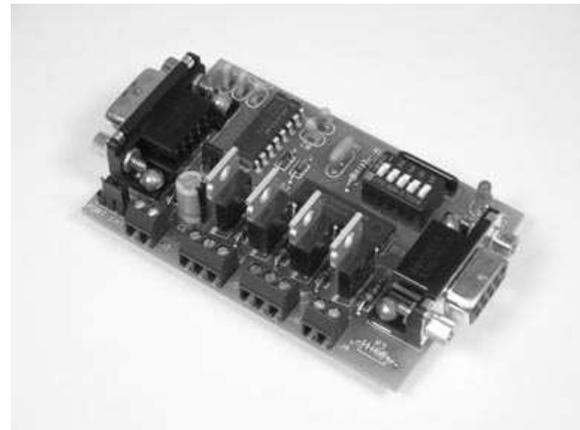
FEATURES

- Drives a unipolar stepper motor rated between 8-30 VDC, and up to 4A.
- DIP switch addressable; stack up to 32 modules on the same port for multiple axes.
- S-curve acceleration and deceleration slope profiles provide smooth start/stop motion.
- User programmable ramp rate, velocity, and idle reduction current.
- 24-bit absolute motor position counter.
- Normally-open limit switch input.
- Supports single phase, dual phase, and half-step drive modes.
- All user configuration data and motor position counter stored in non-volatile memory.
- Industry standard RS-232 interface. Meets all EIA/TIA-232E and V.28 specifications.
- Screw-terminal connectors used on all inputs and outputs.

DESCRIPTION

Connects to the RS-232 serial port of a PC, laptop, or other host. Directly drives a unipolar stepper motor using precise positioning and tracking algorithms. Simple coordinates sent from the host will advance stepper motor to an exact position in the range of 0 to 16,777,215. Host can also instruct motor to move in either direction until a limit switch has been triggered, or use a basic single-step mode which is host incremental.

S-curve acceleration/deceleration slope profiles are automatically generated and incorporated into any multiple step sequence, independent of host. Thus reducing stall conditions during acceleration, and overrun conditions during deceleration. Ideal for use in cost sensitive, open-loop, precision motion control applications requiring a minimal user learning cycle.



SPECIFICATIONS

Drive Type	Quad, open-drain MOSFET
Drive Current	4A maximum
Idle Current	PWM, selectable in 10% increments
Step Rate	Up to 1250 pps
Limit Switch	Normally-open, direction sensitive
Processor	PIC16F628
Clock	4 MHz
Communications	9600 Baud, N, 8, 1
Power Requirements	+8 to +30 VDC
Current Draw	9 mA, plus current drawn by motor
Operating Temperature	-20°C to +80°C
Board Dimensions	3.1" x 2.0" x 1.0"
Weight	1.9 oz

TABLE 1: ADDRESS SETTING

HEADER CHARACTER ASCII (HEX)	DIP SWITCH SETTING
	1=on, 0=off
	1 2 3 4 5
A (41)	0 0 0 0 0
B (42)	0 0 0 0 1
C (43)	0 0 0 1 0
D (44)	0 0 0 1 1
E (45)	0 0 1 0 0
F (46)	0 0 1 0 1
G (47)	0 0 1 1 0
H (48)	0 0 1 1 1
I (49)	0 1 0 0 0
J (4A)	0 1 0 0 1
K (4B)	0 1 0 1 0
L (4C)	0 1 0 1 1
M (4D)	0 1 1 0 0
N (4E)	0 1 1 0 1
O (4F)	0 1 1 1 0
P (50)	0 1 1 1 1
a (61)	1 0 0 0 0
b (62)	1 0 0 0 1
c (63)	1 0 0 1 0
d (64)	1 0 0 1 1
e (65)	1 0 1 0 0
f (66)	1 0 1 0 1
g (67)	1 0 1 1 0
h (68)	1 0 1 1 1
i (69)	1 1 0 0 0
j (6A)	1 1 0 0 1
k (6B)	1 1 0 1 0
l (6C)	1 1 0 1 1
m (6D)	1 1 1 0 0
n (6E)	1 1 1 0 1
o (6F)	1 1 1 1 0
p (70)	1 1 1 1 1

STACKABLE DATA MODULES

All modules in this series incorporate two EIA/TIA-232E serial ports which communicate at 9600 baud, no parity, 8 data bits and 1 stop bit. DB9 connectors are jumpered to satisfy hardware handshaking. The port labeled "HOST" is configured as a DCE device and should be connected to a PC's serial port. The port labeled "SLAVE" is a DTE device and can be left open, or connected to another module's host port. Up to 32 modules can be chained together in this fashion to form a network. Either plugged together end to end, or separated by a cable. Because a module contains two individual bi-directional ports which pass data through, it also acts as a repeater, extending the total allowable length of the RS-232 communications line.

A modem can serve as the host for remote operation, but since a modem uses a DCE port, a "null modem" adapter must be placed between the modem and the data module's host port. A gender changer may also be required. In addition, any hardware/software flow control must be disabled in the terminal program.

Each module in a network should be set to a different address using the on-board 32-position DIP switch. A module will only respond to data packets that begin with its' own unique header character, which is determined by this DIP switch setting. See Table 1. Data packets transmitted by a module will also begin with this header character. The host PC can use the header character to address each individual module in a network, and to identify a module which is talking.

COLLISION CONTENTION

The utilization of the communications line can be thought of more as a single, bi-directional, data bus, operated in a multi-drop mode rather than a standard RS-232 data link. A transmission from a data module travels in both directions, upstream to the host, and downstream to signal other modules that it has seized the line. Before transmitting, a module will listen to the communications line and wait for quiescence. After a silent period equal to the length of one byte, the waiting module will send its data packet using a Carrier Sense Multiple Access with Collision Detection communications protocol. See the application note (AN100) at the back of this manual for more details.

COMMAND SET

The host PC communicates with the Stepper Motor Driver Module using a command set comprised of standard ASCII character strings as depicted in Table 2. Each command must be preceded with the header character determined by the DIP switch setting on the circuit card (see Table 1), and terminated with a carriage return.

MOVE - Moves the stepper motor to a specific coordinate within the range of the 3-byte motor position counter. After reception of this command, the direction of rotation will be determined by comparing the target position to the current motor position. The motor will then ramp up to the speed determined by VELOCITY using an S-curve acceleration profile and remain there until reaching the pre-calculated point of

deceleration. An S-curve deceleration profile will then be used to bring the motor speed back to zero as it approaches the target position. The motor position counter will be updated accordingly.

HOME - Moves the stepper motor in a specific direction until a limit switch is triggered. After reception of this command, the motor will ramp up to the speed determined by VELOCITY using an S-curve acceleration profile and remain there until detecting a trigger from a limit switch. Following the trigger, the motor will continue to step until the value of the index register runs off, and then an S-curve deceleration profile will bring the motor speed back to zero. The motor position counter will be set to 0 if rotation was negative, or 16,777,215 if rotation was positive.

STEP - Moves the stepper motor one step in a specific direction. The motor position counter will

TABLE 2: COMMAND SET

TITLE	COMMAND	DESCRIPTION
MOVE	M <i>pos</i>	Move stepper motor to a specific position (<i>pos</i>) at rate determined by VELOCITY using acceleration and deceleration curves. <i>pos</i> = 0 to 16,777,215. (Note 3)
HOME	H <i>dir index</i>	Move stepper motor in the specific direction (<i>dir</i>) at rate determined by VELOCITY using acceleration curve. Rotation will continue until limit switch activates index runoff and deceleration curve. <i>dir</i> = + or - <i>index</i> = 0 to 255 If <i>index</i> omitted, uses default of 0. (Note 3)
STEP	S <i>dir</i>	Move stepper motor one step in a specific direction (<i>dir</i>). <i>dir</i> = + OR -. (Note 3)
VELOCITY	V <i>value</i>	Sets the pulse-per-second rate used in the MOVE or HOME function. <i>value</i> = 1 to 125, multiplied by 10. Default = 50 (500 pps). (Note 3, 4)
RAMP-RATE	R <i>value</i>	Sets the ramp rate used in the acceleration and deceleration curves. <i>value</i> = 1 to 255. Default = 50. (Note 3, 4)
POSITION	P <i>value</i>	Modifies the motor position counter. <i>value</i> = 0 to 16,777,215. (Note 3) If <i>value</i> is omitted, reads current position. Returns 0 to 16,777,215.
EXCITE	E <i>value</i>	Sets the driver excitation mode. <i>value</i> = 1 to 3. "1" being single phase, "2" being dual phase, 3 being half-step. Default = 1. (Note 3, 4)
IDLE	I <i>value</i>	Sets the idle current (via PWM) which is used at anytime the motor is at rest. <i>value</i> = 0 to 10. Default = 10 (100%). (Note 3, 4)
ERROR	?	This character will be returned after an invalid command or variable.
RESET	!	This character will be returned after a power-on reset, or brown-out.
<p>Note 1: All command strings sent to the data module should be preceded with the header character (see Table 1), and terminated with a carriage return. All responses from the data module will also appear in this format.</p> <p>Note 2: Any spaces shown above in the listing of the command strings are for clarity only. They should not be included in the actual transmission from the host, nor expected in a response from the data module.</p> <p>Note 3: After successful execution, this command will be echoed back to the host in the same format as received.</p> <p>Note 4: If <i>value</i> is omitted, reads the current setting which will be returned to the host in the same format as above.</p>		

TABLE 3: TERMINAL / CONNECTOR DESCRIPTION

NAME	TYPE	ELECTRICAL SPECS	COMMENTS:
HOST	DB9 (female)	EIA/TIA-232E Standard	RS-232 serial port configured as DCE. Connects to host PC. Hardware handshake jumpered.
SLAVE	DB9 (male)	EIA/TIA-232E Standard	RS-232 serial port configured as DTE. Can be connected to another data module's HOST port for networking.
Power Source	Jumper	N/A	Power source selection jumper. Selects either external, or port powered. (Note 1)
+	Screw Term	+8 to +30 VDC	External unregulated power supply input.
-	Screw Term	GND	External power supply ground.
PH1	Screw Term	Up to 30 VDC @ 4 A	Stepper Motor Phase 1 Output
PH2	Screw Term	Up to 30 VDC @ 4 A	Stepper Motor Phase 2 Output
PH3	Screw Term	Up to 30 VDC @ 4 A	Stepper Motor Phase 3 Output
PH4	Screw Term	Up to 30 VDC @ 4 A	Stepper Motor Phase 4 Output
CT1	Screw Term	Tied to VIN+	Phase 1 & 3 center tap.
CT2	Screw Term	Tied to VIN+	Phase 2 & 4 center tap
LIMIT	Screw Term	N/A	Normally-open limit switch input.
Note 1: Selecting "port powered" will draw from the power supply source of an upstream data module. Caution, the USB or COM port of a PC or laptop does not supply enough current to serve as the power supply source.			

be updated accordingly and will roll-over from 16,777,215 to 0 or vice versa if reaching the ends.

VELOCITY - Sets the pulse-per-second rate used in the MOVE or HOME command, selectable between 10 and 1250 pps in multiples of 10. Depending on the particular stepper motor and its load demand, setting velocity too high will result in a stalled condition. This will cause a break down in motor rotation as well as corrupt the motor position counter.

RAMP-RATE - Sets the ramp rate used in the acceleration and deceleration curves, selectable between 1 and 255. 1 = slowest, 255 = fastest. If the ramp rate is too fast, a heavy load may cause a temporary stall during acceleration, and/or a mechanical overrun during deceleration. Either will result in a corrupted motor position counter.

POSITION - Modifies the motor position counter, selectable between 0 and 16,777,215. Can also

be used to read current position of the motor.

EXCITE - Sets the driver excitation mode. Single phase draws less current and the motor runs cooler. Dual phase produces more torque and can be used to reach higher speeds. Half-step uses both single and dual phases to produce twice as many steps per revolution.

IDLE - Sets the idle current which is used at anytime the motor is at rest. Pulse Width Modulation @ 7.7KHz duty-cycle generates the desired drive current, and is selectable in 10% increments of full power. 0 = off, 1 = 10%, 2 = 20%, 3 = 30%, etc. Note, it is not recommended to set this value to 0. Doing so may allow outside forces to move the motor unintentionally and in turn corrupt the motor position counter.

ERROR - Any data string sent from the host containing the correct header character but an invalid command or variable will be responded to with this error indicator.

RESET - Upon power-up or any other reset condition, this indicator is transmitted to the host. Note, all user configuration data and motor position counter is stored in non-volatile memory. Therefore, a reset or loss of power will not corrupt these settings. However, if a reset occurs during a move, the motor position counter will reflect the previous value prior to beginning the move function.

OPERATION

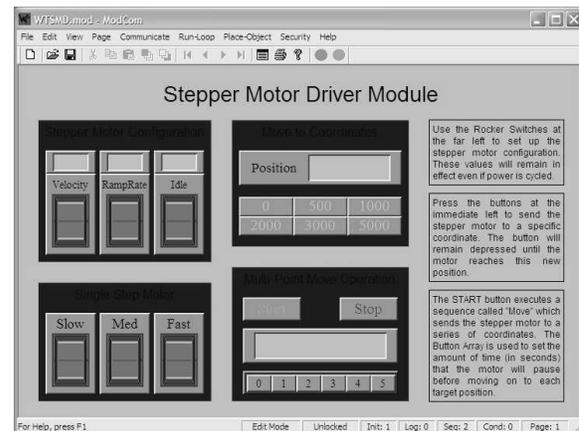
In order to connect a stepper motor to the WTSMD module, you must first identify the wires coming out of the motor housing. A unipolar motor such as that required by the WTSMD will have 6 wires (four phase wires and two center taps). And since there are no color standards, you will have to consult the manufacturer's documentation to identify each of the wires so that they can be attached to the appropriate terminal of the WTSMD.

To hook the WTSMD to a host PC, use a standard RS-232 cable with male and female DB9 connectors on opposite ends. This cable should be wired straight through (pin to pin) with no crossover of the data lines. In other words, not a null modem cable. Connect a suitable DC power source to the + and - terminals of the WTSMD. This power source should be ample enough to supply the current necessary to drive the stepper motor, and it should have a voltage rating within the range of both the motor and the WTSMD card.

When the WTSMD is first powered up, the red LED will flash briefly. This indicates that the on-board microcontroller has booted up, successfully completed its internal diagnostic test, and has transmitted the reset character to the host to signal that it is up and running. The red LED will also flash anytime the module receives or transmits any data packet, thus making it a valuable diagnostic tool when troubleshooting communications problems.

An easy-to-use Windows software package called "ModCom" is available and can be downloaded from www.weedtech.com. This program will allow the user to quickly set up buttons which transmit commands, dials and meters which automatically poll for data, and a variety of other screen objects such as slider controls, event counters and timers, chart recorders that plot the data from any screen object, and more. In addition, conditional statements can be set up to watch for and act

FIGURE 1: MODCOM APPLICATION



upon specific conditions, program scripts can be written by the user to perform complex tasks in the background, and data can be logged to a file or web page at user-defined intervals.

Once ModCom is installed and running, go to the <Communicate> menu item at the top of the screen and click on <Send/Receive>. A dialog box will pop up which you can use to type in the commands from Table 2, transmit them directly to the WTSMD, and see the response coming back. Use this dialog box to familiarize yourself with the command set and to experiment with the various features supported by the module. The experience gained here is significant since these are the same command strings you will use when setting up the other objects in ModCom.

To control a stepper motor, start with the sample application "WTSMD.mod" which can be found in the ModCom subfolder called "Samples". After this file is opened, it will appear as shown in Figure 1. To start the main run-loop, click on the green toolbar button at the top of the screen. At this time, you can experiment with various pre-built functions such as adjusting the motor configuration, single-stepping the motor, moving the motor to a number of preset coordinates, or even sending the motor through a series of stop-and-go points automatically.

To understand how this application works, first halt the run-loop by clicking on the red toolbar button at the top of the screen, then right-click on any screen object to pull up its properties dialog box. With the aid of the helps files which can be accessed by clicking on the <Help> menu item at the top of the screen, study the properties of each screen object to see how they

operate. Also read through the chapter "How to control a stepper motor" in the help files.

LIMIT SWITCH

One or more normally-open limit switches can be connected to the WTSMD and used for braking or emergency stop. Multiple switches should be wired in parallel. During the execution of the HOME command, these switches will initiate a braking action when triggered, causing the stepper motor to begin its deceleration curve. During the execution of the MOVE command, these switches will initiate an emergency stop action when triggered, causing the stepper motor to immediately stop. Note that the motor position value returned to the host (after executing a MOVE command which was interrupted by an emergency stop) will reflect the actual position of the motor, rather than the position used in the MOVE statement.

The limit switch function is direction sensitive. In other words, when the motor is moving in a particular direction and a limit switch is activated, thus stopping the motor with the switch contacts in a closed position, the motor will still be allowed to move in the opposite direction using another MOVE or HOME command. To move the motor beyond any limit switch stop point, the STEP command can be used.

MOTOR INDEXING

Most applications require the stepper motor to be moved to an exact zero position which can then be used as a reference point for all other move functions. The HOME command with its index field is provided for this very purpose.

Since the VELOCITY and RAMP-RATE will effect the distance moved after a limit switch triggers a braking action during a HOME function, the user must first set the appropriate value of these parameters before calculating an index value. To setup a motor reference point which can be used for indexing, a limit switch should be installed on the apparatus at a location which will activate the break and bring the motor to rest just short of the desired stopping point. Use the HOME command to send the motor to this position and then use the STEP command and count the number of extra steps necessary to trim the motor position to the true reference. The number of extra steps should then be used as the index value in the

HOME command which will cause the stepper motor to stop at this exact position every time.

EXTENDING MOTOR SPEED

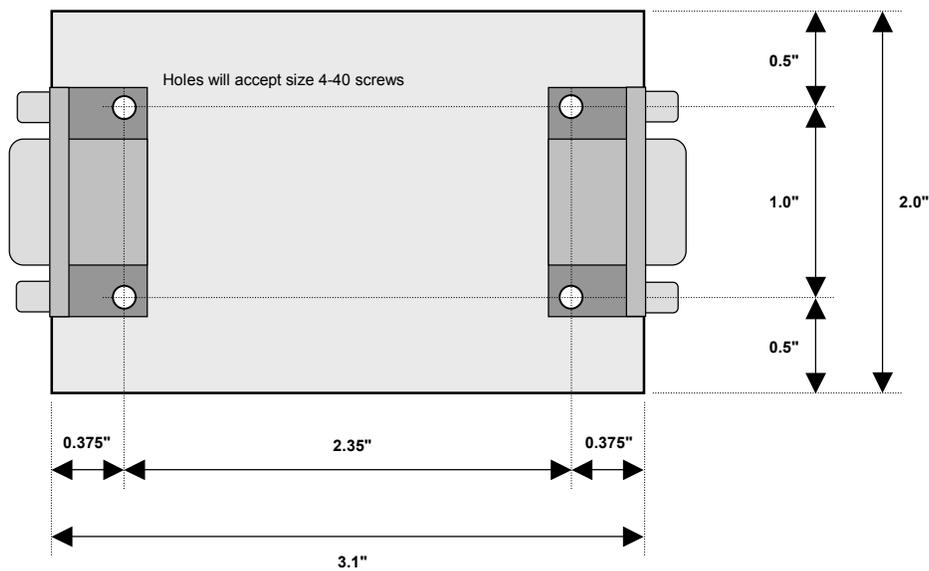
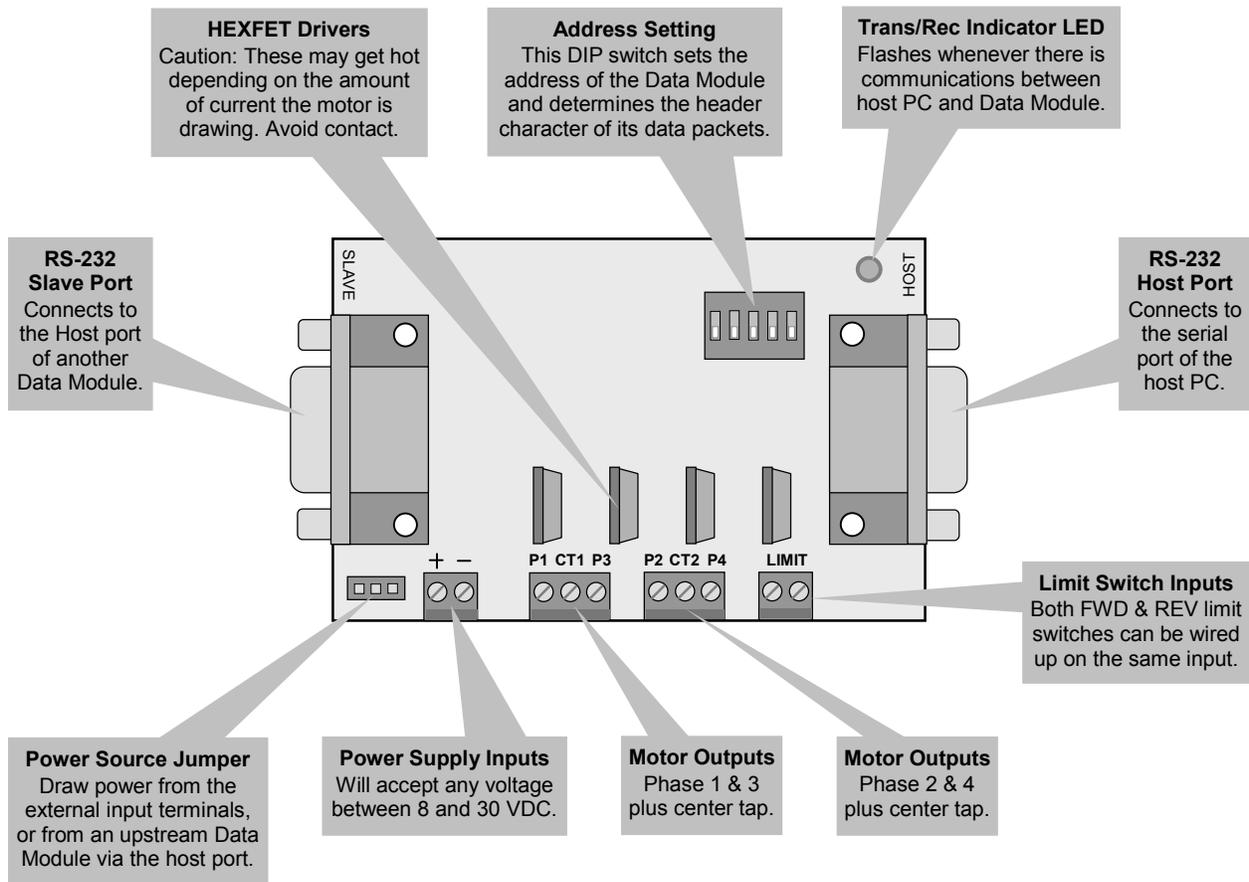
A stepper motor is driven with synchronized pulse patterns controlling the on and off states of each individual coil in the motor. Unfortunately, the natural inductive properties of a coil opposes any quick on/off state changes. This is due to the back EMF generated as the applied voltage increases or decreases. The sharper the transition (or slew rate), the more the back EMF works against it. Because the pulse patterns driving the motor coils are square waves, the slew rate is very great, thus generating large amounts of back EMF as each coil turns on or off. This back EMF, in effect, introduces a delay from the time a pulse is applied to a motor coil, to the time full current flows through it. As the pulse-per-second rate of the motor increases, there will come a point at which each coil does not have enough time to turn completely on, and subsequently, the motor stalls.

To increase the turn on rate of a motor coil, the back EMF must be minimized. This can be accomplished by adding a resistor in series with the coil which reduces the avalanche effect of the turn on voltage. This is because the actual voltage applied to the coil will now be a product of the current already flowing through it, and will increase at the natural inductive time constant without generating excessive back EMF.

This series resistor should be placed in-line with each of the motor voltage supply lines attached to the CT1 and CT2 terminals, and should be of sufficient wattage to handle the current draw by one motor coil. Also, the power supply should be increased to allow for the voltage drop across this resistor. For example, a 12V 50-ohm motor coil using a 50-ohm series resistor requires a 24V power supply.

HIGH CURRENT MOTORS

High current stepper motors will produce noise spikes on the power supply lines. If these spikes are large enough, they will cause a reset of the WTSMD. This will be evident by the host receiving a reset indication during times it is trying to step the motor. If this occurs, remove the motor voltage supply lines from terminals CT1 and CT2, and connect those lines directly to the positive terminal of the stepper motor power source.



STEPPER MOTOR DRIVER
WTSMD-M

