

Controlling Step and Direction Interface Drives

Stepper motor drivers do not have any sort of on-board functions to actually make the motors spin. They are waiting for input from some other source. An inexpensive option for simple stepper motor control is a microcontroller such as an Atmel or Microchip microcontroller, or any one of the microcontroller carriers such as the Basic Stamp or Arduino.

Stepper motor drivers need only two control signals to operate: a step signal, and a direction signal. The direction signal when at logic high (+5V) determines one direction of travel. Clockwise or Counter-clockwise will depend upon how the motor phases are wired, but it does not matter as this is easy to invert in your software to suit your needs. The direction signal when low (GND) then determines the opposite direction of travel as when high.

The motor will not spin simply by setting a direction, it requires a frequency controlled (not PWM) pulse on the step line. When the step line transitions from low (GND) to high (+5V), or high to low on certain drivers; the motor will move to the next step position determined by the step resolution setting dip switches. So if in full-step mode, the motor will move a full step; if half-step mode, half-the angle of a full-step; in quarter-step mode, a quarter of a full-step, etc. Most modern stepper motors are 200 steps per resolution in full step where a full step is 1.8 degrees.

Because motors have inertia, you cannot suddenly send a series of fast pulses to the driver for any motor speeds above a few RPM. You must ramp the pulses up AND down. Otherwise the rotor will lose sync with the magnetic field and will stall. This ramp must be smooth and your pulse train must have very little jitter in it or the motor will stall from that as well. The length of this ramp will be determined by the load on the motor.