TMC 236 / 246 / 239 / 249 Family:

SPI vs. Step Direction– a Comparison and Discussion

Trinamic Motion Control GmbH & Co. KG
Sternstraße 67
D – 20357 Hamburg, Germany
Phone +49-40-51 48 06 – 0
FAX: +49-40-51 48 06 – 60
http://www.trinamic.com
INFO@TRINAMIC.COM

1 What advantages does the SPI interface of the drivers give you, when comparing to the simpler Step/Direction interface?

In fact you will find that Step / Direction looks much simpler to use when compared to SPI on a first glance, but when taking a closer look, the assumed advantages of the simpler interface diminish.

1.1 What is the background of SPI and of Step-Direction interface?

The background for Step-Direction is industrial motion control, where you need to bridge many meters with as few as possible control lines, while providing a simple and robust motor control interface. However, for high-end controls in industrial applications diagnostics and bus structure connections are new trends.

SPI is a very simple on-board interface used for communication between microprocessor and peripherals. Typically it bridges a few centimeters. Microprocessors provide this interface in hardware since the 30 years old 8051 and 6800 family. New CPUs and microcontrollers push SPI bandwidth up to several 10Mbit per second, to allow control of high-bandwidth peripherals.

1.2 What does the SPI interface give you when compared to Step-/Direction?

In comparison to SPI, Step- Direction can not give you information back from the driver, like error signals or StallGuard information. It lacks information about the actual microstep position, and you do not have free control over the motor current. All this is provided by SPI interface. And you have the free choice to switch to full stepping on-the-fly in order to increase motor power temporarily.

If you would add all this to Step-/Direction, you would end up with a simple interface plus roughly 5 control lines and 10 status lines coming back from the controller. This would make up for a 17 line interface! Of course pure Step-/Direction drivers lack this capability. The SPI interface of the TMC driver family shift back all this status lines to the four simple control lines of SPI. We have chosen SPI, because it is the simplest real bus interface available – it is not more than a shift register with clock and data line. As you will see, the draw back of having shifted the Step-/Direction interface back to within the CPU is not a hard task to solve – look up our simple software examples!

1.3 But what is the additional overhead of SPI?

You have to "play back" a stepping table in your CPU and send it via an SPI interface. As software writers find out quickly, this is a straight-forward and simple task. Nearly all microcontrollers provide an SPI interface, or a synchronous shift register interface.

One example: The 80 cent ATmega8 has an SPI interface with up to 8Mbit/s throughput.

1.4 What is the software load?

For a CPU with SPI interface, you need a table look up, plus two writes to the SPI interface. You can either do this for every (micro)step, or with a fixed rate, which should be some factor higher than your maximum full step rate, e.g. some kHz for a typical application.

Example: For the ATmega8, the routine to look up the table and send out two bytes to the driver needs roughly 5 to 10 microseconds, e.g. when running at 2kHz it consumes 1 to 2 percent of the processing power. This update rate is high enough for most simple positioning applications.

1.5 What is the real challenge when driving a stepper motor?

The real challenge is to generate accurate acceleration and deceleration ramps, in order to stop the motor at the target position. Quick and precise reaction to the reference and stop switches (if used) is essential. Motion ramp generation is a hard real time task! – Unless you want to go at motor start/stop velocity of the motor. Then everything you need is a slow few hundred hertz interrupt to do each one step.

If you want precise and quick motor movement, look for the TMC428 motion controller IC. It comes in a tiny SSOP16 package. If you need more than one motor - it controls up to three stepper motors!

The software load, precise motion control makes, often will be the main task for the processor.

1.6 Can I realize SPI via software?

Yes, you need to take the data word to send to the driver, shift out a bit, set and clear the clock line, shift in the send back bit. Do this 12 times for a TMC249A. It will require around a microsecond per bit on a low cost CPU like the ATmega8 or an up-to-date 8051 family CPU (which in fact have a hardware interface).