Application Note

Reduce Wires for Your Next Multi-Motor BLDC Design With tSPI Protocol

TEXAS INSTRUMENTS

Evan Su

Brushless DC

ABSTRACT

Multi-motor systems require more wiring connections than a single-motor system, resulting in added bulk and complexity. Leveraging the tSPI protocol developed by TI, designers can consolidate these traditionally added connections into a common set of 4 wires between the MCU and associated motors. This implementation produces a more elegant system that can bring cost savings and simplicity.

Table of Contents

List of Figures

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

Multi-motor control is a common design scenario in end applications such as drones and gimbals. For example, a basic 3-axis stabilization system such as drone requires independent and precise control of three BLDC motor drivers to generate the correct motions. In a traditional approach, a single motor driver requires 3 or 6 unique wires for the MCU to apply PWM control signals, and the total wire count in the system scales linearly with the number of motors in the system. This often leads to increased system size and complexity of the design. TI's proprietary tSPI interface is a powerful protocol designed to reduce the wires required of such multi-motor control applications. Using tSPI-capable devices such as the DRV8311, multiple motors can be controlled by a common set of 4 wires from the MCU with the same physical layout as the ubiquitous SPI interface. The savings can be immense; in the aforementioned 3-axis system, the number of wires required between MCU and motor driver can be reduced from 18 to only 4. This document will provide an overview of tSPI protocol and its applications.

2 SPI Connections in a Multi-Motor System

SPI (Serial Peripheral Interface) is a widespread serial communication interface used in many applications, including embedded systems and motor control. In an SPI primary-secondary system, the primary device must provide four wires to a secondary device: a clock signal (CLK), a primary output/secondary input (MSDO), and a primary input/secondary output (MSDI), and a chip select (SCS). A very straightforward implementation of a system with independent secondaries could provide common clock, input, and output signals from the primary device to each secondary device and then assign each secondary device a unique chip select wire. As shown in Figure 2-1, the requirement of a unique serial wire for every secondary device will increase the size and complexity of the system design.

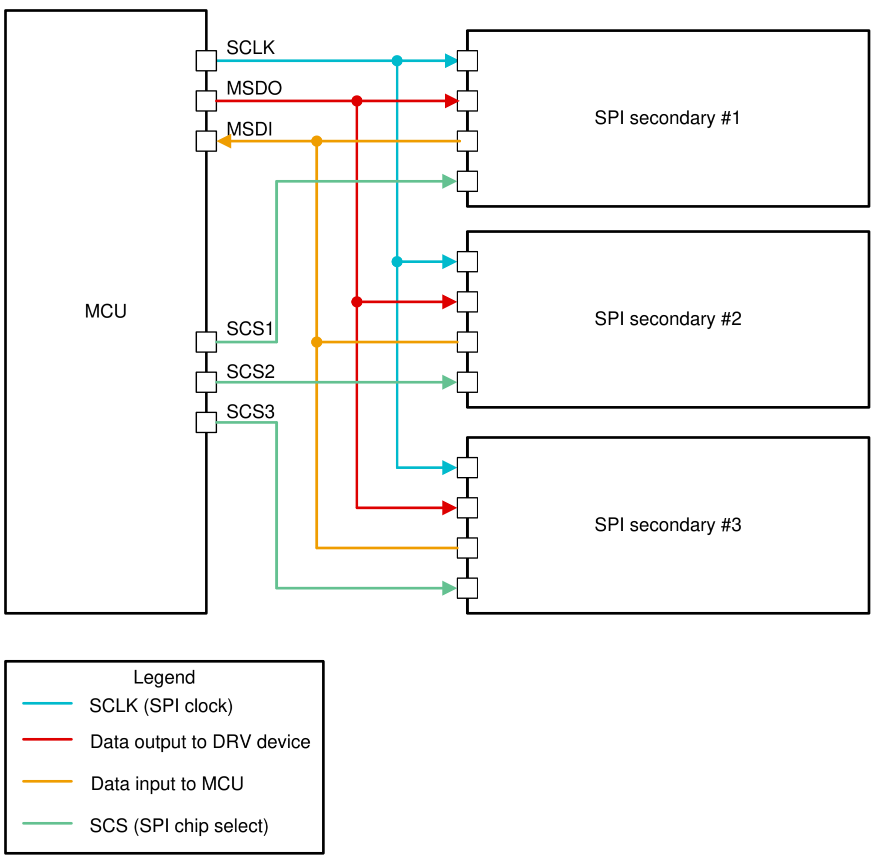

Figure 2-1. Example of Independent Secondary SPI System

These signals do not necessarily need to be unique for each secondary device if more creative architectures are pursued. A common approach is to use daisy-chaining, where secondary devices are essentially pipelined together. As shown in Figure 2-2, in this configuration, the secondary devices relay commands originating from the primary device through the daisy chain in repeated steps.

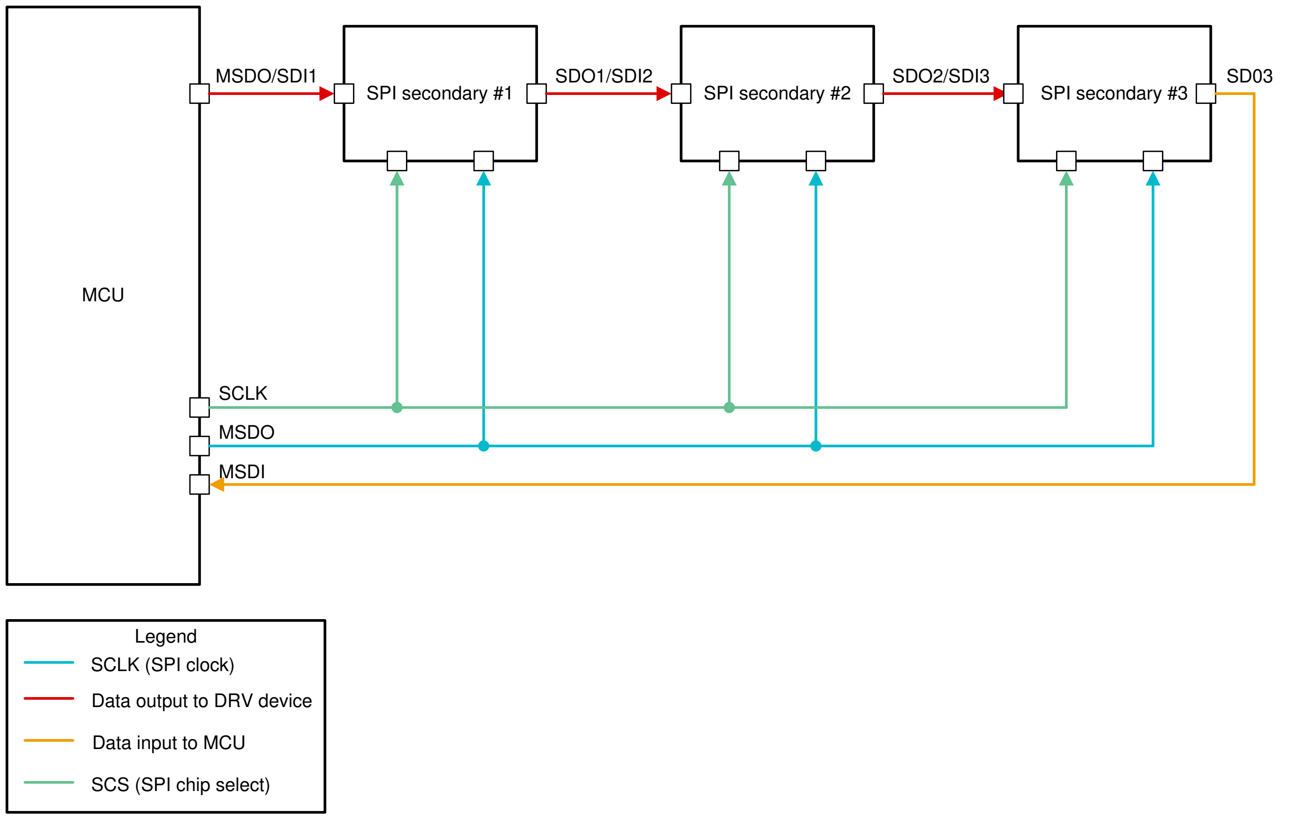

Figure 2-2. Example of Daisy-Chained SPI System

Depending up on number of secondary devices in a daisy chain architecture, the primary device can pass the data over several cycles such that the desired commands propagate to the appropriate secondary device. A single common SCS signal from the primary device allows the secondary devices to execute instructions simultaneously. With this setup, no unique wires are required for any secondary device and the entire system only needs four wires for SPI communication. More information on the details of daisy-chain implementations can be found in the *Daisy Chain Implementation for Serial Peripheral Interface* application report.

Daisy-chaining can successfully reduce the wiring complexity of an SPI implementation by allowing all secondary devices to use only common serial wires. However, daisy-chaining has significant drawbacks in multi-motor control applications, where the MCU is the primary device and motor driver PWM pins are involved for motor control. All devices in a daisy chain must be active at all times to maintain the flow of instructions, so a single inactive device will prevent the system from working properly. It is also not possible to access the daisy-chained secondary devices in a random order. Since these behaviors are undesirable in fast and robust multi-motor control applications, unique PWM wires are still required to be connected directly to the MCU, relegating the function of the SPI interface to only configuration and diagnostics purposes and not addressing the wiring complexity problem discussed in the introduction. The tSPI interface does not suffer from these issues and allows for efficient PWM control completely over a serial interface while maintaining the wire commonality, which will be discussed next.

3 tSPI Interface and Applications

The tSPI interface uses the same physical protocol as SPI. That means it has four wires for clock, input, output, and chip select. This is easily compatible with MCUs. The key difference is that the tSPI protocol allows random access to secondary devices from the primary device via common wires. Up to 15 tSPI devices can be managed through a common chip select. tSPI devices can be used in conjunction with other independent SPI devices connected to the MCU as secondary devices, making it relatively simple to add to existing systems from a hardware perspective as shown in Figure 3-1.

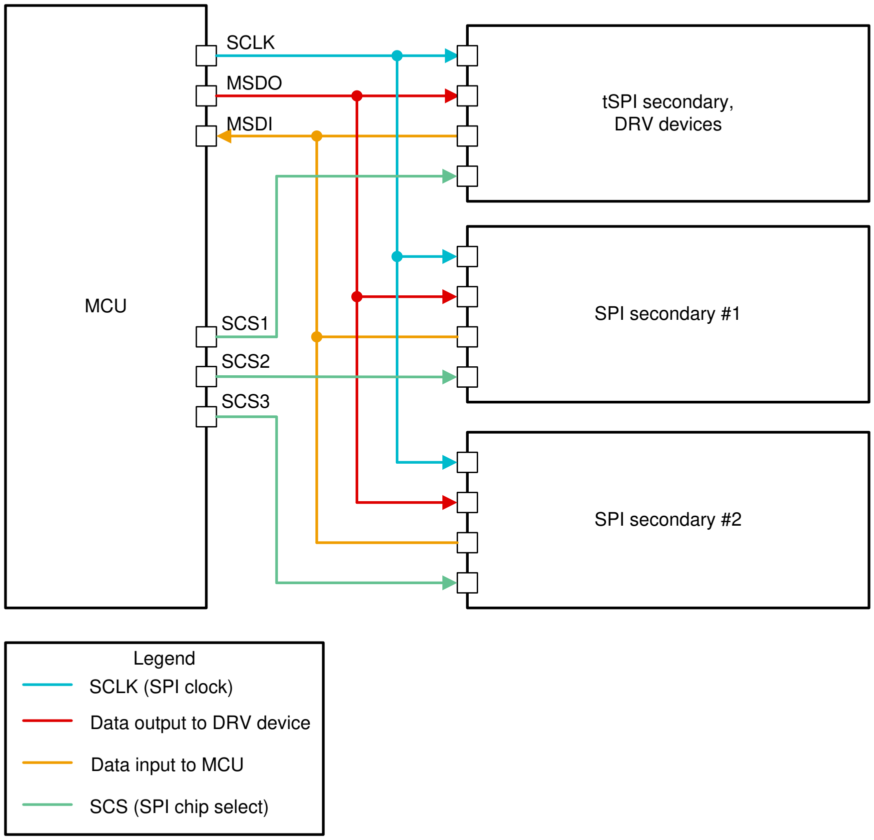

Figure 3-1. Example of tSPI Integration into Independent Secondary SPI System

There are numerous benefits to tSPI protocol on an interface level. For example:

- Random secondary access is permitted, so read and write operations can be performed in any order between secondary devices, which is important for multi motor control.
- There is a provision for a general call address to command all secondary devices managed by the tSPI interface simultaneously, which can be helpful for functions such as starting multiple motors together.
- There is no requirement for all tSPI secondary devices to be active at all times. The interface can successfully perform transactions with any active secondary device regardless of the status of the other devices

The only disadvantage of tSPI is the additional 8 bits of header overhead within a data frame compared to a typical independent SPI implementation, at 32 bits versus 24 bits. More details about the tSPI interface can be found in the data sheet for the DRV8311.

Having discussed the operation of tSPI and the workings of the interface, we can now discuss its benefits in multi-motor control applications. As discussed earlier, a typical hardware setup for motor control applications involves connecting an MCU to multiple PWM pins, which can then be manipulated to control the speed and direction of the motor. For a multi-motor control application, it is possible to use one tSPI device such as a DRV8311 to manage 3 or 6 PWM pins corresponding to a single motor. Multiple such tSPI devices can then be connected to the MCU for a comprehensive and convenient multi-motor application, as shown in Figure 3-2.

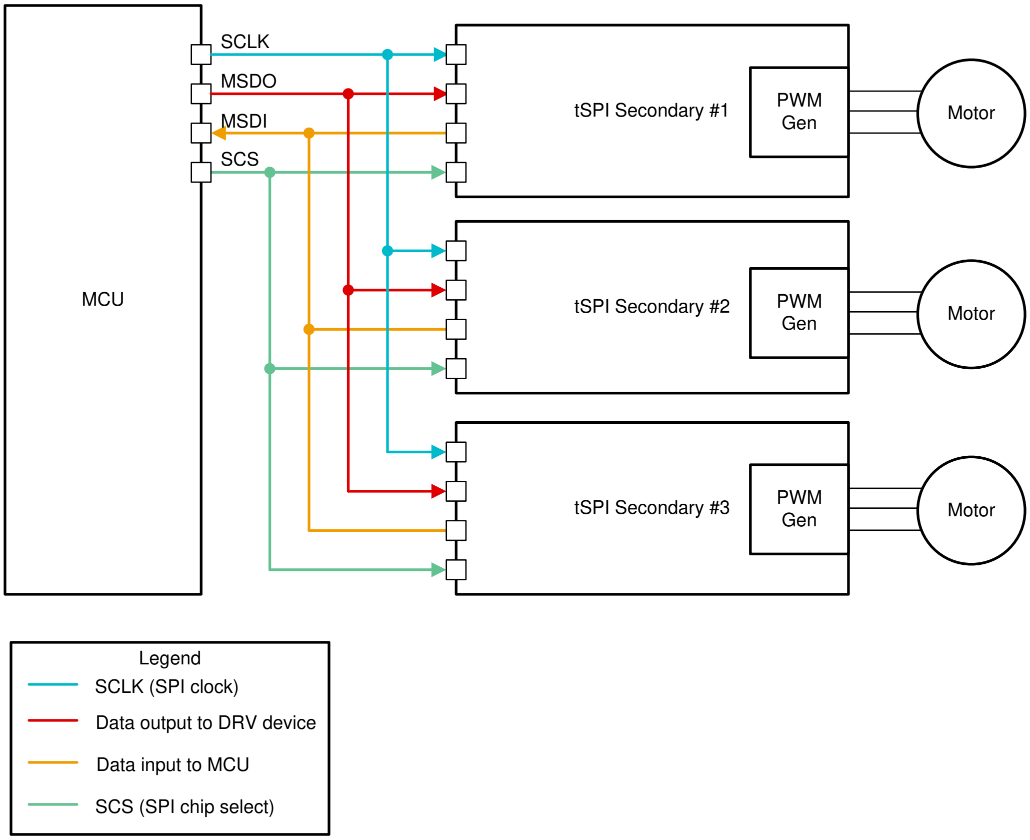

Figure 3-2. Example of tSPI Devices in a Multi-Motor Control Application

In this example, the three motors can be controlled by a common chip select and the MCU only needs 4 wires total. A given motor can be manipulated by sending in write commands to the corresponding tSPI device that is managing its motor driver, which will then lead to appropriate changes in the PWM pins. In general, the formula for the reduction in wires compared to the standard approach is (N * PWMx – 4), where N is the number of motors in the design and PWMx is the number of PWM pins required by each motor, previously discussed to be either 3 or 6. The wire savings can be very significant when multiple motors are present, helping to keep the overall design simple, compact, and easy to manufacture. The reduction in required MCU wiring also allows for more MCU pins to be allocated for other purposes, or perhaps the selection of a smaller MCU depending on the project.

4 Conclusion

tSPI protocol offers a simple and versatile method of controlling multiple motor drivers that can significantly simplify the necessary wiring for multi-motor control applications by allowing PWM duty cycle and current sense amplifier information to be interfaced over a traditional serial interface. tSPI's simple and straightforward SPI physical interface allows it to be easily integrated into existing designs involving MCUs. From a software and communications perspective, the interface is convenient and more versatile compared to traditional approaches like daisy-chained SPI. Products with the tSPI protocol, such as the DRV8311, are therefore highly valuable for any end applications where design complexity and size are concerns. End applications such as drones and gimbals will benefit greatly from the tSPI protocol. With further exposure in learning how to leverage the features of tSPI, users can design their systems to accompish additional performance improvements such as fast multiwrites for multimotor systems with synchronous PWM duty cycle updates. This particular example being mentioned is achieved with the tSPI feature of automatic write pointer address incrementing, which is discussed in the application note Implementing multi-motor systems using tSPI. Check out TI's tSPI protocol enabled device DRV8311 by visiting ti.com/product/DRV8311.