

CIRCUIT CELLAR®

THE MAGAZINE FOR COMPUTER APPLICATIONS

FEATURE ARTICLE

Art Eck

Serial Interface for Embedded Design

Choosing the right interface can greatly reduce the cost of a project, but the right one depends on what features would benefit your project. Art helps you negotiate the pros and cons of SPI, Mico-wire, and I²C. He'll help you decide which serial interface is right for your project.



any mixed-signal semiconductor devices offer serial interfaces that significantly reduce the pin count and cost of the mixed-signal device and the microcontroller in the system. Although the fundamental benefits of using interfaces are obvious, determining the appropriate serial interface for the design is an important consideration. A comparison of the three major serial protocols helps determine

which one may be most desirable for a given design.

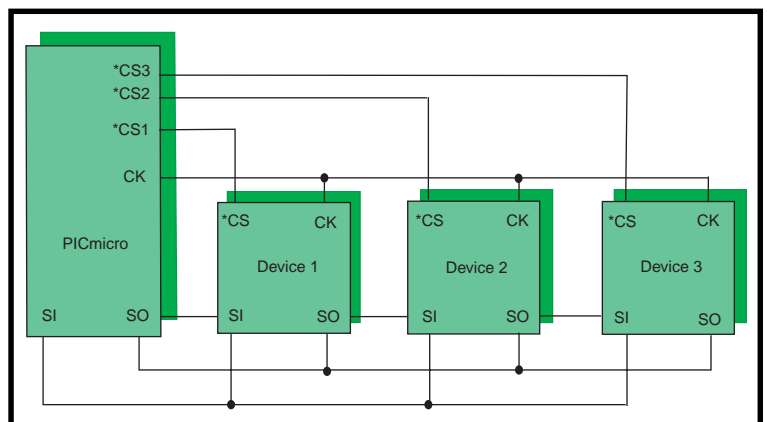
The three major serial protocols available for DACs, ADCs, and digital potentiometers are SPI, Microwire, and I²C. Both Microwire and SPI require three I/O pins for communication and a fourth pin for chip select. SPI has the advantage of being more broadly offered, providing typically higher clock speeds, and having the most compact and easiest code to write.

In terms of development time and demands on system memory, SPI protocol is a clear winner. Not only is the code easier to write, but usually less code is required for a given task. Shorter, easier-to-write code adds up to shorter development time. With a more compact code, a smaller requirement is placed on memory, which can help to reduce the overall system cost. Also, many microcontrollers offer an onboard SPI port that can further reduce development time.

In addition, the SPI serial protocol offers a faster clock speed. With less code to execute and a faster clock, SPI solutions are capable of shorter cycle times when performing a desired task. Where system speed is important, the SPI serial protocol again is the front runner.

The SPI serial protocol suffers a significant setback when compared to

Figure 1—This arrangement can place a significant demand on the number of pins on the microcontroller or even require a separate control device.



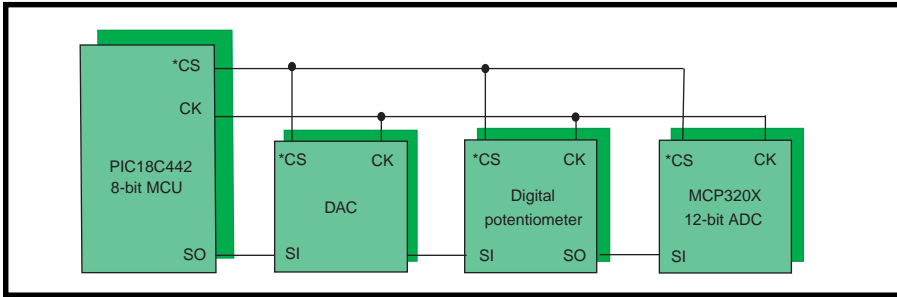


Figure 2—Although the daisy chain configuration eliminates the need for separate selection of each device and the demand that placed on system resources, this configuration also lowers bus speed.

the I²C serial protocol in terms of the chip-select pin. In a traditional SPI bus structure, every SPI device must be selected separately when addressed. This means that the system must have a separate select line to select each device (see Figure 1). This can place a significant demand on the number of pins on the microcontroller or even require a separate control device.

I²C serial protocol allows devices to be selected by a command containing the device address. This address can either be stored in nonvolatile memory in the device or hardwired by address pins on the device. Either way, the I²C protocol offers a significant reduction in system resources to select the device over a traditional SPI configuration. The software command completely eliminates the need for a hardware chip-select pin in order to have proper bus operation.

To minimize the system requirements in an application using multiple SPI devices, many manufacturers offer the ability to daisy chain. It is required that the SPI device have a data-out line (SO), although it does not transmit data back to the microcontroller.

As can be seen in Figure 2, only one chip-select line is needed in the daisy chain configuration. All devices are selected at the same time and each then functions as a shift register, and the data is shifted down the line until it arrives at the proper device. The last device in this chain does not need to have an SO line because there are no more devices to shift data out to.

There is a price to pay for using this configuration. Although the daisy chain configuration eliminates the need for separate selection of each

device and the demand that places on system resources, this configuration also lowers bus speed. A typical SPI bus speed of 10 MHz is lowered to 5.8 MHz as a result of propagation delay of data coming out of the SO pin of the previous device. (This is still significantly faster than most I²C buses.)

To reduce the number of commands needed to send data down the daisy chain, all the devices are loaded with zeros upon assertion of the chip-select pins. In this way, the command shifted out is the NOP command. So, as soon as the command from the microcontroller reaches the proper device, transmission can stop. There is no need to load NOP into all the devices in the chain that do not receive the command to prevent erroneous command execution. With NOP load during chip selection, all devices are preloaded saving valuable command shifting time.

Because the SPI interface is licensed at no cost, it is likely to continue to grow in popularity. A designer new to serial protocols will likely gain the most use from learning SPI, in that it yields a large diversity of products now and is likely to con-

tinue to grow in the years to come. Of all the serial protocols available today, knowledge of SPI is also likely to yield the longest use for the broadest selection of applications.

With all of the benefits of the SPI serial protocol, it seems a designer could just abandon other protocols and use just this one. Although this may be true when compared to Microwire, which seems to gain its acceptance more from legacy than from offering benefits beyond that of SPI, I²C is able to stand on more than the merits of past use.

TRYING I²C

The I²C serial protocol requires only two I/O pins, the least of the three major serial protocols (see Figure 3). This means that not only does the mixed-signal device using I²C serial protocol cost less, but the microcontroller it interfaces to can have a lower pin count. This, coupled with reduced board-space requirements, can offer a lower overall system cost. The system-cost reduction must be weighed against system-cost savings, because of the more compact code that SPI offers.

Noise can be an issue with any mixed-signal design. Although the best solution is to keep serial buses far from noise, this is not always possible. Given that I²C protocol devices are level triggered, noise immunity is likely to be superior when compared to edge-triggered protocols like SPI and Microwire. Whenever noise is an issue, devices with Schmitt triggers should be considered.

Because of the benefits of reduced pin count and the popularity of the

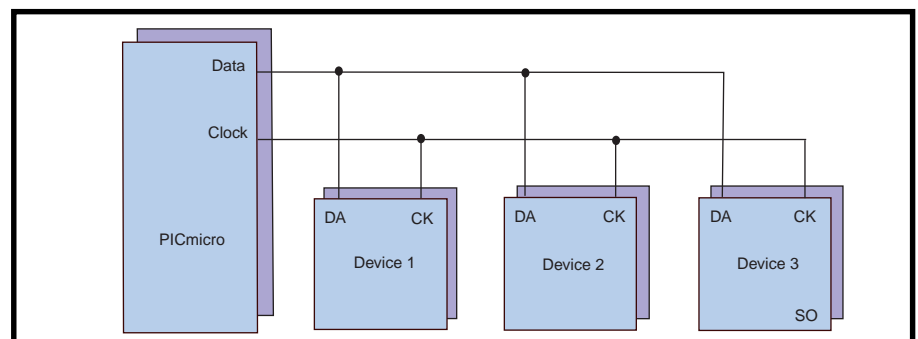


Figure 3—Given that I²C protocol devices are level triggered, noise immunity is likely to be superior when compared to edge-triggered protocols like SPI and Microwire.

I²C serial protocol, it is likely to continue in the industry for a long time. Like SPI, many additional products will also be developed for this protocol. Although the I²C serial protocol may not enjoy as broad an offering of products in mixed-signal devices, it will continue to have a significant presence.

The I²C serial protocol tends to lack speed. Until recently, I²C serial protocol was not defined for operation above 400 kHz. Applications that require significantly fast data transfer may find the I²C serial protocol device not capable of meeting the requirements.

Besides these three major protocols, there are many serial protocols employed by suppliers that are proprietary to their product. Some of these approaches seem to have benefits for a given application but may not have value in the broader use. It is unlikely any of these approaches will gain the acceptance that SPI, Microwire, and I²C serial protocols have, making it difficult to justify investing the time to learn them.

Proprietary protocols lock the application into a particular supplier. The lack of competition is likely to mean higher cost over the life of the project. Even with reasonable prices today, later price reductions are not as likely for devices using standard protocols. Given that support of a proprietary protocol is less broad based, the time used to learn it is less likely to be a useful investment. Therefore, it is recommended that proprietary protocols be avoided whenever possible.

Each of the three major serial protocols offer different value points. Microwire offers a legacy as the oldest of the three. SPI offers greatest execution speed, most compact code, and a broad offering of products. I²C requires only two system I/O pins—half that of SPI and Microwire.

It's up to the designer to determine the set of features that most benefit the application. Whatever the selection, the overall design can enjoy a cost reduction through the use of an appropriate serial protocol. ■

Art Eck is an analog product marketing manager for Microchip Technology, Inc. Before joining Microchip, he previously worked as a product engineering manager of high-power RF transistors for Motorola. He's a father of four and a Scoutmaster in his spare time.

SOURCES

I²C

Microchip Technology, Inc.
(888) 628-6247
(480) 786-7200
Fax: (480) 899-9210
www.microchip.com

Philips Semiconductors
(408) 991-5207
Fax: (408) 991-3773
www.semiconductors.philips.com

SPI

Motorola
(512) 328-2268
Fax: (512) 891-4465
www.mot-sps.com/sps/general/chips-nav.html

Microwire

National Semiconductor
(408) 721-5000
Fax: (408) 739-9803
www.national.com

Circuit Cellar, the Magazine for Computer Applications. Reprinted by permission. For subscription information, call (860) 875-2199, subscribe@circuitcellar.com or www.circuitcellar.com/subscribe.htm.