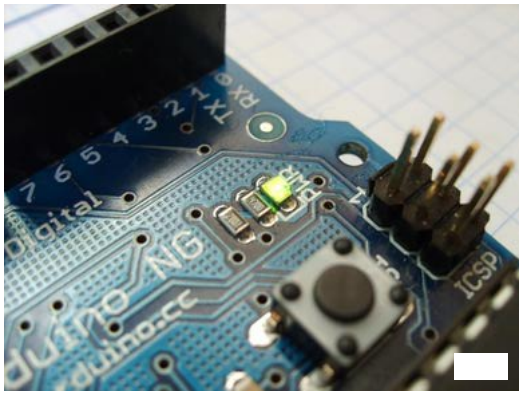
By Matthew Burris
Components Expert



Dusty Dingo/Wikimedia Common/Public Domain

Choosing between I2C and SPI, the two main serial communication options, can be quite a challenge and have a significant impact on the design of a project, especially if the wrong communication protocol is used. Both SPI and I2C bring their own advantages and limitations as communication protocols which make them each suited for specific applications.

SPI

SPI, or Serial to Peripheral Interface, is a very low power, four wire serial communication interface designed for IC controllers and peripherals to communicate with each other.

The SPI bus is a full-dulex bus, which allows communication to flow to and from the master device simultaneously at rates of up to 10Mbps. The high speed operation of SPI generally limits it from being used to communicate between components on separate PCBs due to the increase in capacitance that longer distance communication adds to the signal lines.

PCB capacitance can also limit the length of SPI communication lines.

While SPI is an established protocol, it is not an official standard which leads to several variants and SPI customizations which can lead to compatibility issues. SPI

implementations should always be checked between master controllers and slave peripherals to ensure that the combination will not have any unexpected communication problems that will impact development of a product.

I2C

I2C is an official standard serial communication protocol that only requires two signal lines that was designed for communication between chips on a PCB. I2C was originally designed for 100kbps communication but faster data transmission modes have been developed over the years to achieve speeds of up to 3.4Mbps.

The I2C protocol has been established as an official standard, which provides for good compatibility among I2C implementations and good backward compatibility.

Selecting Between I2C and SPI

Selecting between I2c and SPI, the two main serial communication protocols, requires a good understanding of the advantages and limitations of I2C, SPI, and your application.

Each communication protocol will have distinct advantages which will tend to distinguish itself as it applies to your application. The key distinctions between I2C and SPI are:

- I2C requires only two wires, while SPI requires three or four
- SPI supports higher speed full-dulex communication while I2C is slower
- I2C draws more power than SPI
- I2C supports multiple devices on the same bus without additional select signal lines through in-communication device addressing while SPI requires additional signal lines to manage multiple devices on the same bus
- I2C ensures that data sent is received by the slave device while SPI does not verify that data is received correctly
- I2C can be locked up by one device that fails to release the communication bus
- SPI cannot transmit off the PCB while I2C can, albeit at low data transmission speeds
- I2C is cheaper to implement than the SPI communication protocol
- SPI only supports one master device on the bus while I2C supports multiple master devices
- I2C is less susceptible to noise than SPI
- SPI can only travel short distances and rarely off of the PCB while I2C can transmit data over much greater distances, although at low data rates
- The lack of a formal standard has resulted in several variations of the SPI protocol, variations which have been largely avoided with the I2C protocol

These distinctions between SPI and I2C should make selecting the best communication option for your application easier. Both SPI and I2C are good communication options, but each has a few distinct advantage and preferred applications. Overall, SPI is better for high speed and low power applications while I2C is better for suited to communication with a large number of peripherals and dynamic changing of the master device role among the peripherals on the I2C bus. Both SPI and I2C are robust, stable communication protocols for embedded applications that are well suited for the embedded world.

Related Articles

- **Overview of I2C**
- **Level Shifting Basics**
- **Getting Started with PCB Design**
- **FPGA Accessory for the Raspberry Pi**
- **4 PCB Troubleshooting Techniques**
- **Three Main Failure Modes of Electronics**