

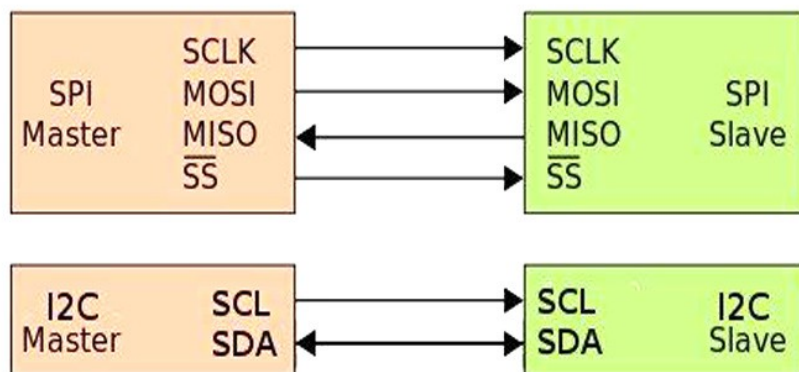


[Home](#) > [Electronics](#) > I2C or SPI Serial Communication, which one to go with?

I2C or SPI Serial Communication, which one to go with?

 Maurizio  September 30, 2015 
Electronics  No Comments



I2C or SPI communication, this is the question! Serial Peripheral Interface Bus (SPI) was established by Motorola and Inter Integrated Circuit (I2C) was invented by Philips. So the main protocols to consider are: SPI protocol and I2C protocol.

POPULAR POSTS



[How to create teletext pages](#)

February 23, 2016 • 3 Comments



[Serial Port Communication in C#](#)

February 8, 2016 • 2 Comments



[The Decibel Explained and](#)

1. INTRODUCTION

Before adding serial communication to your design, let's say adding a serial eeprom, you should have an understanding of the different types of serial communication that you can use. **There are two main protocols to consider, SPI protocol and I2C protocol.**

Both systems have their unique advantages and disadvantages which make them more or less suitable for a given application.

2. SPI COMMUNICATION

SPI supports full duplex communication with higher throughput than I2C. It is not limited to 8-bit words so you can send any message sizes with arbitrary content and purpose. The SPI interface does not require pull-up resistors, which translates to lower power consumption. However, I2C is simpler by having fewer lines which means fewer pins are required to interface to an IC. **When communicating with more than one slave, I2C has the advantage of in-band addressing as opposed to have a chip select line for each slave.** I2C also supports slave acknowledgment which means that you can be absolutely sure that you're actually communicating with something. With SPI, a master can be sending data to nothing at all and have no way to know that. **In general SPI is better suited for applications that deal with longer data streams** and not just words like address locations. Mostly longer data streams exist in applications where you're working with a digital signal processor or analog-to digital converter. For example, SPI would be perfect for playing back some audio stored in an eeprom and played through a digital to analog converter DAC. Moreover, since SPI can support

Helpful Tips

March 7, 2016 • No

Comment



Back to
Vacuum
Tubes:

Reinventing transistors

March 8, 2016 • No

Comment



Transistor

Characteristics

March 11, 2016 • No

Comment



LVDS
Interface
in

Automotive Systems

March 14, 2016 • No

Comment



NFC,
between
business
and

humanitarian operations

March 15, 2016 • No

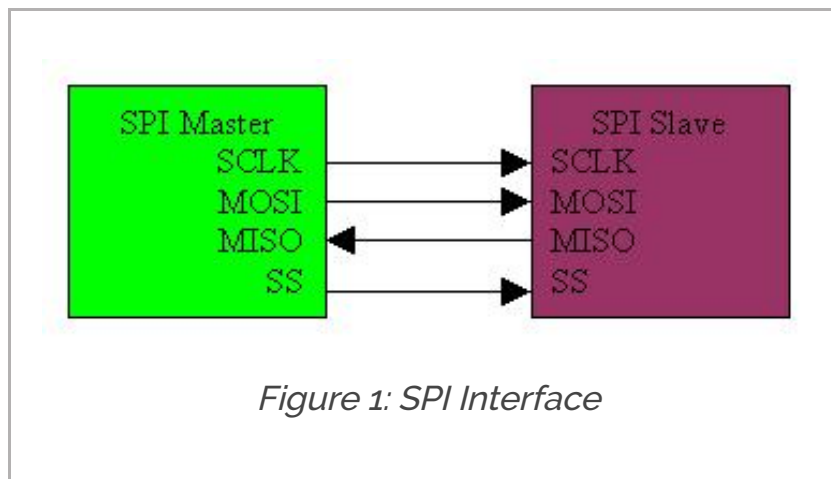
Comment



Using
Electronic
Motor
Control to

Achieve 1 SEER in HVAC Systems

significantly higher data rates comparing to I2C, mostly due to its duplex capability, it is far more suited for higher speed applications reaching tens of megahertz. And since there is no device addressing involved in SPI the protocol is a lot harder to use in multiple slave systems. **This means when dealing with more than one node, generally I2C is the way to go.**



As shown in figure 1, SPI has 4 lines. The SCLK line is the clock line, the clock is generated by the master and drives the communication in both directions, and this line is an input to all slaves. **The MOSI line is the master data output, slave data input, and it carries data from the master to the slave.** The MISO line is the master data input, slave data output, and it carries data from the slave to the master. Finally the SS or sometimes known as the CS line is the slave select or chip select line, it is toggled to select a slave to be communicated with. Usually the transfer sequence consist of driving the SS line low with a general I/O pin, sending X number of clock signals with the proper polarity and phase, driving the SS line high to end the communication. **As the clock signals are generated, data is transferred in both directions, therefore in a “transmit only” system the received bytes have to be discarded and in a “receive only” system a dummy byte has to**

March 18, 2016 • No Comment



Cost-effective WPAN solutions for sensing and control

March 21, 2016 • No Comment



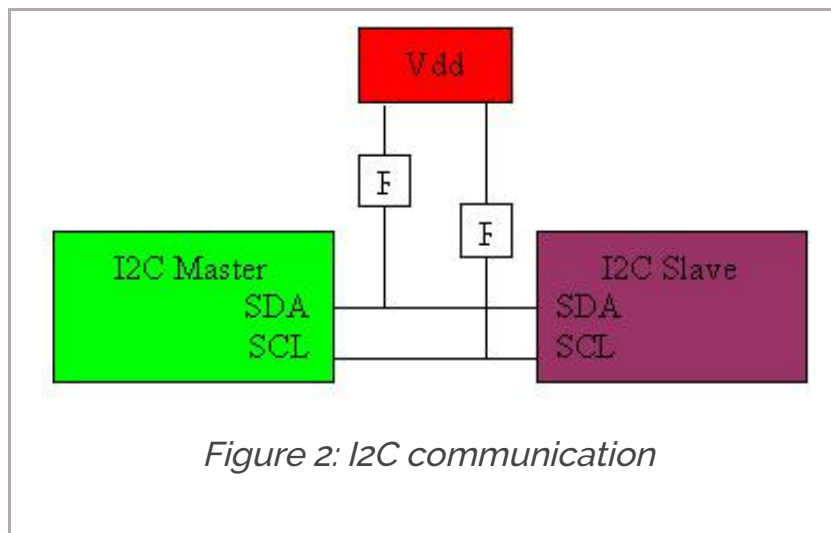
Remove brushless DC motor commutation spikes

January 18, 2016 • No Comment

be transmitted. Care has to be taken not to toggle the SS during the communication since this will introduce errors. The clock polarity and clock phase control's on which edge of the clock signal the data is received and sent. This has to be set up to match between the master and the slave. There is also a 3-Wire version of SPI however it only supports half duplex communication. This set up uses a SISO line, this single bidirectional data line carries data in and out of the slave. This mode has a tendency of not being supported by microcontrollers however it is easily implemented with bit banging in software.

3. I2C COMMUNICATION

I2C (Figure 2) consists of two bidirectional lines that are pulled up to Vdd. The SDA line is the serial data line and SCL is the serial clock line.



The standard I2C modules support a 7-bit slave address and can support up to 112/128 nodes, some extended I2C modules can support 10-bit slave addressing however all the modules are limited to 400pF bus capacitance. Standard speeds are 10kbit/s Low-speed mode and 100kbit/s Standard mode. The extended versions support 400kbit/s Fast mode, 1Mbit/s Fast mode plus, and 3.4Mbit/s

High-speed mode. **The three types of messages that are defined by the I2C protocol** are a single message where the master writes to a slave, a single message where the master reads from a slave, and a combined message where a master issues at least two reads and/or writes to one or more slaves. The sequence of the communication begins with the master sending a start bit followed by the 7 or 10-bit slave address and finally a bit that selects if the operation is a write(0) or a read(1). At this point, if the slave address exists on the bus the slave will send an acknowledgment bit to the master. **The data is then transferred on the SDA line in the direction that was specified by the master.** An acknowledgment bit is sent at the end of each transferred byte by the receiving end of the transmission. The only exception is that when the master is in receive mode and the slave in transmit mode the master will not send an acknowledge bit after the very last bit received. Lastly the communication is stopped with the master sending a stop command. The start and stop commands are simply a transition from high to low (Start) on the SDA line with SCL high or low to high (Stop) on the SDA line with SCL high. Transitions for the data bits are always performed while SCL is low; the high state is only for the start/stop commands.

Hopefully this gives you a **better understanding of these two protocols**, and their differences. It should be a lot simpler to select one over the other for a given application. But life is not always that easy, you still have to consider if other types of communications are not more suitable for your application, you could even be better off not using a serial protocol at all but to go with a parallel type communication.

Tags: [i2c](#), [i2c bus](#), [i2c communication](#), [serial communication](#), [spi interface](#), [spi protocol](#)

About The Author



Maurizio

Maurizio Di Paolo Emilio, PhD in Physics and telecommunications engineer, collaborates with various technical magazines as author and

with the University of L'Aquila in the field of X-ray, high voltage system and optics as software/hardware designer. He is the author of various books (<http://amazon.com/author/maurizio>) and many technical and scientific publications in the field of electronic design, Linux, PCB, data acquisition systems, embedded systems, x-ray imaging and its applications, high voltage pulsed system.

Leave a Reply

You must be [logged in](#) to post a comment.

RECENT POSTS

- Calibrating the eQADC module of the Freescale MCP5510 Microcontroller

RECENT COMMENTS

- Maurizio on Serial Port Communication in C#
- sunilkumar on Serial Port



Family

- CE Mark – Electromagnetic Fields
- Ethernet: IP
- Generating Analog Voltage with Digital Circuit
- Digital TV Standards: DVB-T, ATSC, ISDB-T

Communication in C#

- Stoqn4o_9212 on Howto Measure RTD (Resistance Temperature Detectors) over long distances
- Maurizio on How to create teletext pages
- Emanuele on How to create teletext pages

Electronics Open Source © 2015 - EMCelettronica Srl - P.iva 06433601009 - Some Rights Reserved!