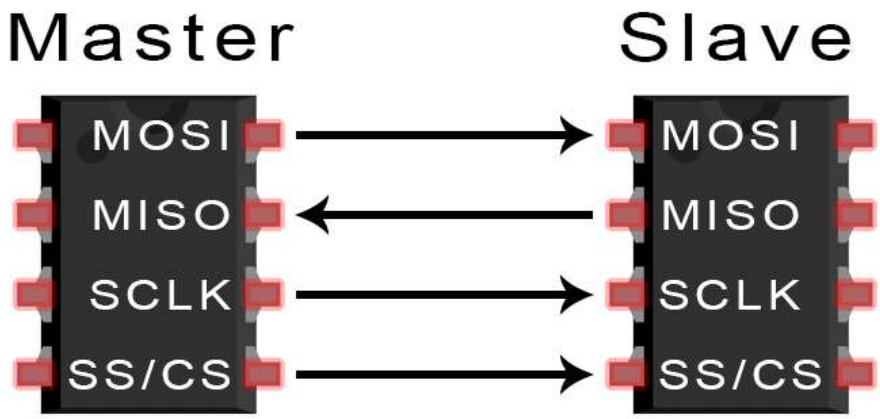
BASICS OF THE SPI COMMUNICATION PROTOCOL

Posted by Scott Campbell | DIY Electronics | 59



When you connect a microcontroller to a sensor, display, or other module, do you ever think about how the two devices talk to each other? What exactly are they saying? How are they able to understand each other?

Communication between electronic devices is like

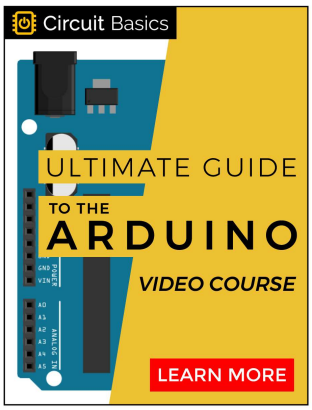
SEARCH ...

SUBSCRIBE

Get new tutorials sent to your inbox!

EMAIL ADDRESS

SUBSCRIBE



discuss the basics of the three most common protocols: Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C), and Universal Asynchronous Receiver/Transmitter (UART) driven communication.

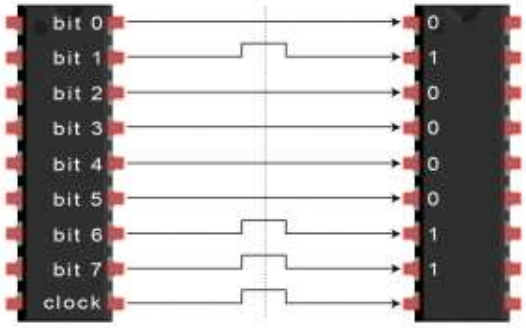
First, we'll begin with some basic concepts about electronic communication, then explain in detail how SPI works. In the next article, we'll discuss UART driven communication, and in the third article, we'll dive into I2C.

SPI, I2C, and UART are quite a bit slower than protocols like USB, ethernet, Bluetooth, and WiFi, but they're a lot more simple and use less hardware and system resources. SPI, I2C, and UART are ideal for communication between microcontrollers and between microcontrollers and sensors where large amounts of high speed data don't need to be transferred.

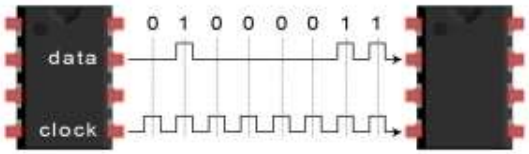
SERIAL VS. PARALLEL COMMUNICATION

Electronic devices talk to each other by sending *bits* of data through wires physically connected between devices. A bit is like a letter in a word, except instead of the 26 letters (in the English alphabet), a bit is binary and can only be a 1 or 0. Bits are transferred from one device to another by quick changes in voltage. In a system operating at 5 V, a 0 bit is communicated as a short pulse of 0 V, and a 1 bit is communicated by a short pulse of 5 V.

The bits of data can be transmitted either in parallel or serial form. In parallel communication, the bits of data are sent all at the same time, each through a separate wire. The following diagram shows the parallel transmission of the letter "C" in binary (01000011):



In serial communication, the bits are sent one by one through a single wire. The following diagram shows the serial transmission of the letter "C" in binary (01000011):

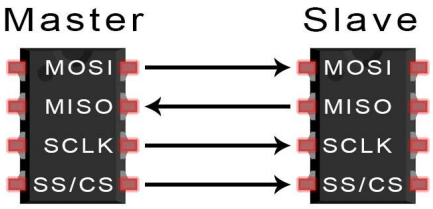


INTRODUCTION TO SPI COMMUNICATION

SPI is a common communication protocol used by many different devices. For example, SD card reader modules, RFID card reader modules, and 2.4 GHz wireless transmitter/receivers all use SPI to communicate with microcontrollers.

One unique benefit of SPI is the fact that data can be transferred without interruption. Any number of bits can be sent or received in a continuous stream. With I2C and UART, data is sent in packets, limited to a specific number of bits. Start and stop conditions define the beginning and end of each packet, so the data is interrupted during transmission.

while the slave (usually a sensor, display, or memory chip) takes instruction from the master. The simplest configuration of SPI is a single master, single slave system, but one master can control more than one slave (more on this below).



MOSI (Master Output/Slave Input) – Line for the master to send data to the slave.

MISO (Master Input/Slave Output) – Line for the slave to send data to the master.

SCLK (Clock) – Line for the clock signal.

SS/CS (Slave Select/Chip Select) – Line for the master to select which slave to send data to.

Wires Used	4
Maximum Speed	Up to 10 Mbps
Synchronous or Asynchronous?	Synchronous
Serial or Parallel?	Serial
Max # of Masters	1
Max # of Slaves	Theoretically unlimited*

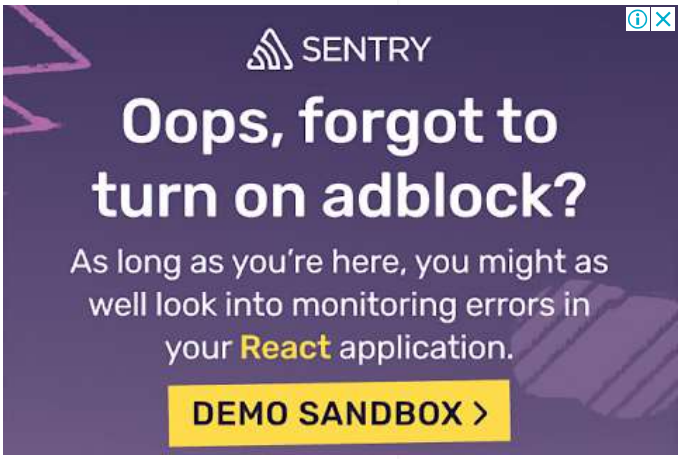
*In practice, the number of slaves is limited by the load capacitance of the system, which reduces the ability of the master to accurately switch between voltage levels.

HOW SPI WORKS

THE CLOCK

The clock signal synchronizes the output of data bits from the master to the sampling of bits by the slave. One bit of data is transferred in each clock cycle, so the speed of data transfer is determined by the frequency of the clock signal. SPI communication is always initiated by the master since the master configures and generates the clock signal.

set to a pre-configured baud rate that dictates the speed and timing of data transmission.



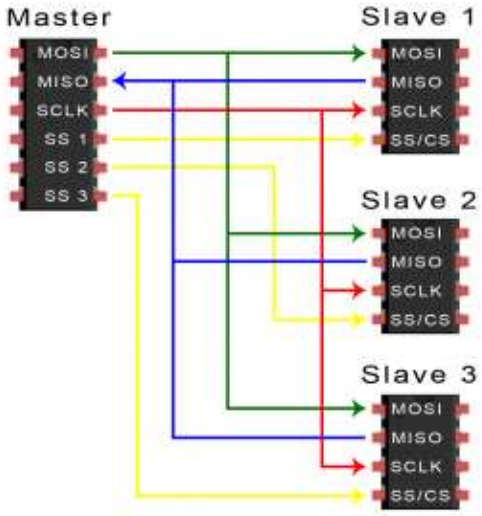
The clock signal in SPI can be modified using the properties of *clock polarity* and *clock phase*. These two properties work together to define when the bits are output and when they are sampled. Clock polarity can be set by the master to allow for bits to be output and sampled on either the rising or falling edge of the clock cycle. Clock phase can be set for output and sampling to occur on either the first edge or second edge of the clock cycle, regardless of whether it is rising or falling.

SLAVE SELECT

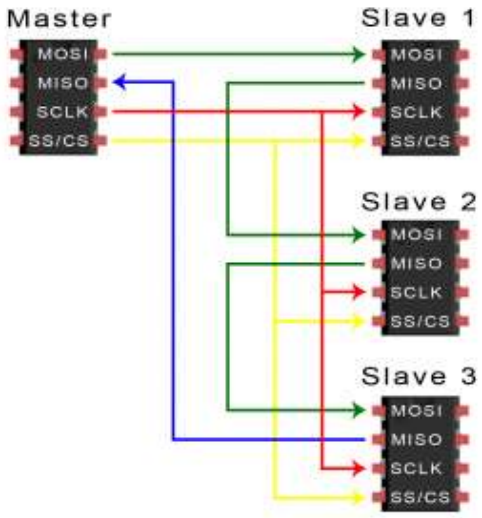
The master can choose which slave it wants to talk to by setting the slave's CS/SS line to a low voltage level. In the idle, non-transmitting state, the slave select line is kept at a high voltage level. Multiple CS/SS pins may be available on the master, which allows for multiple slaves to be wired in parallel. If only one CS/SS pin is present, multiple slaves can be wired to the master by daisy-chaining.

MULTIPLE SLAVES

SPI can be set up to operate with a single master and a single slave, and it can be set up with multiple slaves controlled by a single master. There are two ways to connect multiple slaves to the master. If the master has multiple slave select pins, the slaves can be wired in parallel like this:



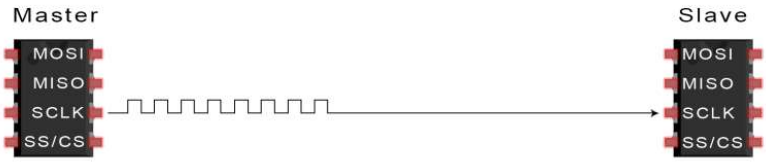
If only one slave select pin is available, the slaves can be daisy-chained like this:



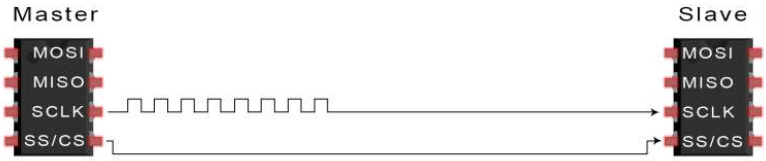
MOSI AND MISO

The master sends data to the slave bit by bit, in serial through the MOSI line. The slave receives the data sent from the master at the MOSI pin. Data sent from the master to the slave is usually sent with the most significant bit first.

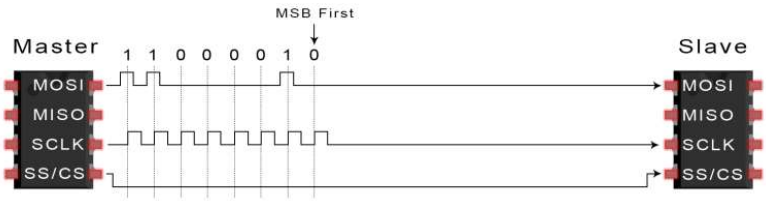
The slave can also send data back to the master through the MISO line in serial. The data sent from the slave back to the master is usually sent with the least significant bit first.



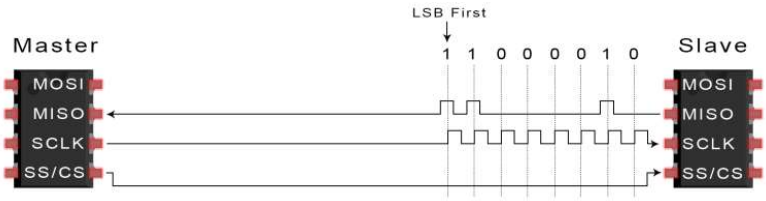
2. The master switches the SS/CS pin to a low voltage state, which activates the slave:



3. The master sends the data one bit at a time to the slave along the MOSI line. The slave reads the bits as they are received:



4. If a response is needed, the slave returns data one bit at a time to the master along the MISO line. The master reads the bits as they are received:



ADVANTAGES AND DISADVANTAGES OF SPI

There are some advantages and disadvantages to using SPI, and if given the choice between different communication protocols, you should know when to use SPI according to the requirements of your project:

ADVANTAGES

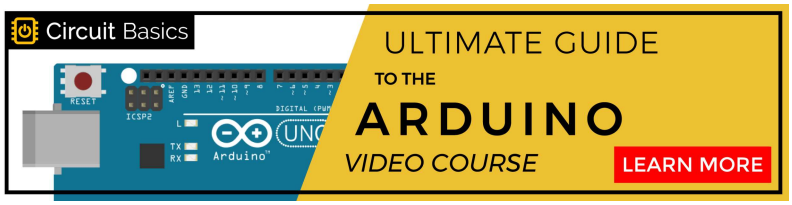
- Separate MISO and MOSI lines, so data can be sent and received at the same time

DISADVANTAGES

- Uses four wires (I2C and UARTs use two)
- No acknowledgement that the data has been successfully received (I2C has this)
- No form of error checking like the parity bit in UART
- Only allows for a single master

Hopefully this article has given you a better understanding of SPI. Continue on to part two of this series to learn about UART driven communication, or to part three where we discuss the I2C protocol.

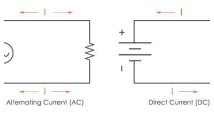
If you have any questions, feel free to ask it in the comment section, we're here to help. And be sure to subscribe, we send out an email each time we publish new tutorials!



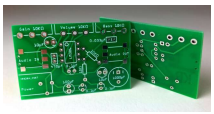
SHARE:

RELATED POSTS

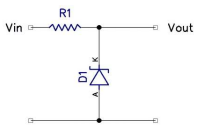
How Variable Resistors Work



What is Direct Current?



How to Design a PCB Layout



How to Make Voltage Regulator Circuits

59 COMMENTS

Greg Zater on February 13, 2016 at 11:20 pm
Great post, thanks (:

REPLY

Nhan on February 14, 2016 at 7:02 am
Simple enough:)

REPLY

SpaceMonkey on February 14, 2016 at 10:59 am
Awesome and very well explained ! Can't wait for part 2 and 3 :)

REPLY

zamberson on February 14, 2016 at 12:18 pm
Excellent! Thanks!

REPLY

Black Hack on February 14, 2016 at 7:31 pm
Very nice and erudite explanation of what could be a confusing technical issue. I especially like the fact you brought up an answer many of us would have with this statement, "With I2C and UART, data is sent in packets", making clear there is no header data along with encapsulation. Great Job.

Black Hack Wireless

REPLY

Wannachat Surasiang on February 15, 2016 at 6:51 pm
Thank!!

REPLY

slave like master do to be synchronize the returning data?

REPLY

Dnyaneshwar Sahane on March 23, 2019 at 11:27 am

Yono,
If you r talking about SPI then there is separate line called MISO. On this line slave will send the data. What and when do should send by slave which is mentioned in tje respective device datasheet.... All the cmunication details mentioned in the datasheet of slave device accordingly master need to send the command to slave. Clock is required for every bit which will be generated by master only.

REPLY

yonoodle on February 23, 2016 at 2:37 am

or does the clock never stop even there's no data sending/receiving?

REPLY

Vinod negali on March 18, 2016 at 4:03 pm

Excellent explanation. Easy to understand specially for beginners.

REPLY

Pramod Chavan on June 7, 2016 at 5:09 pm

all posts(for UART,SPI & I2C) are helpful

REPLY

Purushotham on November 17, 2016 at 2:29 pm

Very nicely explained in simple manner.
Thank you.

REPLY

Can somebody please explain how communication happens in daisy chained mode.

REPLY

SyedIbrahim on December 12, 2016 at 4:05 am

Good post, in a simple language and terms, where everyone can understand easily. But, i also want to know about the modes of operation in SPI. Have you explained that in a different post? Or can you add it?

REPLY

Rohit on February 12, 2017 at 10:25 am

good post
heuristic process of learning in simple way.

REPLY

Sarah on March 16, 2017 at 3:58 pm

Thank you so much ^_^

REPLY

mahesh on July 6, 2017 at 5:18 pm

keep going...simple explaining

REPLY

Sivaprakash on October 5, 2017 at 3:52 pm

Thank u so much for ur simple explanations.

REPLY

raghu on October 10, 2017 at 4:49 pm

how will the slave sense that the master is sending bit 1 or 0?

REPLY

Onnie on October 14, 2017 at 2:42 pm

Good Article.. Simple :)

This article is written as an introductory part of communication protocols but there is a lot of comparison with other protocols before introducing to them. Otherwise one of the best i've found. Nice and simple!

REPLY

Sathyapriya.n on December 4, 2017 at 1:05 pm

If u will have coding with demo i will soooooooooooooooooo! good.Now itseif that like only having.Thank u sooooooooo much.

REPLY

Tseganeh on December 27, 2017 at 1:05 pm

how do the master know if there is data sent from slave? doesn't there need to be clock signal sent by slave like master do to be synchronize the returning data?

REPLY

deload on June 13, 2018 at 11:09 am

I think the master expects a response from the slave and thus keeps on sending the clock signal (see arrow directions in illustrations) so the slave can timely send data bits back. This leaves the question how the master recognizes that response communication is beginnig as there may be zeroes at the beginning. Maybe response is expected immediately?

REPLY

Prashanthreddy on January 3, 2023 at 1:58 pm

Spi Supports Multiple master's

REPLY

Alin on January 8, 2018 at 12:01 pm

Hello! Nice article, but i think there is a small mistake

Have a nice day!

REPLY

Naveena on March 9, 2018 at 12:06 pm

What is the maximum distance we can use this communication, minimum to maximum Data speed rate and How many slave we can communicate from master practically.

REPLY

Mitch 🐻🐻🚪🎄 on April 26, 2018 at 2:29 pm

Doesn't answer why they don't make MOSI soup.

REPLY

Nithin on May 1, 2018 at 9:49 am

Amazing and succinct explanation! Thank You :)

REPLY

deload on June 13, 2018 at 12:47 pm

Am I right to assume that the term "slave select" is actually wrong? Looking at the daisy-chaned components my understanding is that communication is only possible to all slaves at once. If not could you explain how one slave can be selected among others in the daisy chain?

REPLY

bablu on July 8, 2018 at 5:48 pm

the only website i have ever seen upto now on which i get the sufficient and better information regarding all communication protocols.

REPLY

Rivanda on July 12, 2018 at 9:18 am

Aciu uz straipsni!

REPLY

REPLY

Budi on September 29, 2018 at 9:30 am

Great explanation. Quite simple and easy to understand.
Thanks alot

REPLY

mahesh on October 5, 2018 at 7:28 am

good explanation .
thanks a lot

REPLY

Venkatesh on February 20, 2019 at 3:17 am

Good explanation thank you so much

REPLY

Peyman Amiri on March 5, 2019 at 9:03 am

Thanks alto. So useful and practical !!

REPLY

Peyman Amiri on March 5, 2019 at 9:03 am

Thanks a lot. So useful and practical !!

REPLY

Taver on March 8, 2019 at 11:29 pm

Thanks for the post. Such a wonderful one

REPLY

ahmet on March 28, 2019 at 7:59 am

I have idea about that protocol after read this post
thanks for this

REPLY

configuration? I think a few other people asked with no response. This should be more clear.

REPLY

Jack on March 28, 2019 at 4:14 pm

What is a “significant bit”?

REPLY

Leticia on March 30, 2019 at 11:58 am

Thanks!!

REPLY

Johann Delgado on May 13, 2019 at 5:30 am

Thanks a lot for this explanation

REPLY

giri chowday on July 20, 2019 at 8:54 am

easy to understand ….nice article tqyou….

REPLY

Blenfid on July 26, 2019 at 9:45 am

Bright and clear explanations. I just would like to know how is the election of a specific slave possible in the daisy-chained shape or how is the desired slave be activated in preference to others ?

REPLY

sawan on August 26, 2019 at 5:56 pm

I have a doubt regarding the daisy circuit. Why the MISO of one slave connected to the MOSI of other slave? Shouldn’t all the MOSI come from the master to make the euivalent circuit as above that?

REPLY

abhijeet on September 4, 2019 at 6:23 am

vishal on December 12, 2019 at 5:54 am
Very well explained

REPLY

Constantin on January 10, 2020 at 3:58 pm
Thank you for the detailed and clear explanation!

REPLY

Christian Gingras on April 30, 2020 at 8:57 pm
Comparing the 3 hardware protocol, only full duplex UART allows a slave device to send on it's own some form of message telling the task is completed or a new event happened.
Both I2C and SPI need to use asynchronous polling to verify if the slave finished a task.
In case of SPI EEPROM, for example, there is a status register always available.
In case of I2C, many chip don't answer anything when busy, exactly like if there was a hardware problem. Futhermore, when the I2C device finish, it come back on line at random time, including in the middle of any I2C activity. So, it can easily interpret data as new commands with unpredictable results.

REPLY

Christian Gingras on April 30, 2020 at 9:11 pm
You said in the advantage of SPI over I2C that it is almost twice as fast. Looking at typical datasheet, the maximum speed for both are:
I2C : 400 kilo bit per second
SPI : 20 mega bit per second as master, 4 mega bit per second as slave

There is a new version of I2C that read 800 k, but the equivalent generation SPI is 40 meg. This let me to believe that SPI is more than twice the speed of I2C.

Second comment is about the "SPI Step of transmission" where clock is shown as first step. Chip

The SPI is actually always full duplex, every bit sent by the master is accompanied by a bit answered from the slave.

The steps are:
Step #1: set chip select low
Step #2 start 8 clock pulse with the data 8 bit data (the slave is answering 8 bit at the same time)
Step #3 collect the 8 bit answer from the slave
->Loop to Step #2 as many times as needed for the message length
Step #4 set chip select high ; the slave then analyze the packet and execute whatever command it contained.

The answer from the slave is always one byte late compared to the master.

REPLY

Rajanbabu on May 22, 2020 at 8:01 am
Nice and Great way to teach basics of SPI for beginners.

REPLY

Pavithra on June 15, 2020 at 1:52 pm
Very well explained! Thank you for the simple and neat points.

REPLY

Alfian Nur on February 23, 2021 at 2:25 am
it is really good writing and help me out to understand communication and those pins. Thanks!

REPLY

boopathi on June 26, 2021 at 10:20 am
In SPI has any error checking options ? like if bits not received etc ?

REPLY

REPLY

Peter on October 14, 2022 at 5:02 pm

Exceptionally well explained.

REPLY

channa on May 19, 2023 at 7:05 am

Thank you for the detailed and clear explanation.
If only one slave select pin is available, the slaves can be daisy-chained like this followed by figure as you showed.
In this if master make slave select pin low then it will be effect in all slaves, How it will identifies which salve?
Since there is no addressing associated with salve as in I2C protocol.

REPLY

LEAVE A REPLY

Your email address will not be published. Required fields are marked *

COMMENT

NAME * EMAIL * WEBSITE

☐ Save my name, email, and website in this browser for the next time I comment.

☐ Notify me of follow-up comments by email.

☐ Notify me of new posts by email.

For security, use of Google's reCAPTCHA service is required which is subject to the Google Privacy Policy and Terms of Use.

Copyright **Circuit Basics**