PART 1

**SPI History:**

SPI is a serial communication bus developed by Motorola. It is a full-duplex protocol that functions on a master-slave paradigm that is ideally suited to data stream application.

Serial Peripheral Interface (SPI) is an interface bus commonly used to send data between microcontrollers and small peripherals such as shift registers, sensors, and SD cards. It uses separate clock and data lines, along with a select line to choose the device you wish to talk to.

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.

Devices communicating via SPI are in a master-slave relationship. The master is the controlling device (usually a microcontroller), 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).

**synchronous protocol:**

Synchronous serial communication describes a serial communication protocol in which "data is sent in a continuous stream at a constant rate.

Synchronous communication requires that the clocks in the transmitting and receiving devices are synchronized – running at the same rate – so the receiver can sample the signal at the same time intervals used by the transmitter. No start or stop bits are required. For this reason, synchronous communication permits more information to be passed over a circuit per unit time than asynchronous serial communication. Over time the transmitting and receiving clocks will tend to drift apart, requiring resynchronization.

**master – slave protocol:**

Master/slave is a model of communication where one device or process has unidirectional control over one or more other devices. In some systems a master is selected from a group of eligible devices, with the other devices acting in the role of slaves.

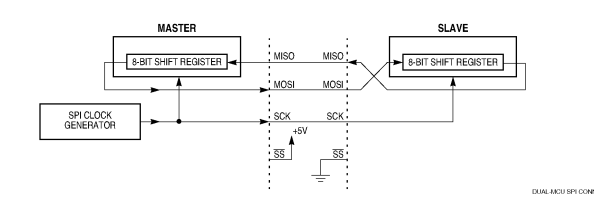
In other words, "The master/slave configuration is basically used for load sharing purposes when two identical motors connected to two different drives are coupled to a common load". One drive is defined as the master and is configured for running in the speed-control mode whereas the other defined as slave is configured for running in torque-control mode.

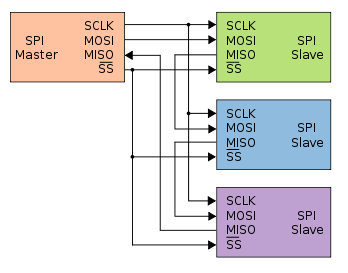
**Clock rates:**

The clock rate typically refers to the frequency at which a chip like a central processing unit (CPU), one core of a multi-core processor, is running and is used as an indicator of the processor's speed. It is measured in clock cycles per second or its equivalent, the SI unit hertz (Hz), the clock rate of the first generation of computers was measured in hertz or kilohertz (kHz), but in the 21st century the speed of modern CPUs is commonly advertised in gigahertz (GHz). This metric is most useful when comparing processors within the same family, holding constant other features that may affect performance. Video card and CPU manufacturers commonly select their highest performing units from a manufacturing batch and set their maximum clock rate higher, fetching a higher price.

**Master manipulates the clock.**

Part 2





SCK pin is an output, It idles high or low, depending on the CPOL bit in the SPCR, until data is written to the shift register.

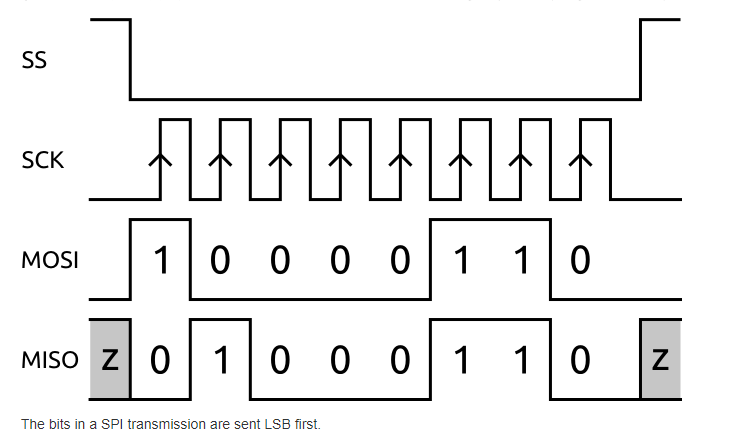
SS: the slave start logic receives a logic low at the SS pin and a clock input at the SCK.

MOSI: data from the master is received serially at the slave MOSI line and loads the 8-bit shift register.

MISO: during a write cycle, data is written into the shift register, then the slave waits for a clock train from the master to shift the data out on the slave’s MISO line.

Part 3:

Embedded electronics is all about interlinking circuits (processors or other integrated circuits) to create a symbiotic system. In order for those individual circuits to swap their information, they must share a common communication protocol. Hundreds of communication protocols have been defined to achieve this data exchange, and, in general, each can be separated into one of two categories: parallel or serial.



The master drives the signals.

An SPI bus has one master and one or more slaves. The master can talk to any slave on the bus, but each slave can only talk to the master. Each slave on the bus must have its own unique slave select signal. The master uses the slave select signals to *select* which *slave* it will be talking to.

As a synchronous serial interface, SPI is not symmetric.