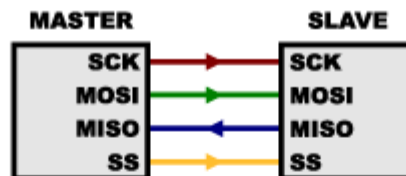
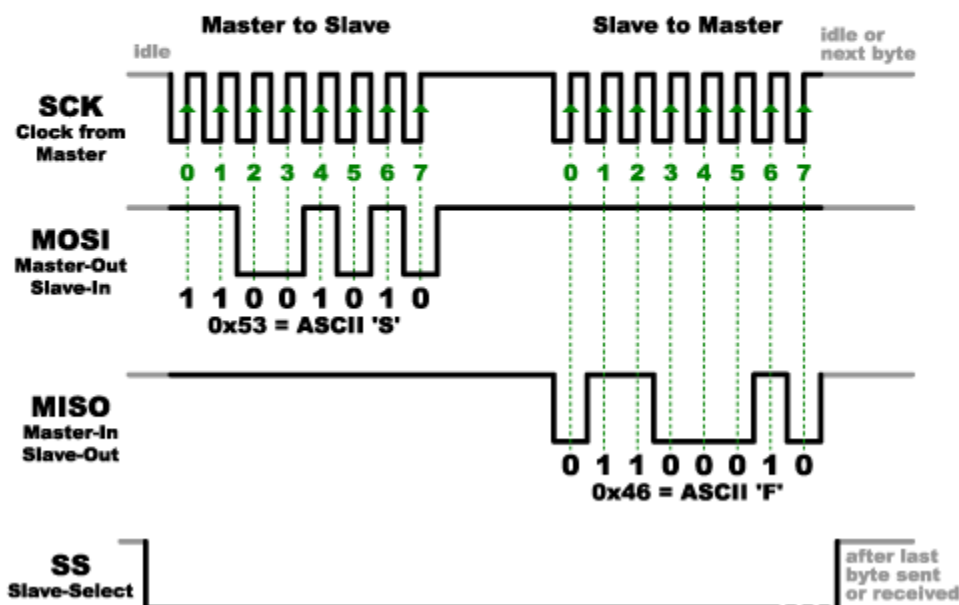


Serial Peripheral Interface (SPI)

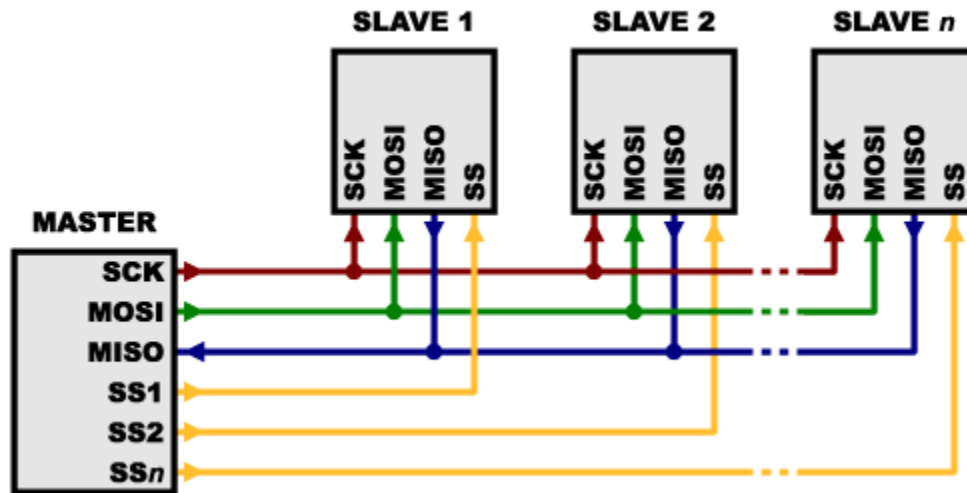
SPI is another very commonly used serial interface for connecting digital devices. The bus involves three wires, the Serial Clock Line (SCK) the Master In Slave Out (MISO) data line, and the Master Out Slave In (MOSI) data line. The SPI interface is a synchronous serial interface like I2C, and the clock line keeps the devices on the bus synchronized. The SPI interface allows bi-directional communications between the master and slave devices on separate dedicated lines. Data going from the master to the slave is sent over the MOSI line. Data going from the slave to the master is sent over the MISO line. Finally, there is a line that is not, strictly speaking, part of the bus called Slave Select (SS), also sometimes referred to as the Chip Select (SS). Each device on the bus has its own dedicated slave select line, which is used to specify which device is currently active on the bus. You can see from the image below how the slave and master are wired in SPI.



In the image below we can analyze some traffic going over the SPI bus. First, you can see that the clock is oscillating from high to low as data is being written to keep the two devices synchronized. Second, you can see the MOSI line going from high to low to write bits from the master to the slave. Next, you see the MISO line going from high to low to send a reply back to the master. Finally, you should see that the SS line is held low (active) for the entire transaction. A device must be active for it to communicate.



The SPI bus allows for as many slave devices as we have general purpose I/O pins to spare for their SS lines. You can see from the image below how the devices all share common SCK, MOSI, and MISO, but how each device has its own dedicated SS line. This makes it trivially easy to select the device with which you want to communicate.



As was the case with I2C, the SPI interface provides a mechanism by which bytes can be read or written between devices. The SPI interface does not implement any device specific functionality. Each device will require a library that encapsulates the procedures a specific device might implement.