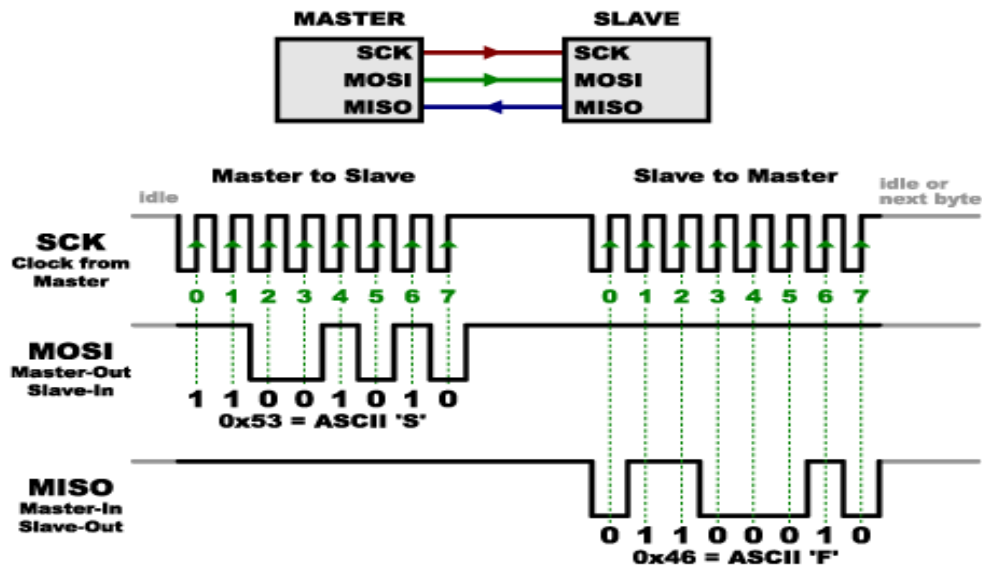


# SPI RECEIVING DATA

You might be thinking to yourself, self, that sounds great for one-way communications, but how do you send data back in the opposite direction? Here's where things get slightly more complicated.

In SPI, only one side generates the clock signal (usually called CLK or SCK for Serial Clock). The side that generates the clock is called the "master", and the other side is called the "slave". There is always only one master (which is almost always your microcontroller), but there can be multiple slaves (more on this in a bit).



When data is sent from the master to a slave, it's sent on a data line called MOSI, for "Master Out / Slave In".

If the slave needs to send a response back to the master, the master will continue to generate a prearranged number of clock cycles, and the slave will put the data onto a third data line called MISO, for “Master In / Slave Out”.

Notice we said “prearranged” in the above description. Because the master always generates the clock signal, it must know in advance when a slave needs to return data and how much data will be returned. This is very different than asynchronous serial, where random amounts of data can be sent in either direction at any time. In practice this isn't a problem, as SPI is generally used to talk to sensors that have a very specific command structure. For example, if you send the command for “read data” to a device, you know that the device will always send you, for example, two bytes in return. (In cases where you might want to return a variable amount of data, you could always return one or two bytes specifying the length of the data and then have the master retrieve the full amount.)

Note that SPI is “full duplex” (has separate send and receive lines), and, thus, in certain situations, you can transmit and receive data *at the same time* (for example, requesting a new sensor reading while retrieving the data from the previous one).

Your device's datasheet will tell you if this is possible.

Source: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>