<http://starter-kit.nettigo.eu/2014/connecting-and-programming-nrf24l01-with-arduino-and-other-boards/> Hack Mty

Library SPI (serial peripheral interface)

Synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices over short distances , it can also used to communicate between 2 microcontrollers.

With SPI communication there is always one master device(usually a microcontroller) which controls the peripheral device.Tipically there are 3 lines common to all of devices.

MISO: Master in slave out – The Slave line for sending data to the master,

MOSI: Master out slave in - The Master line for sending data to the peripherals.

SCK: Serial clock – The clock pulses which synchronize data transmission generated by the master.

And one line specific for every device :

SS( Slave select) – the pin on each device that the master can use to enable and disable specific devices

When a device's Slave Select pin is low, it communicates with the master. When it's high, it ignores the master. This allows you to have multiple SPI devices sharing the same MISO, MOSI, and CLK lines.

To write code for a new SPI you need to take into consideration the following

What is the maximum SPI speed your device can use? This is controlled by the first parameter in SPISettings. If you are using a chip rated at 15 MHz, use 15000000. Arduino will automatically use the best speed that is equal to or less than the number you use with SPISettings.

Is data shifted in Most Significant Bit (MSB) or Least Significant Bit (LSB) first? This is controlled by second SPISettings parameter, either MSBFIRST or LSBFIRST. Most SPI chips use MSB first data order.

Is the data clock idle when high or low? Are samples on the rising or falling edge of clock pulses? These modes are controlled by the third parameter in SPISettings.

he SPI standard is loose and each device implements it a little differently. This means you have to pay special attention to the device's datasheet when writing your code.

Generally speaking, there are four modes of transmission. These modes control whether data is shifted in and out on the rising or falling edge of the data clock signal (called the clock phase), and whether the clock is idle when high or low (called the clock polarity). The four modes combine polarity and phase according to this table: