SD card with ESP8266

The communication between the ESP8266 and the SD card uses Serial Peripheral Interface (SPI), it is a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances. It can also be used for communication between two ESP8266. With an SPI connection there is always one master device (usually a microcontroller) which controls the peripheral devices. Typically there are three lines common to all the devices as follows:

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.

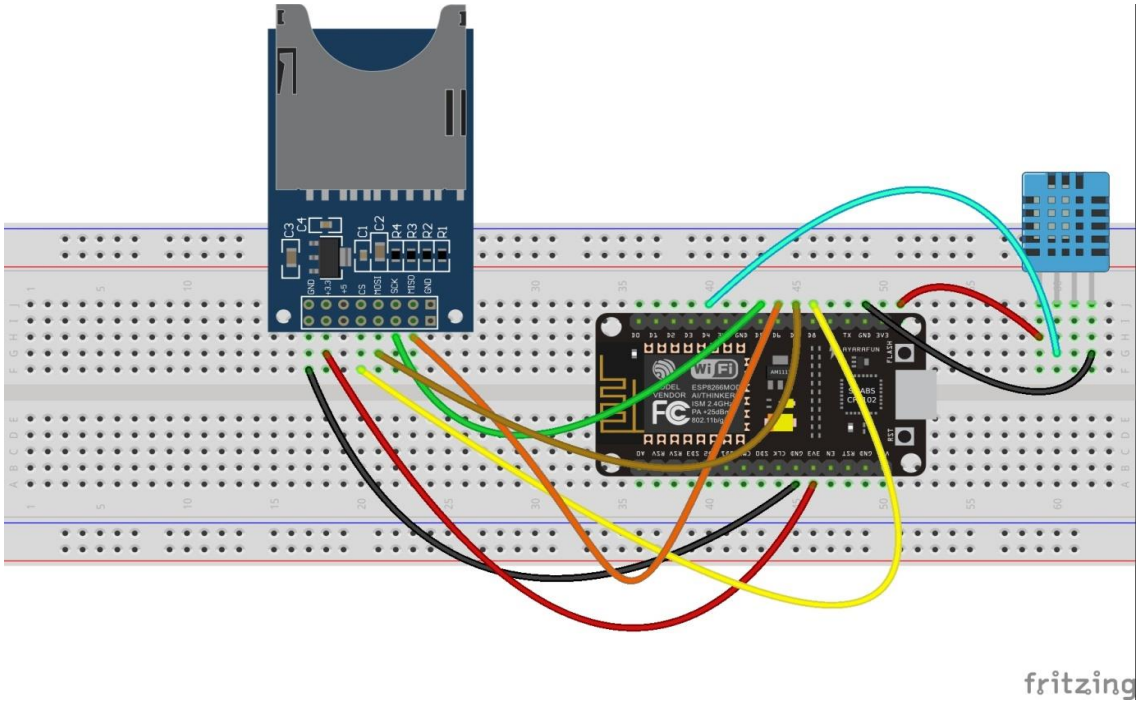
SCK (Serial Clock) - The clock pulses which synchronize data transmission generated by the master

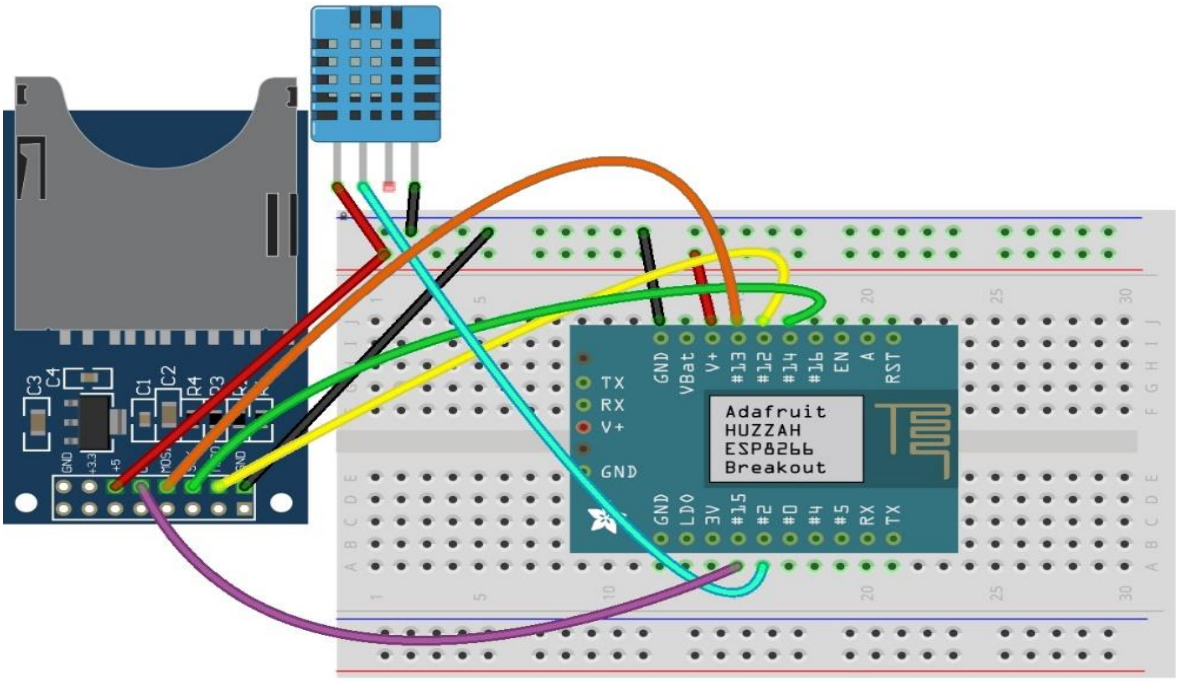
SS (Slave Select) - the pin on each device that the master can use to enable and disable specific devices.

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.

Preparing the SD card to use with ESP8266

Most of the SD cards are supported by ESP8266 the Arduino library supports SD card to have the format type FAT16 or FAT32, make sure you format your SD card before using it with ESP8266.





You need a SD card Module with a SD card and a DHT Temperature and Humidity sensor with Nodemcu to make this example work.

Connect the SD card module as follows to the Nodemcu.

D5 (GPIO 14 in generic boards) pin of Nodemcu to SCK (GPIO 14 in generic boards)

D6 (GPIO 12 in generic boards) pin of Nodemcu to MISO

D7 (GPIO 13 in generic boards) pin of Nodemcu to MOSI

D8 (GPIO 15 in generic boards) pin of Nodemcu to CS

Vcc to 3.3v

Gnd to Gnd

This example will store the temperature and humidity data on SD card connected to the Nodemcu, the temperature data's can be later on used for inspection by connecting the SD card to the card reader.

Arduino Ide program for storing DHT11 sensor data into an SD card

```
#include <SD.h>
#include <SPI.h>
#include "DHT.h"
#define DHTPIN 2
File Dhtemp;
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

void setup()
{
 // Open serial communications and wait for port to open:
 Serial.begin(9600);
 dht.begin();
 while (!Serial) {
  ; // wait for serial port to connect. Needed for Leonardo only
 }



 Serial.print("Initializing SD card...");

 if (!SD.begin(15)) {
  Serial.println("initialization failed!");
  return;
 }
 Serial.println("initialization done.");
```

```
}

void loop()
{
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();



   if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
        return;
  }


    Dhtemp = SD.open("Temp.txt", FILE_WRITE);

  // if the file opened okay, write to it:
  if (Dhtemp) {
    Serial.print("Humidity:    ");
    Serial.print(h);
    Serial.print("%\t");
    Serial.print("Temperature:    ");
    Serial.print(t);
    Serial.println("*C");
    Serial.print("Writing to test.txt...");
    Dhtemp.print("Humidity:  ");
```

```
  Dhtemp.print(h);
  Dhtemp.print("%\t");
  Dhtemp.print("Temperature:   ");
  Dhtemp.print(t);
  Dhtemp.println("*C");
  // close the file:
  Dhtemp.close();
  Serial.println("done.");
 } else {
  // if the file didn't open, print an error:
  Serial.println("error opening Temp.txt");
 }
}
```

This code is really simple to follow as it uses the predefined Arduino library to write the data's of temperature and humidity to the SD card. It's just like writing the data to the serial monitor, copy and paste the above code into Arduino IDE and upload the code to the Nodemcu, if you are not getting the data or the file is not saved properly in your SD card you can check the SD card whether you inserted it correctly or not, if the problem persists format the SD card and check it again.