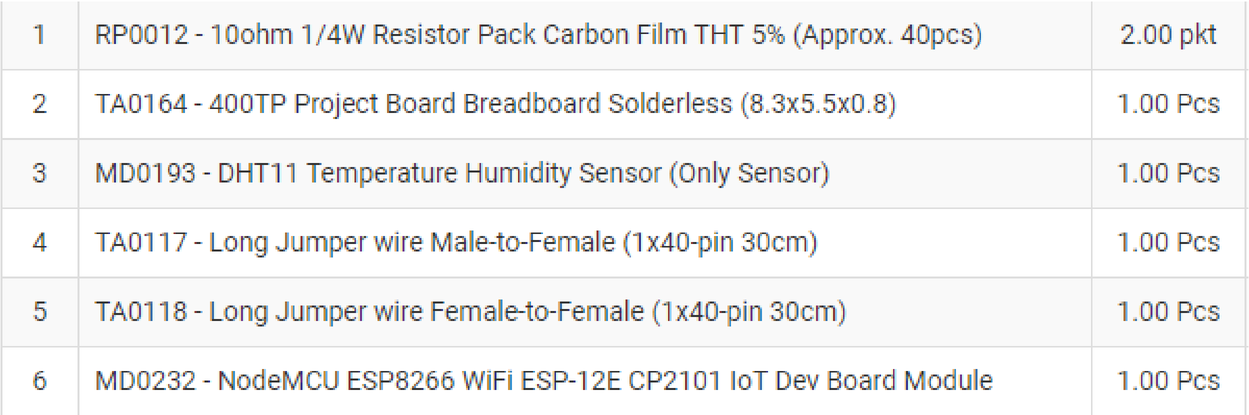
INTERNET OF THINGS

COURSE WORK

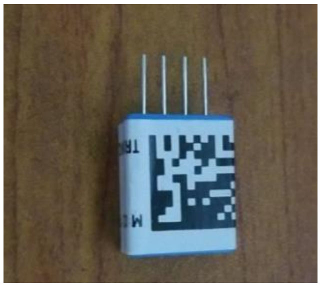
COHNDNE20.1f-021
JADHUSAN M .S

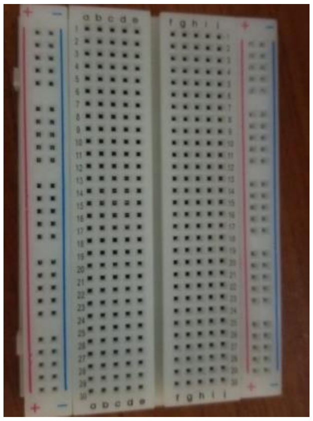
REQUIRMENTS



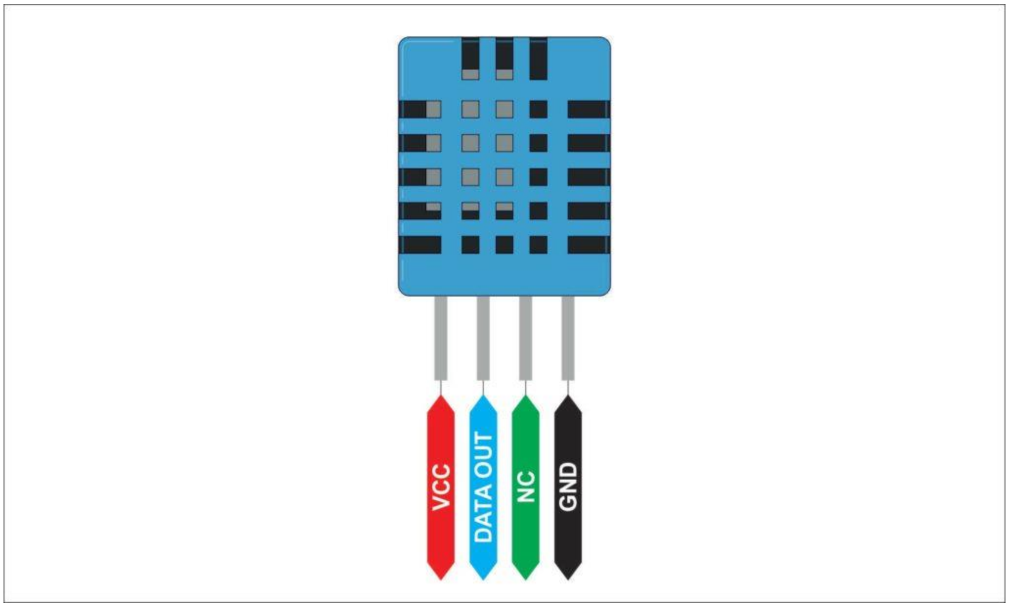
1	RP0012 - 10ohm 1/4W Resistor Pack Carbon Film THT 5% (Approx. 40pcs)	2.00 pkt
2	TA0164 - 400TP Project Board Breadboard Solderless (8.3x5.5x0.8)	1.00 Pcs
3	MD0193 - DHT11 Temperature Humidity Sensor (Only Sensor)	1.00 Pcs
4	TA0117 - Long Jumper wire Male-to-Female (1x40-pin 30cm)	1.00 Pcs
5	TA0118 - Long Jumper wire Female-to-Female (1x40-pin 30cm)	1.00 Pcs
6	MD0232 - NodeMCU ESP8266 WiFi ESP-12E CP2101 IoT Dev Board Module	1.00 Pcs

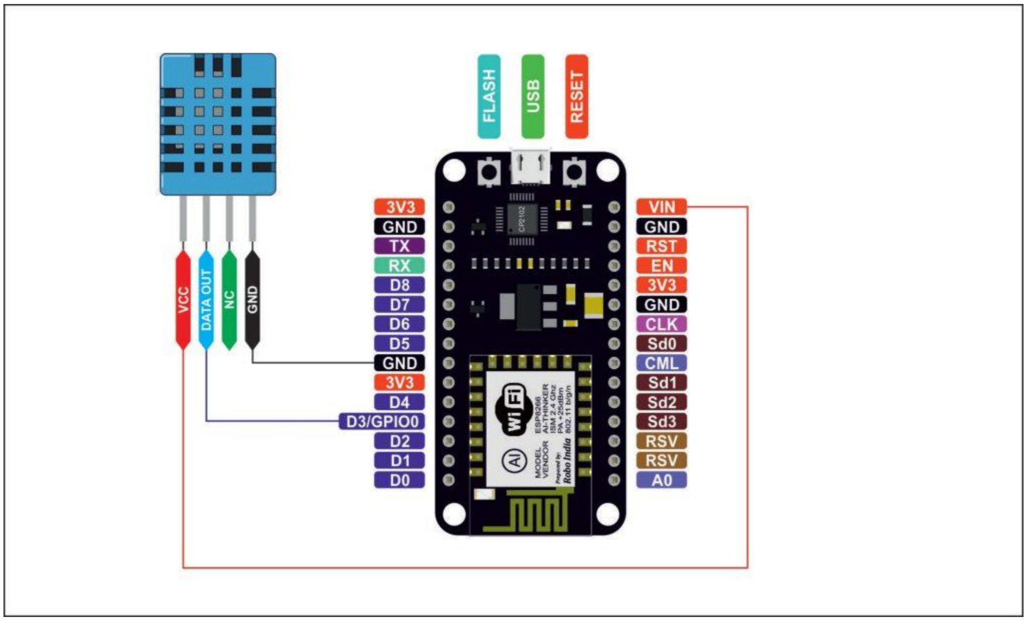
PRODUCTS

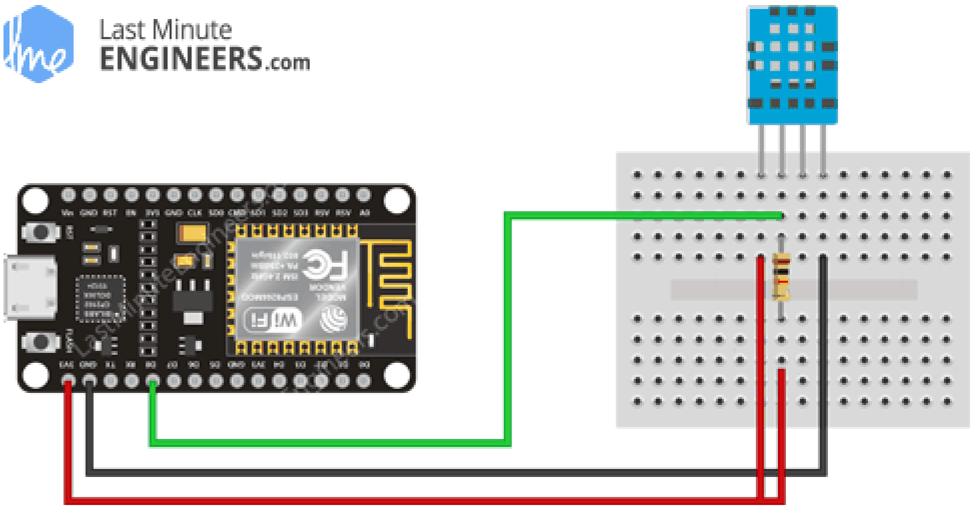




DIAGRAMS

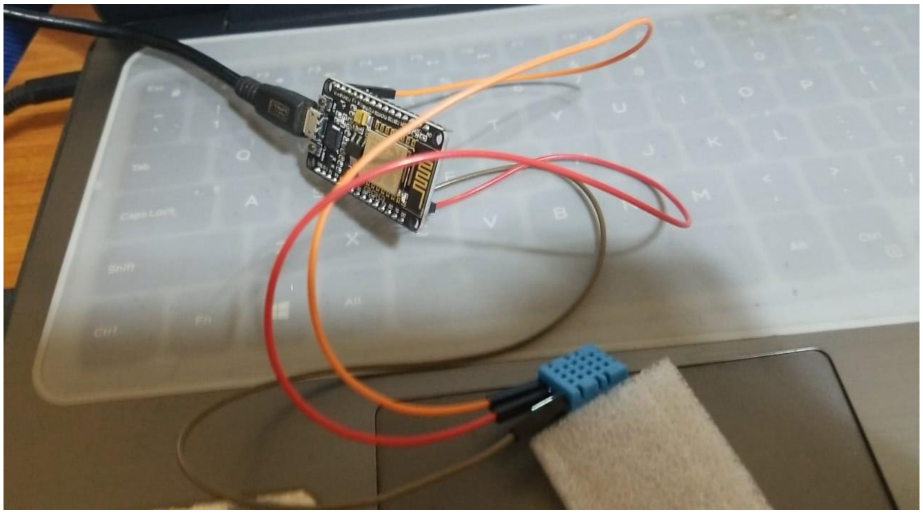

VCC
DATA OUT
NC
GND

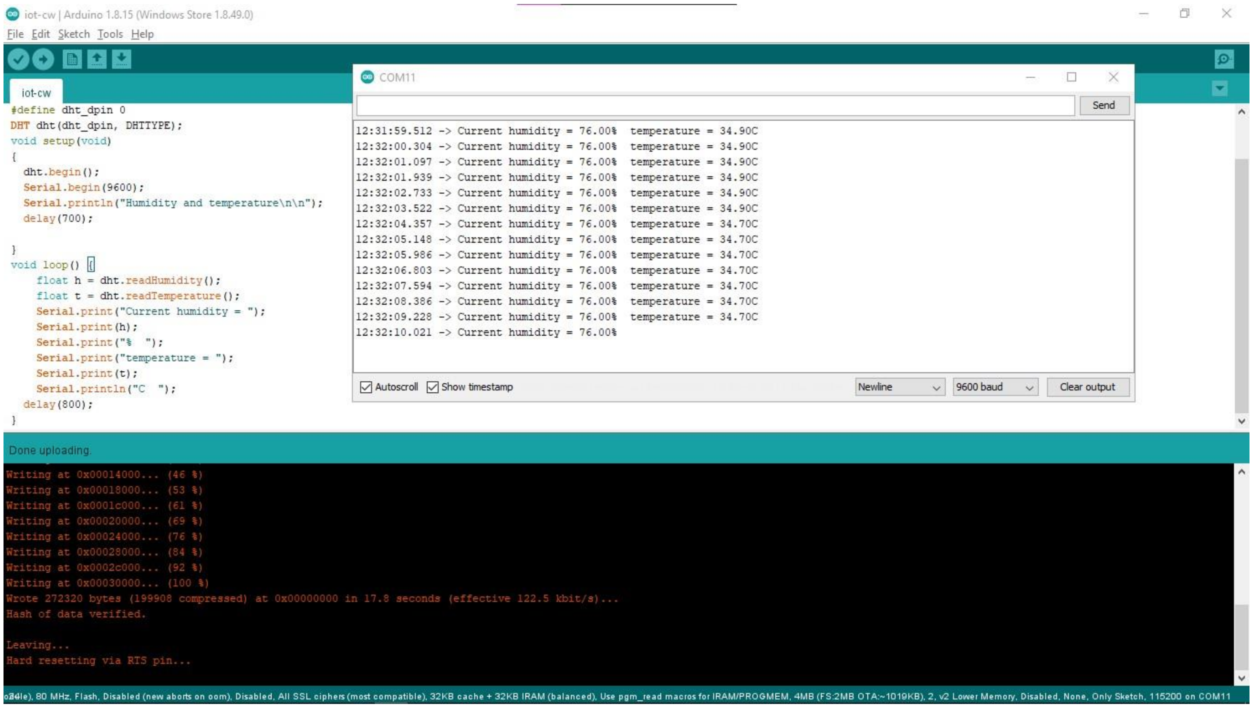

VCC
DATA OUT
NC
GND
FLASH
USB
RESET
3V3
GND
TX
RX
D8
D7
D6
D5
GND
3V3
D4
D3/GPIO0
D2
D1
D0
VIN
GND
RST
EN
3V3
GND
CLK
Sd0
CML
Sd1
Sd2
Sd3
RSV
RSV
A0
CP2102
Wi Fi
AI
MODEL
VENDOR
ESP8266
AI-THINKER
ISM 2.4 Ghz
PA +25dBm
802.11 b/g/n
Prepared by
Robo India



Now place the sensor on to your breadboard besides NodeMCU. Connect VCC pin on the sensor to the 3.3V pin on the NodeMCU and ground to ground. Also connect Data pin on the sensor to D8 pin of the ESP8266 NodeMCU. Finally, we need to place a pull-up resistor of 10KΩ between VCC and data line to keep it HIGH for proper communication between sensor and NodeMCU.

Testing without breadboard




iot-cw | Arduino 1.8.15 (Windows Store 1.8.49.0)
File Edit Sketch Tools Help
iot-cw
COM11
Send
☑ Autoscroll ☑ Show timestamp
Newline
9600 baud
Clear output
Done uploading.

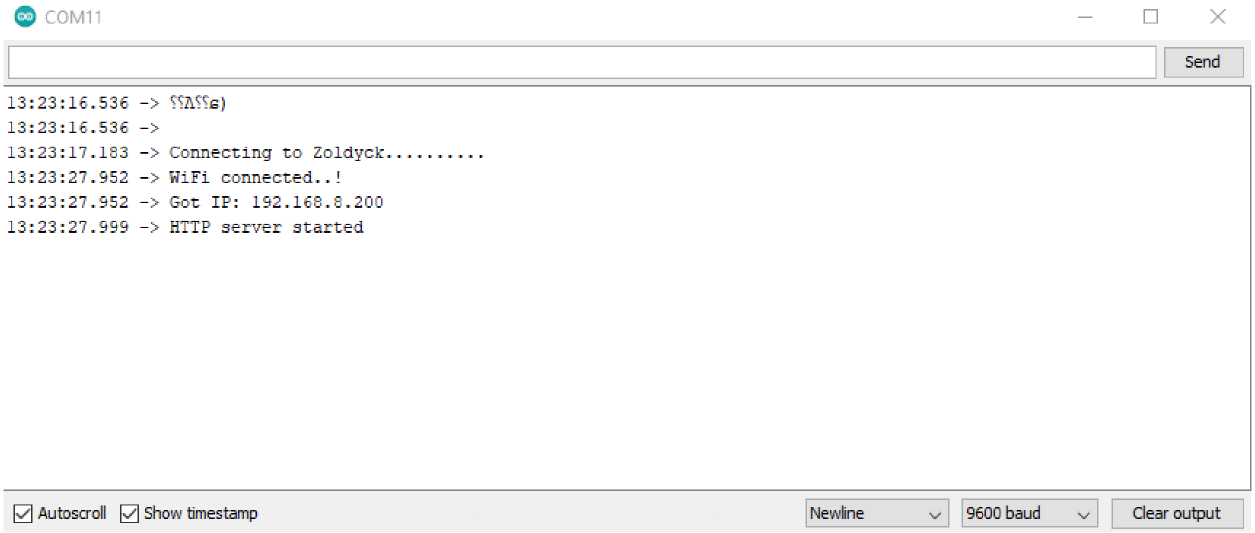
```
#define dht_dpin 0
DHT dht(dht_dpin, DHTTYPE);
void setup(void)
{
  dht.begin();
  Serial.begin(9600);
  Serial.println("Humidity and temperature\n\n");
  delay(700);

}
void loop() {
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    Serial.print("Current humidity = ");
    Serial.print(h);
    Serial.print("%  ");
    Serial.print("temperature = ");
    Serial.print(t);
    Serial.println("C  ");
  delay(800);
}
```

```
12:31:59.512 -> Current humidity = 76.00%  temperature = 34.90C
12:32:00.304 -> Current humidity = 76.00%  temperature = 34.90C
12:32:01.097 -> Current humidity = 76.00%  temperature = 34.90C
12:32:01.939 -> Current humidity = 76.00%  temperature = 34.90C
12:32:02.733 -> Current humidity = 76.00%  temperature = 34.90C
12:32:03.522 -> Current humidity = 76.00%  temperature = 34.90C
12:32:04.357 -> Current humidity = 76.00%  temperature = 34.70C
12:32:05.148 -> Current humidity = 76.00%  temperature = 34.70C
12:32:05.986 -> Current humidity = 76.00%  temperature = 34.70C
12:32:06.803 -> Current humidity = 76.00%  temperature = 34.70C
12:32:07.594 -> Current humidity = 76.00%  temperature = 34.70C
12:32:08.386 -> Current humidity = 76.00%  temperature = 34.70C
12:32:09.228 -> Current humidity = 76.00%  temperature = 34.70C
12:32:10.021 -> Current humidity = 76.00%
```

```
Writing at 0x00014000... (46 %)
Writing at 0x00018000... (53 %)
Writing at 0x0001c000... (61 %)
Writing at 0x00020000... (69 %)
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 272320 bytes (199908 compressed) at 0x00000000 in 17.8 seconds (effective 122.5 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

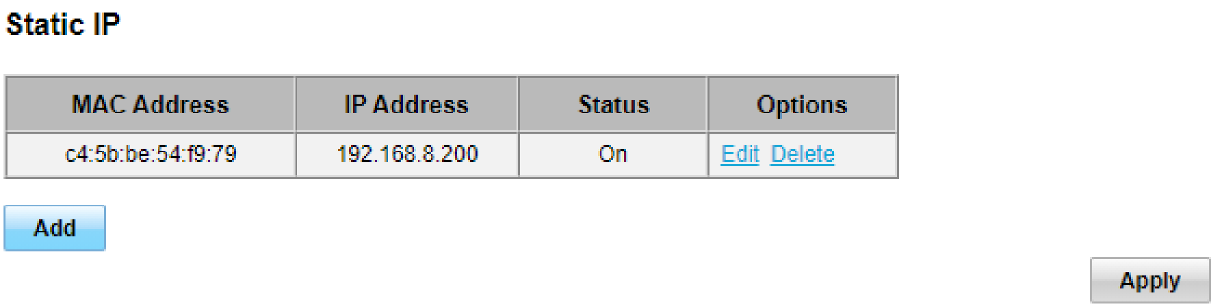
o24le), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM11

Create ESP8266 NodeMCU Web Server using WiFi Station (STA) mode



```
13:23:16.536 -> ʕʕΛʕʕɘ)
13:23:16.536 ->
13:23:17.183 -> Connecting to Zoldyck..........
13:23:27.952 -> WiFi connected..!
13:23:27.952 -> Got IP: 192.168.8.200
13:23:27.999 -> HTTP server started
```

Connecting to my Home network. And I’m using here a static IP address.



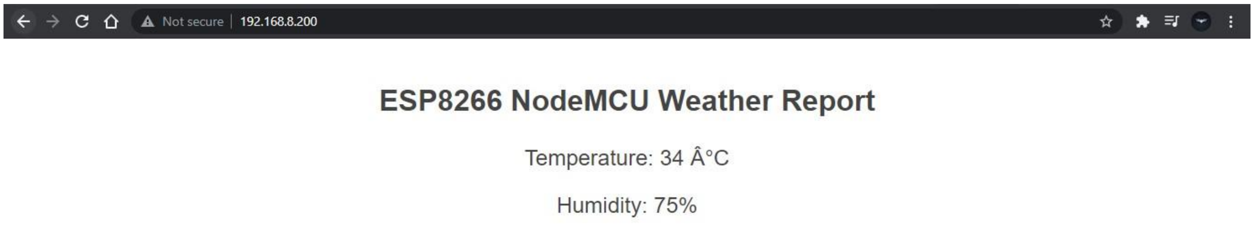
Static IP

MAC Address	IP Address	Status	Options
c4:5b:be:54:f9:79	192.168.8.200	On	Edit Delete

Accessing the Web Server

Next, load up a browser and point it to the IP address shown on the serial monitor. The ESP8266 NodeMCU should serve up a web page showing temperature and relative humidity.

Using the Static Ip Address we can access the page

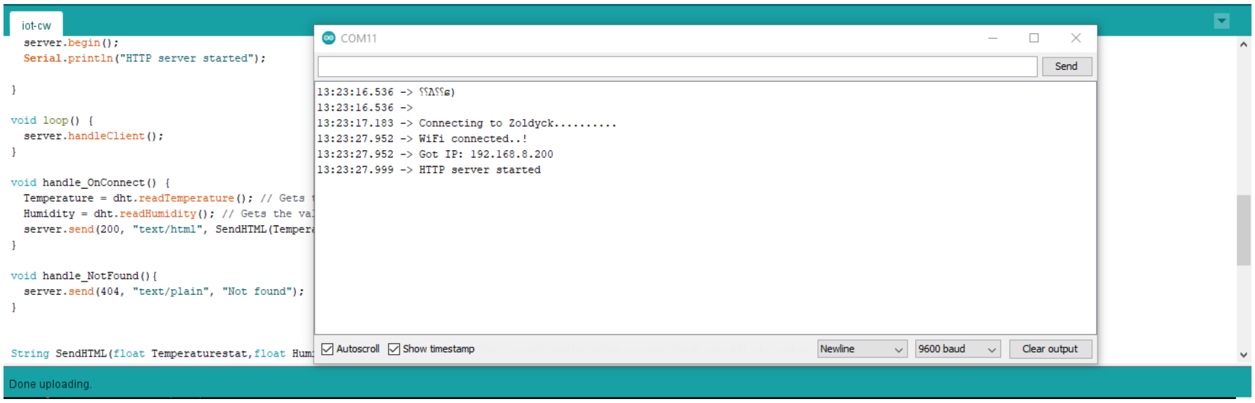


ESP8266 NodeMCU Weather Report

Temperature: 34 Â°C

Humidity: 75%

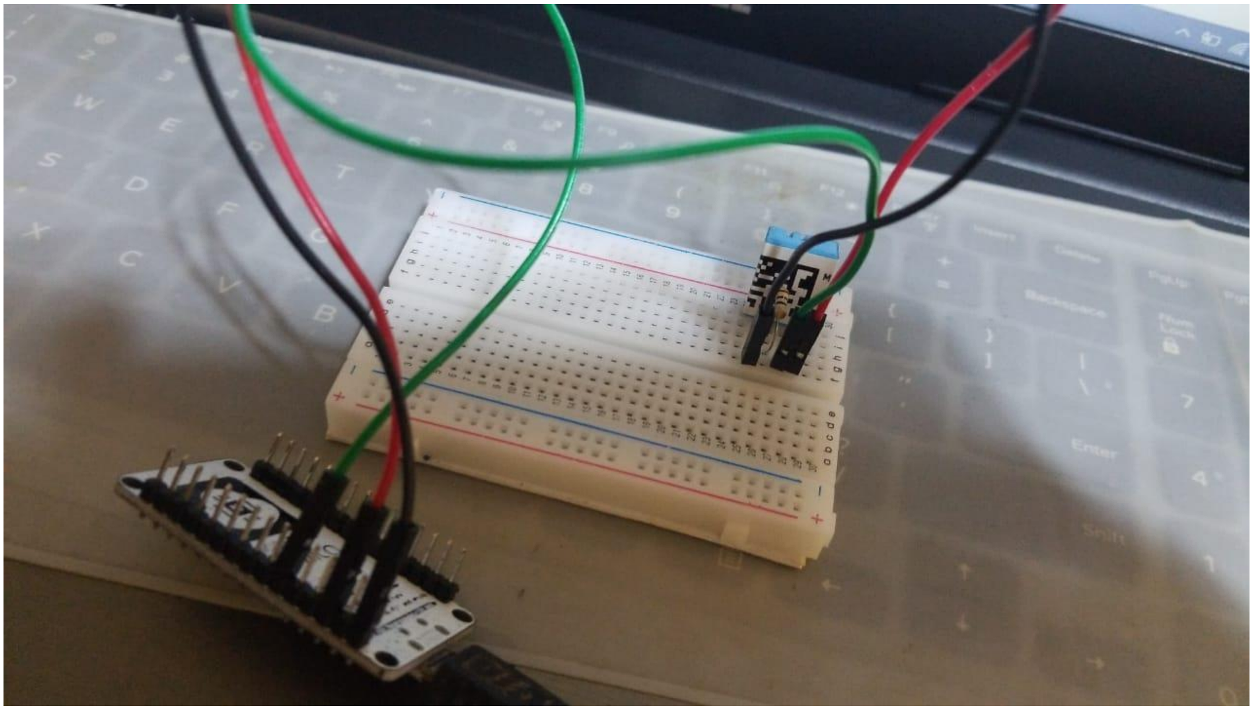
DEVICE SETUP



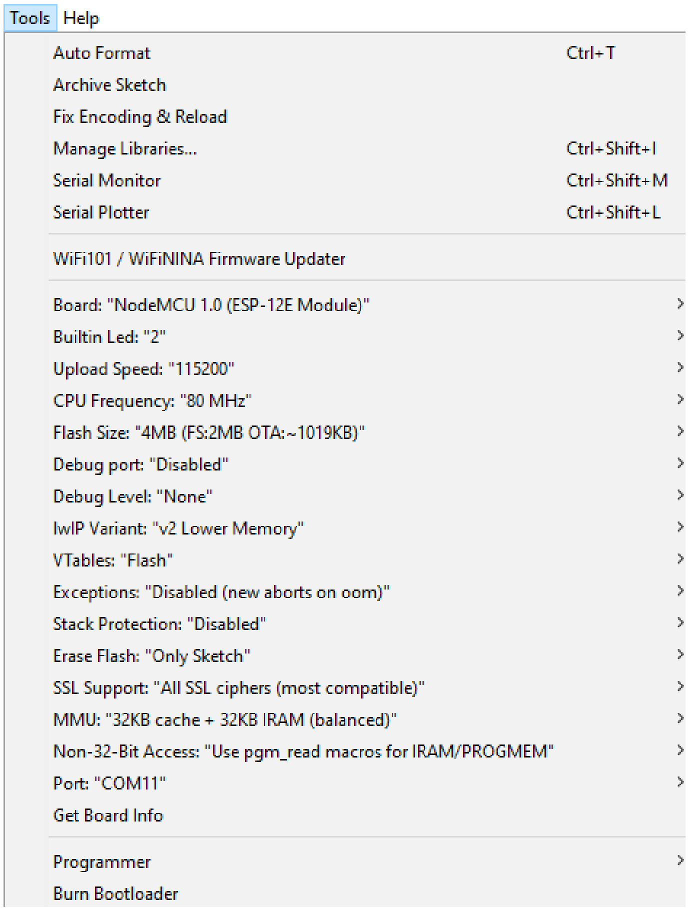
```
  server.begin();
  Serial.println("HTTP server started");

}

void loop() {
  server.handleClient();
}

void handle_OnConnect() {
  Temperature = dht.readTemperature(); // Gets t
  Humidity = dht.readHumidity(); // Gets the val
  server.send(200, "text/html", SendHTML(Tempera
}

void handle_NotFound(){
  server.send(404, "text/plain", "Not found");
}


String SendHTML(float Temperaturestat,float Humi
```

```
13:23:16.536 -> ␚␚Λ␚␚ɞ)
13:23:16.536 ->
13:23:17.183 -> Connecting to Zoldyck..........
13:23:27.952 -> WiFi connected..!
13:23:27.952 -> Got IP: 192.168.8.200
13:23:27.999 -> HTTP server started
```

☑ Autoscroll ☑ Show timestamp Newline 9600 baud



ARDUINO SETUP



Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L

WiFi101 / WiFiNINA Firmware Updater

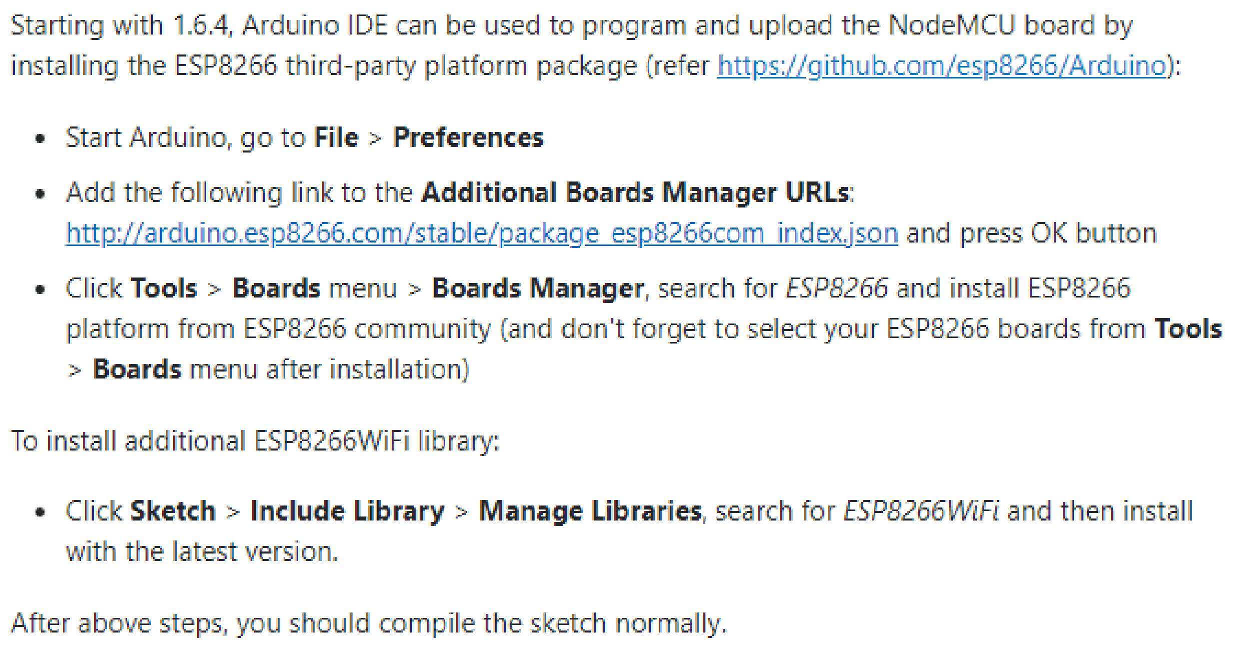
Board: "NodeMCU 1.0 (ESP-12E Module)"
Builtin Led: "2"
Upload Speed: "115200"
CPU Frequency: "80 MHz"
Flash Size: "4MB (FS:2MB OTA:~1019KB)"
Debug port: "Disabled"
Debug Level: "None"
lwIP Variant: "v2 Lower Memory"
VTables: "Flash"
Exceptions: "Disabled (new aborts on oom)"
Stack Protection: "Disabled"
Erase Flash: "Only Sketch"
SSL Support: "All SSL ciphers (most compatible)"
MMU: "32KB cache + 32KB IRAM (balanced)"
Non-32-Bit Access: "Use pgm_read macros for IRAM/PROGMEM"
Port: "COM11"
Get Board Info

Programmer
Burn Bootloader

CODE

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include "DHT.h"

#define DHTTYPE DHT11   // DHT 11
#define dht_dpin 0
DHT dht(dht_dpin, DHTTYPE);



/*Put your SSID & Password*/
const char* ssid = "Zoldyck";  // Enter SSID here
const char* password = "killua666";  //Enter Password here
ESP8266WebServer server(80);
float Temperature;
float Humidity;
void setup(void)
{
  dht.begin();
  Serial.begin(9600);
  Serial.println("Humidity and temperature\n\n");
  delay(700);
  Serial.print("Connecting to "); Serial.print(ssid);
  //connect to your local wi-fi network
  WiFi.begin(ssid, password);
  //check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected..!");
  Serial.print("Got IP: ");  Serial.println(WiFi.localIP());
  server.on("/", handle_OnConnect);
  server.onNotFound(handle_NotFound);
  server.begin();
  Serial.println("HTTP server started");
}
void loop() {
  server.handleClient();
}

void handle_OnConnect() {
  Temperature = dht.readTemperature(); // Gets the values of the temperature
  Humidity = dht.readHumidity(); // Gets the values of the humidity
  server.send(200, "text/html", SendHTML(Temperature,Humidity));
}

void handle_NotFound(){
  server.send(404, "text/plain", "Not found");
}


String SendHTML(float Temperaturestat,float Humiditystat){
   String ptr = "<!DOCTYPE html> <html>\n";
  ptr +="<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">\n<meta http-equiv=\"refresh\" content=\"5\">";
  ptr +="<title>ESP8266 Weather Report</title>\n";
  ptr +="<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}\n";
  ptr +="body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;}\n";
  ptr +="p {font-size: 24px;color: #444444;margin-bottom: 10px;}\n";
  ptr +="</style>\n";
  ptr +="</head>\n";
  ptr +="<body>\n";
  ptr +="<div id=\"webpage\">\n";
  ptr +="<h1>ESP8266 NodeMCU Weather Report</h1>\n";

  ptr +="<p>Temperature: ";
  ptr +=(int)Temperaturestat;
  ptr +=" °C</p>";
  ptr +="<p>Humidity: ";
  ptr +=(int)Humiditystat;
  ptr +="%</p>";

  ptr +="</div>\n";
  ptr +="</body>\n";
  ptr +="</html>\n";
  return ptr;
 }
```

FIXING ERRORS



Starting with 1.6.4, Arduino IDE can be used to program and upload the NodeMCU board by installing the ESP8266 third-party platform package (refer https://github.com/esp8266/Arduino):

- Start Arduino, go to **File** > **Preferences**
- Add the following link to the **Additional Boards Manager URLs**: http://arduino.esp8266.com/stable/package_esp8266com_index.json and press OK button
- Click **Tools** > **Boards** menu > **Boards Manager**, search for *ESP8266* and install ESP8266 platform from ESP8266 community (and don't forget to select your ESP8266 boards from **Tools** > **Boards** menu after installation)

To install additional ESP8266WiFi library:

- Click **Sketch** > **Include Library** > **Manage Libraries**, search for *ESP8266WiFi* and then install with the latest version.

After above steps, you should compile the sketch normally.