DOP: / /2023 DOS: / / 2023

Experiment No: 04

Aim: - Write a program for ESP8266 DHT11/DHT22 Temperature and Humidity Web Server with Arduino IDE

Theory:

The ESP8266 is a low-cost, Wi-Fi enabled microcontroller that can be used for a variety of applications. The DHT11 and DHT22 are two popular temperature and humidity sensors that can be interfaced with the ESP8266 to create a simple weather monitoring system.

The ESP8266 is a low-cost Wi-Fi microchip with full TCP/IP stack capability. It can be programmed using the Arduino IDE and is a popular choice for Internet of Things (IoT) projects due to its affordability and ease of use. The DHT11 and DHT22 are two popular types of temperature and humidity sensors that can be used with the ESP8266.

The DHT11 sensor is a basic, low-cost digital temperature and humidity sensor. It can measure temperatures from 0 to 50 degrees Celsius with an accuracy of ±2 degrees Celsius, and relative humidity from 20% to 90% with an accuracy of ±5%. The DHT22 sensor, also known as the AM2302, is a more advanced version of the DHT11. It can measure temperatures from -40 to 80 degrees Celsius with an accuracy of ±0.5 degrees Celsius, and relative humidity from 0% to 100% with an accuracy of ±2%. Both sensors use a single-wire serial interface to communicate with the ESP8266.

To use the DHT11 or DHT22 with the ESP8266, you will need to connect the sensor to the microcontroller using a pull-up resistor and a voltage divider. The pull-up resistor is used to ensure that the data line remains high when it is not being used, while the voltage divider is used to reduce the voltage level of the data line to a level that can be read by the ESP8266.

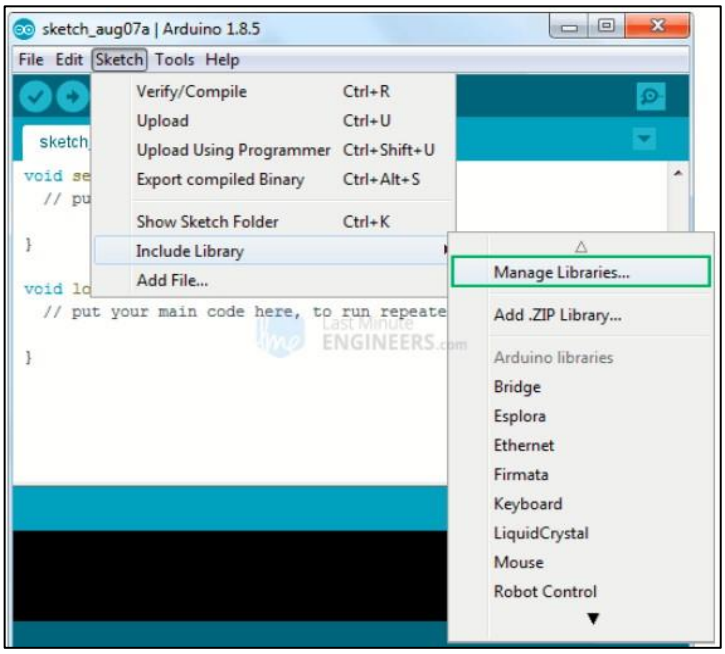
Once the sensor is connected to the ESP8266, you can use a library such as the Adafruit DHT library to read temperature and humidity values from the sensor. The library communicates with the sensor using the single-wire serial interface and provides a simple API for reading temperature and humidity values.

It is important to note that the DHT11 and DHT22 sensors are sensitive to timing and can be affected by environmental factors such as temperature and humidity. It is recommended to take multiple readings and average the values to improve the accuracy of the sensor readings. Additionally, the DHT11 and DHT22 sensors are not recommended for applications where high accuracy is required, as they have a limited measurement range and accuracy compared to more advanced sensors.

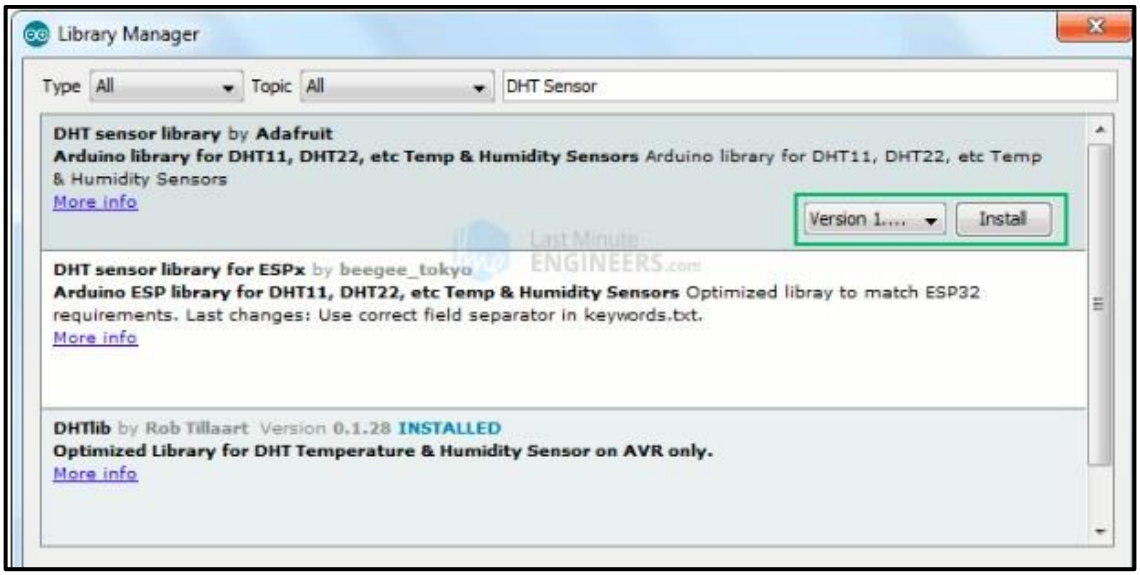
• Steps for reading sensor data.

Step 1: To begin reading sensor data, you will need to install the DHT sensor library. It is available from the Arduino library manager.

To install the library, navigate to Sketch > Include Library > Manage Libraries… Wait for the Library Manager to download the libraries index and update the list of installed libraries.



Filter your search by entering 'DHT sensor'. Look for the DHT sensor library by Adafruit. Click on that entry and then choose Install.



The DHT sensor library makes use of the Adafruit Sensor support backend. So, look for Adafruit Unified Sensor in the library manager and install it as well.

▪ Creating an ESP8266 Web Server using WiFi Station (STA) mode :

As the title implies, we will configure an ESP8266 web server in Station (STA) mode to serve web pages to any connected client on the existing network.

Before you begin uploading the sketch, you must replace the following two variables with your network credentials so that the ESP8266 can connect to an existing network.

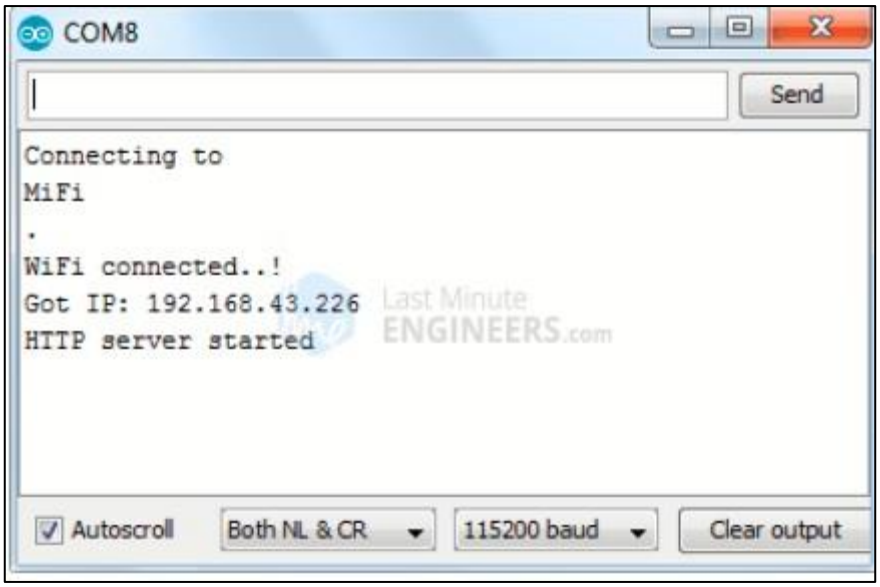
INPUT:

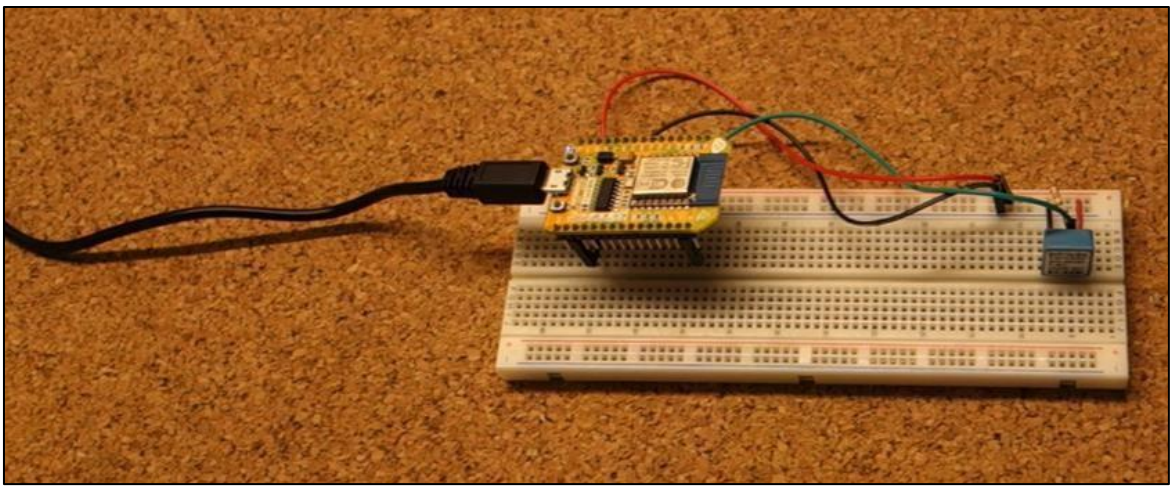
```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include "DHT.h"

// Uncomment one of the lines below for whatever DHT sensor type you're using!
//#define DHTTYPE DHT11   // DHT 11
//#define DHTTYPE DHT21   // DHT 21 (AM2301)
#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321

/*Put your SSID & Password*/ const char* ssid = "YourNetworkName";
// Enter SSID here const char* password = "YourPassword";  //Enter
Password here ESP8266WebServer server(80);

// DHT Sensor uint8_t
DHTPin = D8;

// Initialize DHT sensor.
DHT dht(DHTPin, DHTTYPE);

float Temperature;
float Humidity;

void setup() {
Serial.begin(115200);   delay(100);

  pinMode(DHTPin, INPUT);

  dht.begin();

  Serial.println("Connecting to ");
  Serial.println(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, initialscale=1.0,
user-scalable=no\">\n";
  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;
}
```

Accessing the Web Server:

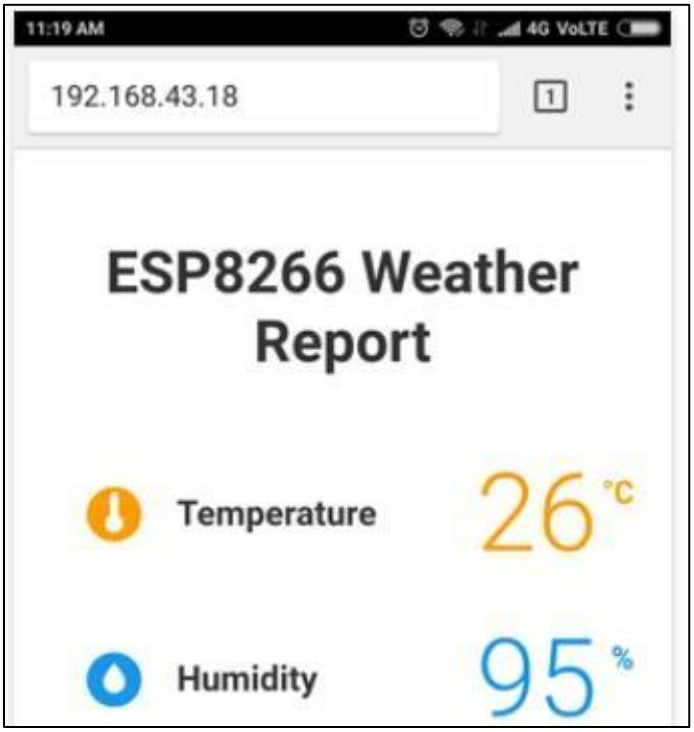
After uploading the sketch, open the Serial Monitor at 115200 baud and press the RESET button on the Node MCU. If everything is fine, it will display the dynamic IP address obtained from your router as well as the “HTTP server started” message.





Next, launch a browser and navigate to the IP address displayed on the serial monitor. The ESP8266 should serve a web page with the current temperature and relative humidity.

Result:-



Conclusion:

Thus, we were able to write a program for ESP8266 DHT11/DHT22 Temperature and Humidity Web Server with Arduino IDE. It is important to note that the DHT11 and DHT22 sensors are sensitive to timing and can be affected by environmental factors such as temperature and humidity