**EMBEDDED SYSTEMS**

**(EL-419)**

**LABORATORY MANUAL**

**SPRING 2019**

****

**Engr. Aneela Sabir**

## Monitoring temperature and humdity data through Adafruit io

**(LAB # 07 & 08)**

Student Name: \_Kamran, Moiz Hussain, Aqib Khan\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Roll No: \_i140420,i15-443,402\_\_\_\_\_\_\_\_\_\_Section: \_A\_\_\_

Date performed: \_\_\_13/3\_\_\_\_\_\_\_\_\_\_, 2019

**MARKS AWARDED: \_\_\_\_\_\_\_\_/ 10**

## Lab # 07 & 08: Monitoring temperature and humidity data through Adafruit io.

**Learning Objectives:**

1. Introduction to DHT11 temperature and humidity sensor
2. How to send your sensor data to Adafruit io

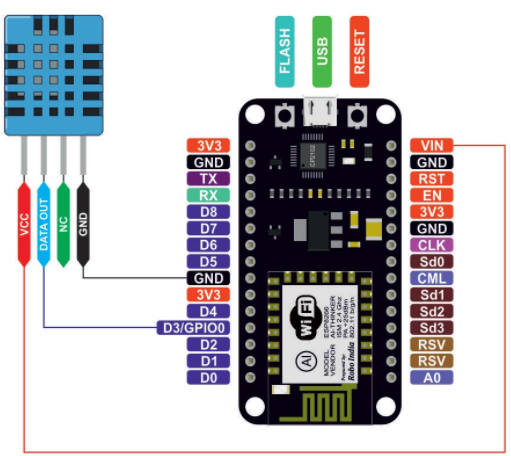
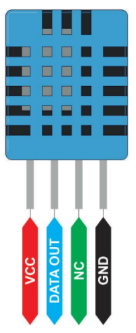
#### **Equipment Required:**

1. NodeMCU ESP8266 Breakout Board
2. USB-A to micro-USB Cable
3. DHT11 sensor
4. Jumper wires
5. Bread board
6. Arduino IDE

**Introduction:**

**DHT11 Sensor:**

The DHT11 is chosen because it is lab calibrated, accurate and stable and its signal output is digital. Most important of all, it is relatively inexpensive for the given performance.  Pinout of the sensor is shown in the figure. Connection diagram of DHT11 and NodeMcu ESP8266 is also given below.



**DHT11 Library for Arduino IDE:**

Download library for DHT sensor uploaded on SLATE->Embedded Systems Resources -> Lab Manuals ->Lab 7 & 8 and compy it to C:\……..\Documents\Arduino\libraries

**Example Code:** Given code is used to interface DHT11 with ESP8266. The code will display Temperature and Humidity value on Serial Monitor.

#include "DHT.h" // including the library of DHT11 temperature and humidity sensor

#define DHTTYPE DHT11 // DHT 11

#define dht\_dpin 0

DHT dht(dht\_dpin, DHTTYPE);

void setup(void)

{

dht.begin();

Serial.begin(115200);

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 = ");

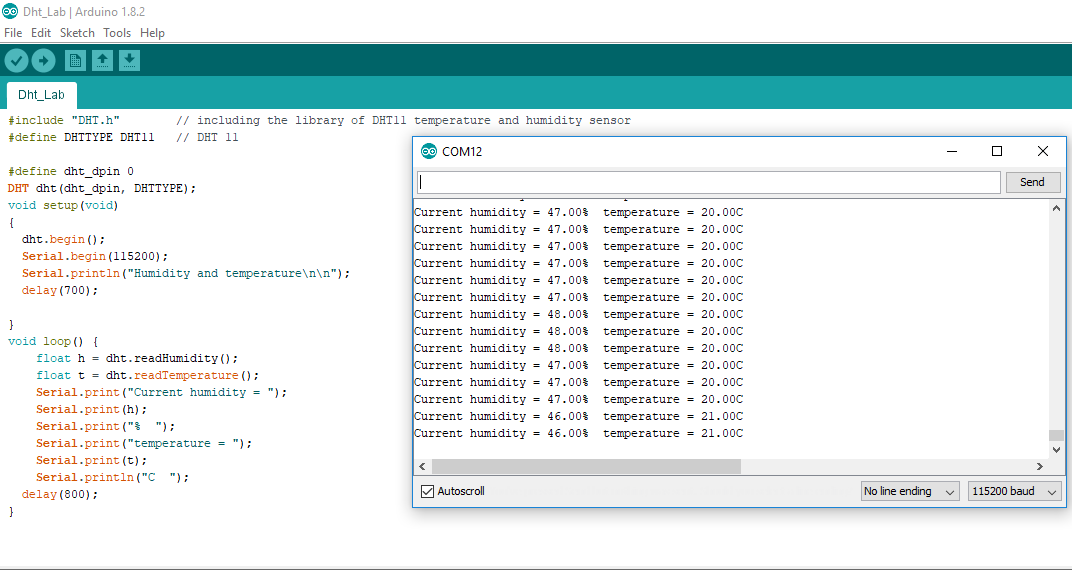
Serial.print(t);

Serial.println("C ");

delay(800);

}

**Output:**



Now we want to modify DHT11 example code to upload the value of Temperature and Humidity on a webpage. For this, youe need to open code named “DHT11Webpage” (Available on SLATE->Embedded Systems Resources -> Lab Manuals ->Lab 7 & 8). Check the code line by line and compile and upload it to your NodeMcu ESP8266 module.

#include "DHT.h" // including the library of DHT11 temperature and humidity sensor

#define DHTTYPE DHT11 // DHT 11

#define dht\_dpin 0

DHT dht(dht\_dpin, DHTTYPE);

void setup(void)

{

dht.begin();

Serial.begin(115200);

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);

}

|  |
| --- |
| **Output screenshot:** |

**Task:**

1. Modify DHT11 example code to upload the value of Temperature and Humidity to Adafruit io (or any other cloud).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Adafruit MQTT Library ESP8266 Example

Must use ESP8266 Arduino from:

https://github.com/esp8266/Arduino

Works great with Adafruit's Huzzah ESP board & Feather

----> https://www.adafruit.com/product/2471

----> https://www.adafruit.com/products/2821

Adafruit invests time and resources providing this open source code,

please support Adafruit and open-source hardware by purchasing

products from Adafruit!

Written by Tony DiCola for Adafruit Industries.

MIT license, all text above must be included in any redistribution

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <ESP8266WiFi.h>

#include "Adafruit\_MQTT.h"

#include "Adafruit\_MQTT\_Client.h"

#include "DHT.h"

#define DHTPIN D3 // what digital pin we're connected to NodeMCU (D6)

// Uncomment whatever type you're using!

#define DHTTYPE DHT11 // DHT 11

//#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321

//#define DHTTYPE DHT21 // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V

// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1

// to 3.3V instead of 5V!

// Connect pin 2 of the sensor to whatever your DHTPIN is

// Connect pin 4 (on the right) of the sensor to GROUND

// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.

// Note that older versions of this library took an optional third parameter to

// tweak the timings for faster processors. This parameter is no longer needed

// as the current DHT reading algorithm adjusts itself to work on faster procs.

DHT dht(DHTPIN, DHTTYPE);

char str\_hum[16];

char str\_temp[16];

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* WiFi Access Point \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define WLAN\_SSID "ghouri docking station"

#define WLAN\_PASS "logfirghouriaya"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Adafruit.io Setup \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define AIO\_SERVER "io.adafruit.com"

#define AIO\_SERVERPORT 1883 // use 8883 for SSL

#define AIO\_USERNAME "moiz23"

#define AIO\_KEY "5da51ad995d54d668c7ee9b85c61ca45"

/\*\*\*\*\*\*\*\*\*\*\*\* Global State (you don't need to change this!) \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Create an ESP8266 WiFiClient class to connect to the MQTT server.

WiFiClient client;

// or... use WiFiFlientSecure for SSL

//WiFiClientSecure client;

// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.

Adafruit\_MQTT\_Client mqtt(&client, AIO\_SERVER, AIO\_SERVERPORT, AIO\_USERNAME, AIO\_KEY);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Feeds \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>

Adafruit\_MQTT\_Publish temp = Adafruit\_MQTT\_Publish(&mqtt, AIO\_USERNAME "/feeds/temp");

Adafruit\_MQTT\_Publish hum = Adafruit\_MQTT\_Publish(&mqtt, AIO\_USERNAME "/feeds/hum");

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Sketch Code \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// Bug workaround for Arduino 1.6.6, it seems to need a function declaration

// for some reason (only affects ESP8266, likely an arduino-builder bug).

void MQTT\_connect();

void setup() {

Serial.begin(115200);

delay(10);

dht.begin();

Serial.println(F("Adafruit MQTT demo"));

// Connect to WiFi access point.

Serial.println(); Serial.println();

Serial.print("Connecting to ");

Serial.println(WLAN\_SSID);

WiFi.begin(WLAN\_SSID, WLAN\_PASS);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println();

Serial.println("WiFi connected");

Serial.println("IP address: "); Serial.println(WiFi.localIP());

}

void loop() {

// Ensure the connection to the MQTT server is alive (this will make the first

// connection and automatically reconnect when disconnected). See the MQTT\_connect

// function definition further below.

MQTT\_connect();

// Wait a few seconds between measurements.

delay(2000);

// Reading temperature or humidity takes about 250 milliseconds!

// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)

float h = dht.readHumidity();

// Read temperature as Celsius (the default)

float t = dht.readTemperature();

// Read temperature as Fahrenheit (isFahrenheit = true)

float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).

if (isnan(h) || isnan(t) || isnan(f)) {

Serial.println("Failed to read from DHT sensor!");

return;

}

// Compute heat index in Fahrenheit (the default)

float hif = dht.computeHeatIndex(f, h);

// Compute heat index in Celsius (isFahreheit = false)

float hic = dht.computeHeatIndex(t, h, false);

Serial.print("Humidity: ");

Serial.print(h);

Serial.print(" %\t");

Serial.print("Temperature: ");

Serial.print(t);

Serial.print(" \*C ");

Serial.print(f);

Serial.print(" \*F\t");

Serial.print("Heat index: ");

Serial.print(hic);

Serial.print(" \*C ");

Serial.print(hif);

Serial.println(" \*F");

Serial.print("Temperature in Celsius:");

Serial.println(String(t).c\_str());

Serial.print("Temperature in Fahrenheit:");

Serial.println(String(f).c\_str());

Serial.print("Humidity:");

Serial.println(String(h).c\_str());

//dtostrf(gps\_latitude, 4, 2, str\_temp);

//dtostrf(gps\_longitude, 4, 2, str\_hum);

// Now we can publish stuff!

Serial.print(F("\nSending Humidity value: "));

Serial.print(String(h).c\_str());

Serial.print("...");

if (! hum.publish(String(h).c\_str())) {

Serial.println(F("Failed"));

} else {

Serial.println(F("OK!"));

}

Serial.print(F("\nSending Temperature value: "));

Serial.print(String(t).c\_str());

Serial.print("...");

if (! temp.publish(String(t).c\_str())) {

Serial.println(F("Failed"));

} else {

Serial.println(F("OK!"));

}

// ping the server to keep the mqtt connection alive

// NOT required if you are publishing once every KEEPALIVE seconds

/\*

if(! mqtt.ping()) {

mqtt.disconnect();

}

\*/

}

// Function to connect and reconnect as necessary to the MQTT server.

// Should be called in the loop function and it will take care if connecting.

void MQTT\_connect() {

int8\_t ret;

// Stop if already connected.

if (mqtt.connected()) {

return;

}

Serial.print("Connecting to MQTT... ");

uint8\_t retries = 3;

while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected

Serial.println(mqtt.connectErrorString(ret));

Serial.println("Retrying MQTT connection in 5 seconds...");

mqtt.disconnect();

delay(5000); // wait 5 seconds

retries--;

if (retries == 0) {

// basically die and wait for WDT to reset me

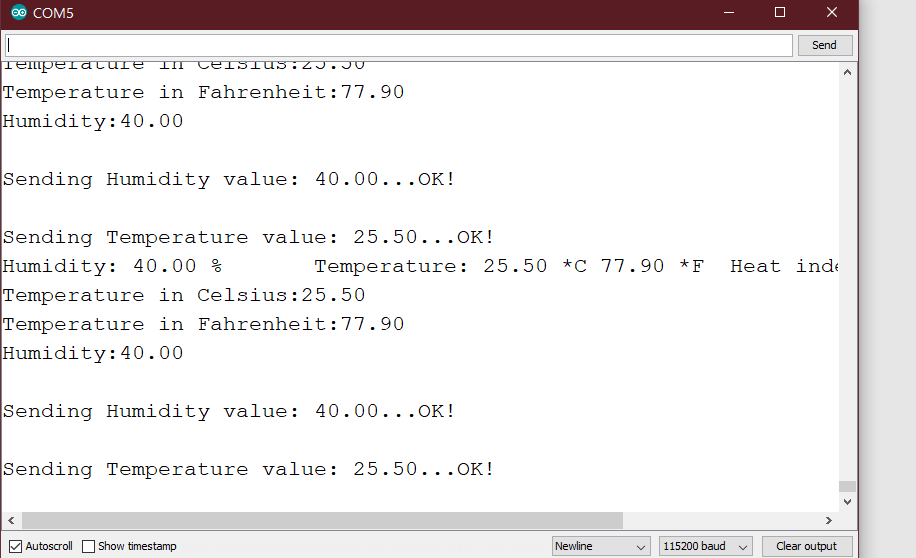
while (1);

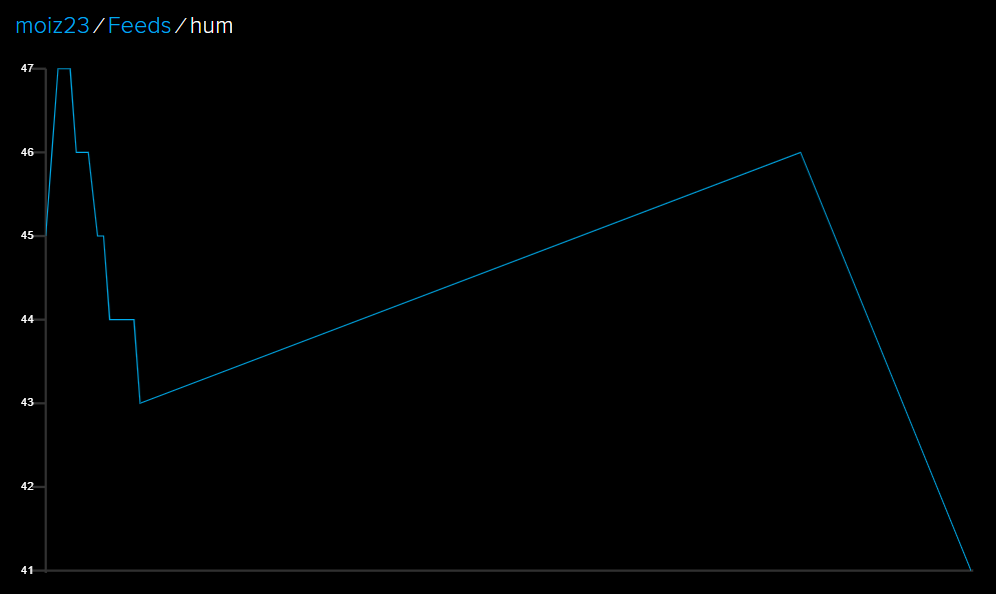
}

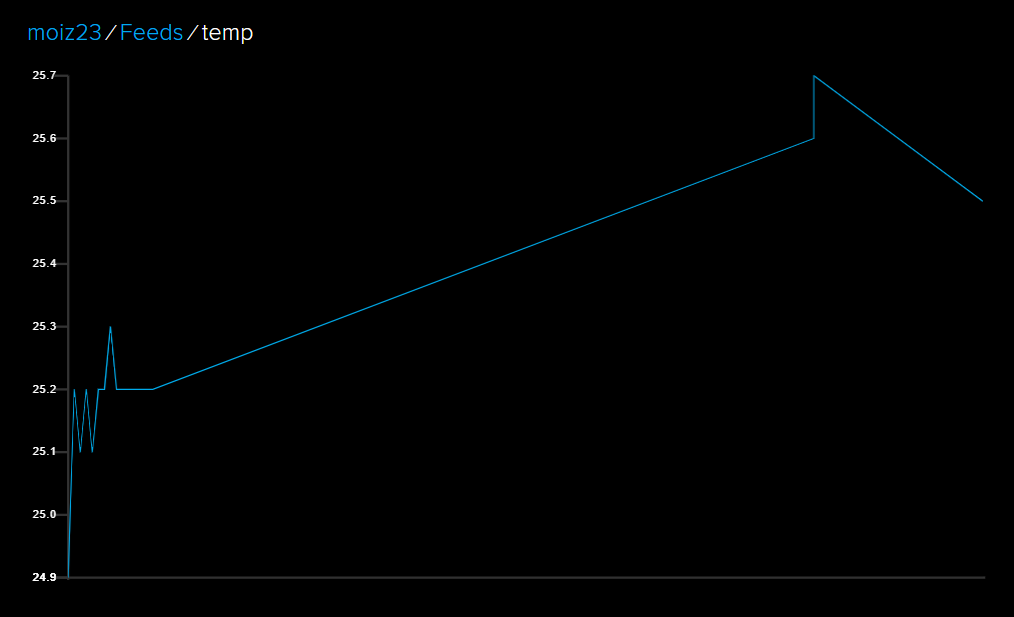
}

Serial.println("MQTT Connected!");

}

****

****

****