INTERNET OF THINGS PROJECT

IOT PROJECT ON TEMPERATURE AND HUMIDITY

YEAR:2020 -21

DONE BY: DHARSHINII RAVIKUMAR

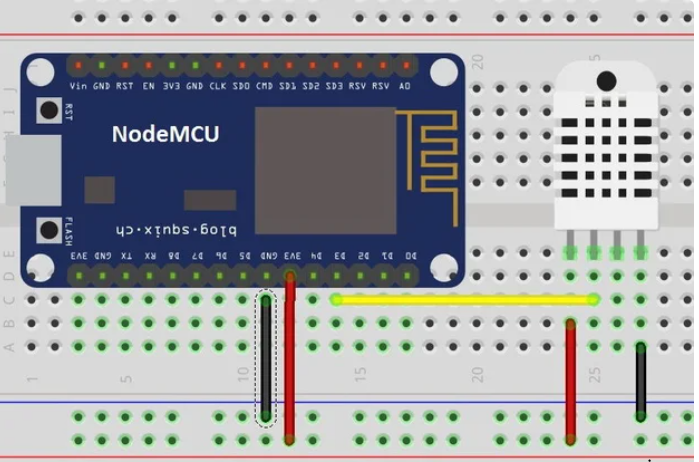
REFERRED FROM: <https://www.instructables.com/id/NodeMCU-IoT-Project-DHT11/>

IOT PROJECT ON TEMPERATURE AND HUMIDITY

REQUIREMENTS:

1. ESP8266 MODULE
2. DHT11 SENSOR
3. JUMPER WIRE
4. BREAD BOARD

CONNECTIONS:



PROGRAM CODE:

/\* \_\_\_ \_\_\_ \_\_\_ \_ \_ \_\_\_ \_\_\_ \_\_\_\_ \_\_\_ \_\_\_\_   
 \* / \_ \ /\_\_\_)/ \_ \| | | |/ \_ \ / \_ \ / \_\_\_) \_ \| \

\*| |\_| |\_\_\_ | |\_| | |\_| | |\_| | |\_| ( (\_\_| |\_| | | | |

\* \\_\_\_/(\_\_\_/ \\_\_\_/ \\_\_ |\\_\_\_/ \\_\_\_(\_)\_\_\_\_)\_\_\_/|\_|\_|\_|

\* (\_\_\_\_/

\* Use NodeMCU to drive DHT11 and send temperature/humidity value to MQTT server

\* Tutorial URL  [www.osoyoo.com](http://osoyoo.com/2016/11/24/use-nodemcu-to-send-temperaturehumidity-data-to-mqtt-iot-broker/)

\* CopyRight  [www.osoyoo.com](http://www.osoyoo.com/)

\*/

#include

#include

#include

dht DHT;

// Define NodeMCU D3 pin to as temperature data pin of DHT11

#define DHT11\_PIN D3

// Update these with values suitable for your network.

const char\* ssid = "\*\*\*\*\*\*";

const char\* password = "\*\*\*\*\*\*";

const char\* mqtt\_server = "broker.mqtt-dashboard.com";

//const char\* mqtt\_server = "iot.eclipse.org";

WiFiClient espClient;

PubSubClient client(espClient);

long lastMsg = 0;

char msg[50];

int value = 0;

void setup\_wifi() {

delay(100);

// We start by connecting to a WiFi network

Serial.print("Connecting to ");

Serial.println(ssid);

WiFi.begin(ssid, password);

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

{

delay(500);

Serial.print(".");

}

randomSeed(micros());

Serial.println("");

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

}

void callback(char\* topic, byte\* payload, unsigned int length)

{

Serial.print("Command is : [");

Serial.print(topic);

int p =(char)payload[0]-'0';

int chk = DHT.read11(DHT11\_PIN);

// if MQTT comes a 0 message, show humidity

if(p==0)

{

Serial.println("to show humidity!]");

Serial.print(" Humidity is: " );

Serial.print(DHT.humidity, 1);

Serial.println('%');

}

// if MQTT comes a 1 message, show temperature

if(p==1)

{

// digitalWrite(BUILTIN\_LED, HIGH);

Serial.println(" is to show temperature!] ");

int chk = DHT.read11(DHT11\_PIN);

Serial.print(" Temp is: " );

Serial.print(DHT.temperature, 1);

Serial.println(' C');

}

Serial.println();

} //end callback

void reconnect() {

// Loop until we're reconnected

while (!client.connected())

{

Serial.print("Attempting MQTT connection...");

// Create a random client ID

String clientId = "ESP8266Client-";

clientId += String(random(0xffff), HEX);

// Attempt to connect

//if you MQTT broker has clientID,username and password

//please change following line to if (client.connect(clientId,userName,passWord))

if (client.connect(clientId.c\_str()))

{

Serial.println("connected");

//once connected to MQTT broker, subscribe command if any

client.subscribe("OsoyooCommand");

} else {

Serial.print("failed, rc=");

Serial.print(client.state());

Serial.println(" try again in 5 seconds");

// Wait 6 seconds before retrying

delay(6000);

}

}

} //end reconnect()

void setup() {

Serial.begin(115200);

setup\_wifi();

client.setServer(mqtt\_server, 1883);

client.setCallback(callback);

int chk = DHT.read11(DHT11\_PIN);

Serial.print(" Starting Humidity: " );

Serial.print(DHT.humidity, 1);

Serial.println('%');

Serial.print(" Starting Temparature ");

Serial.print(DHT.temperature, 1);

Serial.println('C');

}

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

long now = millis();

// read DHT11 sensor every 6 seconds

if (now - lastMsg > 6000) {

lastMsg = now;

int chk = DHT.read11(DHT11\_PIN);

String msg="real time temperature: ";

msg= msg+ DHT.temperature;

msg = msg+" C ;real time Humidity: " ;

msg=msg+DHT.humidity ;

msg=msg+"%";

char message[58];

msg.toCharArray(message,58);

Serial.println(message);

//publish sensor data to MQTT broker

client.publish("OsoyooData", message);

}

}

OUTPUT:

