

ASSIGNMENT NO:

AIM:

Upload data from environmental sensor to cloud server (You can use any public cloud IBM Watson IoT cloud or Google or AWS etc.).

OBJECTIVE:

- To understand functionalities of various single board embedded platform fundamentals.
- To develop a temperature sensing application and sending this data to Ubidots.

THEORY:

Ubidots is an open data platform for monitoring your data online. You can set the data as private or public depending on your choice. Ubidots takes minimum of 15 seconds to update your readings. Its a great platform for building your IOT projects.

We will read the temperature and humidity from the DHT22 and then we will send it to the API of the Ubidots channel. We will get the API after creating the channel.

Temperature sensor:

It is a device ,a thermocouple or RTD,that provides temperature measurement through an electrical signal.

Thermocouple:

It is made from two dissimilar metals that generate electrical voltage in direct proportion to changes in temperature.The wires are joined together to form measuring junction and reference junction.

RTD:

Resistor temperature detection is a variable resistor that will change its electrical resistance in direct proportion to changes in temperature in a precise,repeatable and linear manner.

Setting up the Ubidots Account:

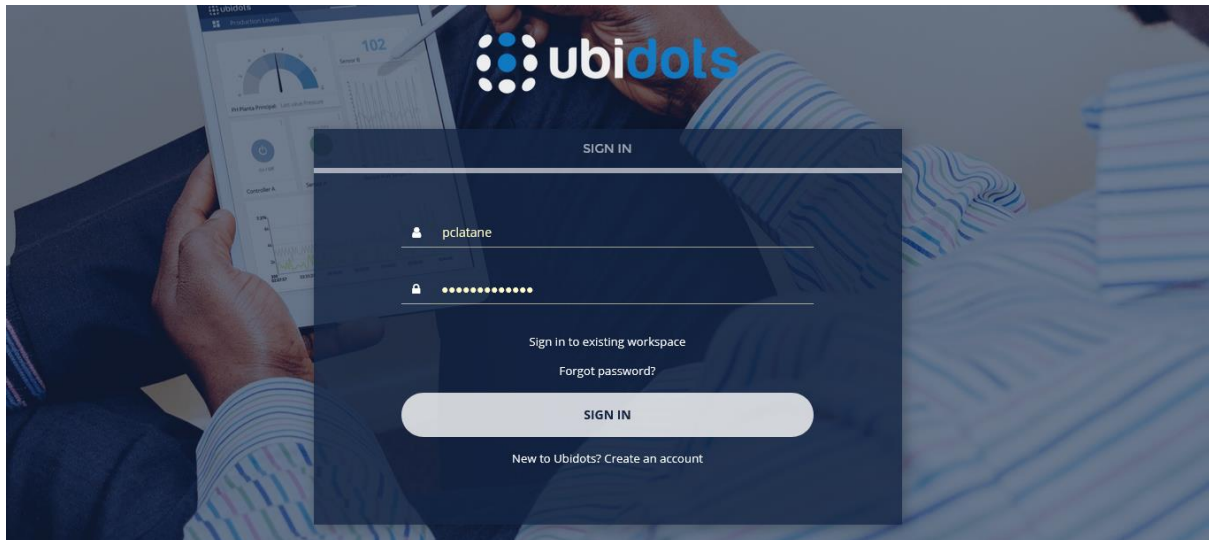
Whats exactly Ubidots?

“**Ubidots** is an open source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. It enables the creation of sensor logging applications, location tracking applications, and a social network of things with status updates”. Ubidots is an Internet

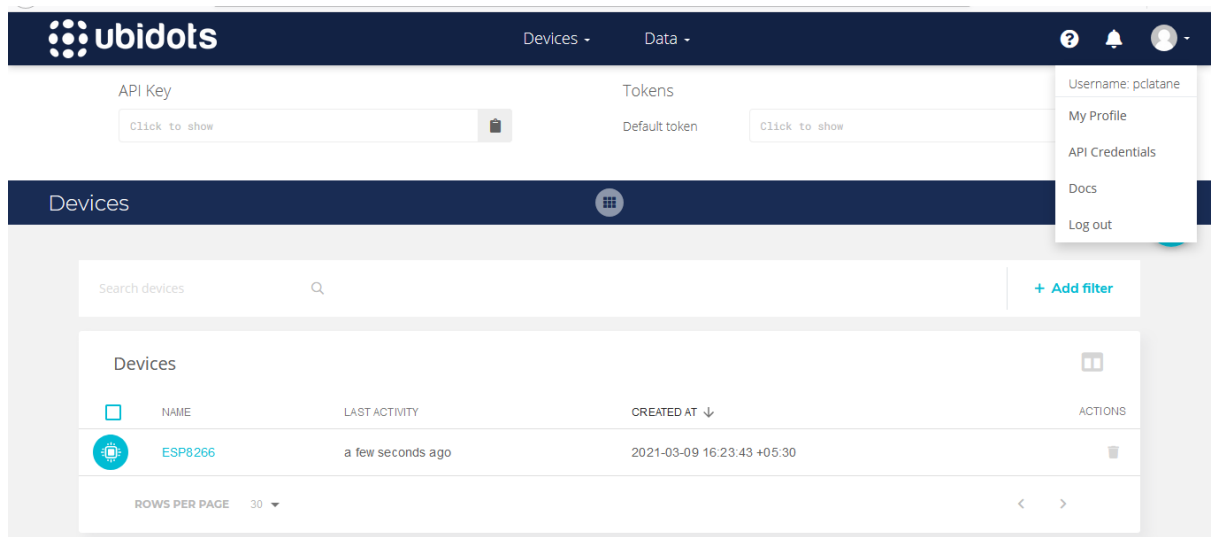
of Things (IoT) platform that lets you collect and store sensor data in the cloud and develop IoT applications. Ubidots IoT platform provides apps that let you analyze and visualize your data in MATLAB, and then act on the data. The Sensor data sent to Ubidots from Arduino, Raspberry Pi, BeagleBone Black, and other hardware.

Create a new project on Ubidots for Education-

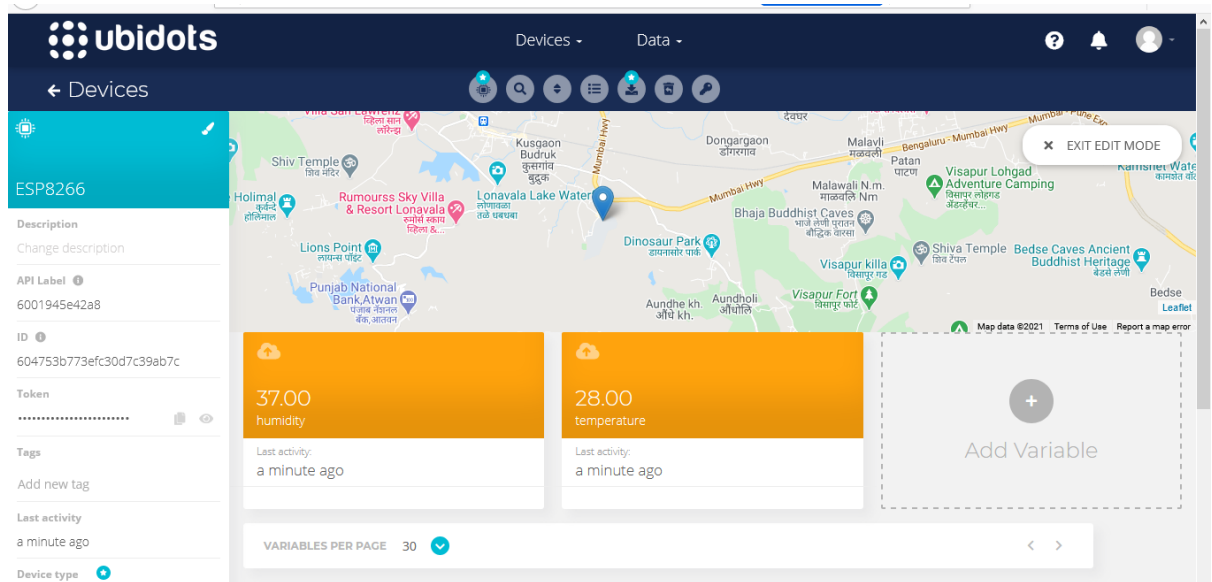
1. First of all, go to the following link and sign up to Ubidots for Education. If you already have an account, then sign in <https://stem.ubidots.com/accounts/signin/>



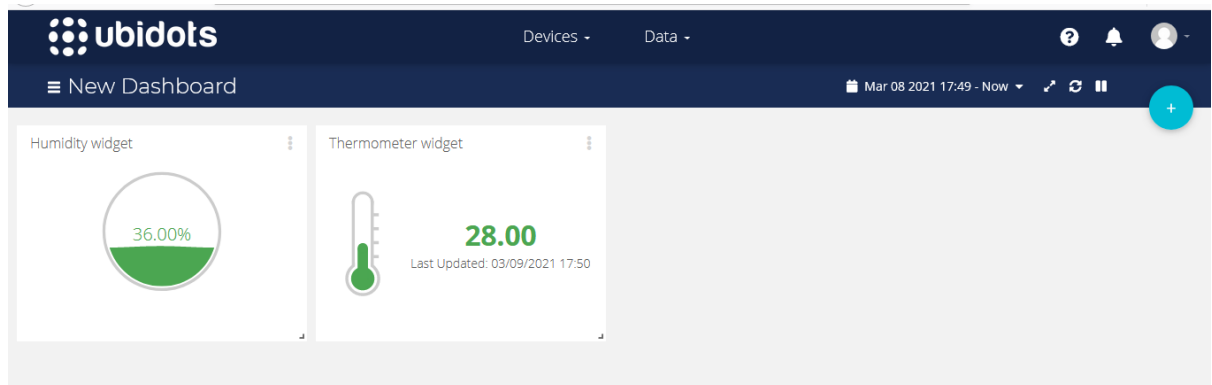
2. After creating the account or logging in, you will be see a page like shown below. In “Devices” goto “API Credentials” to get token from “Show default token” which is tobe ued in program



3. In selected device “Add variable”



4. Now in Data goto Dashboards to add widget



5. Now goto Explore data to observe reading graphically



COMPONENTS REQUIRED:

- ESP8266 (NodeMCU)
- DHT11
- Breadboard
- Connecting Wires

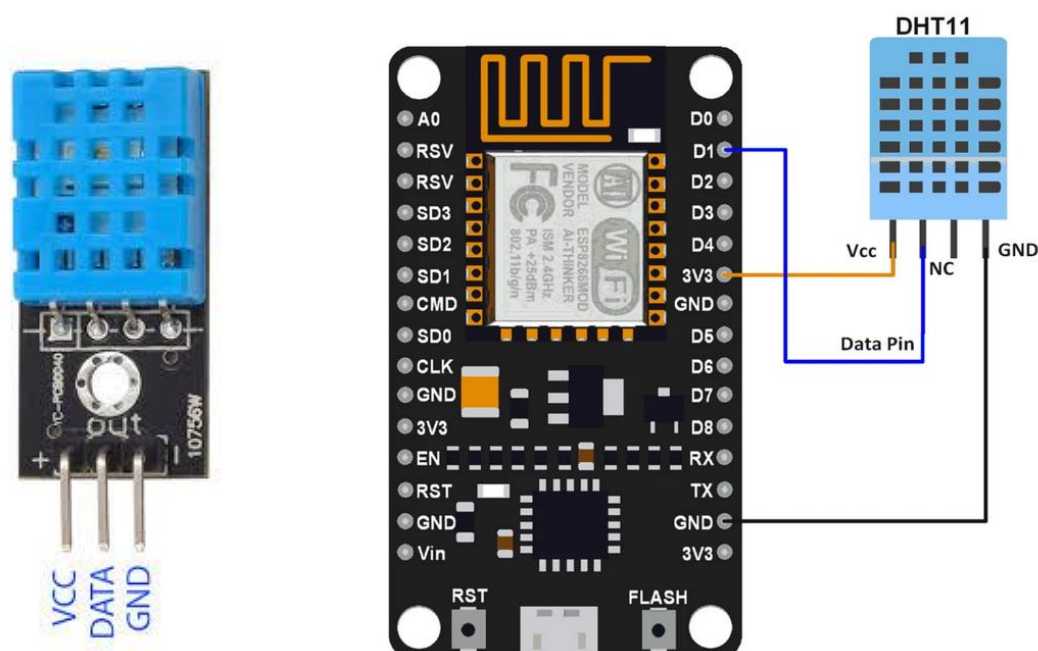
Circuit Diagram

In this experiment, we are going to log and monitor temperature data over internet using Ubidots IoT server. And we can view the logged data and graph over time in their website. It is made using ESP8266 WiFi module and DHT11 humidity and temperature sensor. ESP8266 WiFi chip reads the current humidity and temperature from DHT11 using ADC and sends it to Ubidots server for live monitoring from anywhere in the world.

In this experiment, **DHT11 humidity and temperature sensor** can be used with esp8266 or Nodemcu and program from Arduino IDE, the DHT11 has used to send data to Ubidots and it was really productive there is no sudden spike in temperature change, it was gradual and results are quite good.” Same temperature output can be observed on serial monitor of arduino IDE.

Connection:

- Connect 5V pin of ESP8266 to first pin of DHT11 (+)
- Connect D0 pin of ESP8266 to second pin of DHT11 (out) in series with 1k resistor
- Connect GND pin of ESP8266 to third pin of DHT11 (-)



Program:

```
#include <ESP8266WiFi.h>
#include "dht.h"
#include "UbidotsMicroESP8266.h"
#define TOKEN "BBFF-xqLJHkOXPjhTY0iUGRxLXyWhejABgk" // Put here
your Ubidots TOKEN
#define WIFISSID "OPPO_pcl"
#define PASSWORD "pclatane123"
#define dht_apin D0
dht DHT;
Ubidots client(TOKEN);

void wifi()
{
    Serial.print("connecting");
    while (WiFi.status() != WL_CONNECTED)
    {
        Serial.print(".");
        delay(500);
        break;
    }
    Serial.println();
    Serial.print("connected: ");
    Serial.println(WiFi.localIP());
    client.wifiConnection(WIFISSID, PASSWORD);}
float resolution=3.3/1023;
void setup(){
    Serial.begin(9600);
}
void loop()
{
    int readdata = DHT.read11(dht_apin);
    float t = DHT.temperature;
    float h = DHT.humidity;
```

```
delay(1000);
Serial.print("Current humidity = ");
Serial.print(h);
// P.print("Humid=");
delay(1000);
Serial.print("% ");
Serial.print("temperature = ");
Serial.print(t);
Serial.println("C ");

client.add("temperature", t);
client.add("humidity", h);
// client.add("intensity", ldr);
delay(1000);
client.sendAll(true);
delay(1000);
}
```

CONCLUSION:

After the study of this assignment we are familiar with the Ubidots platform and how to send temperature and humidity data to Ubidots cloud.