

ASSIGNMENT NO: 02**TITLE:**

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

AIM:

To Design and implement parameter monitoring IoT system keeping records on Cloud such as 'environment humidity and temperature monitoring'.

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:

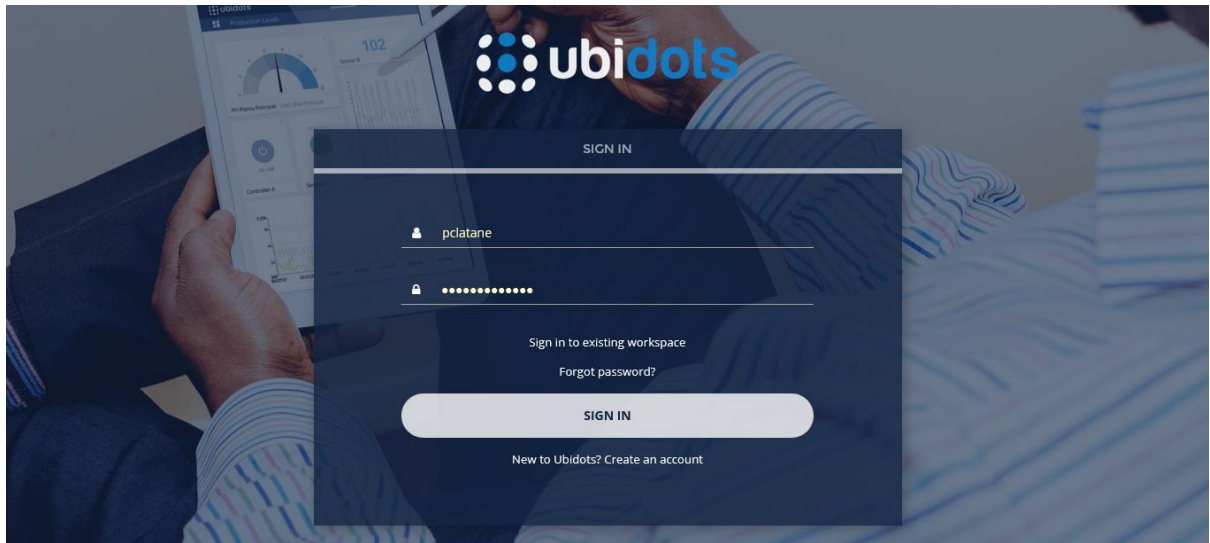
Resistor temperature detection is a variable resistor that will change its electrical resistance in direct proportion to changes in temperature in a precise,repeatable and linear manner.

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.

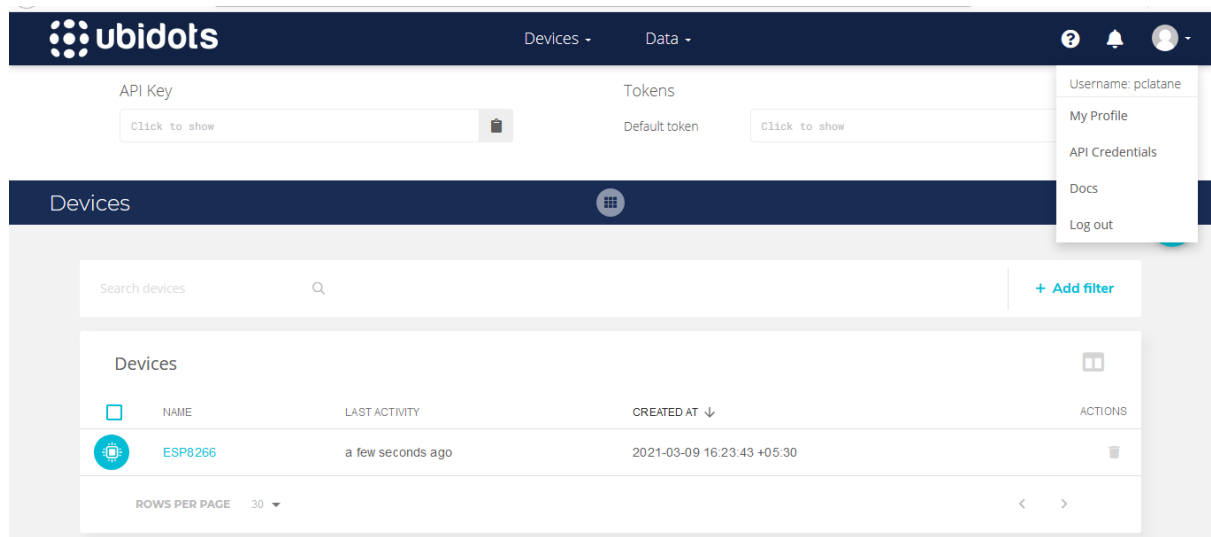
Create a new project on Ubidots for Education-

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 to be used in program



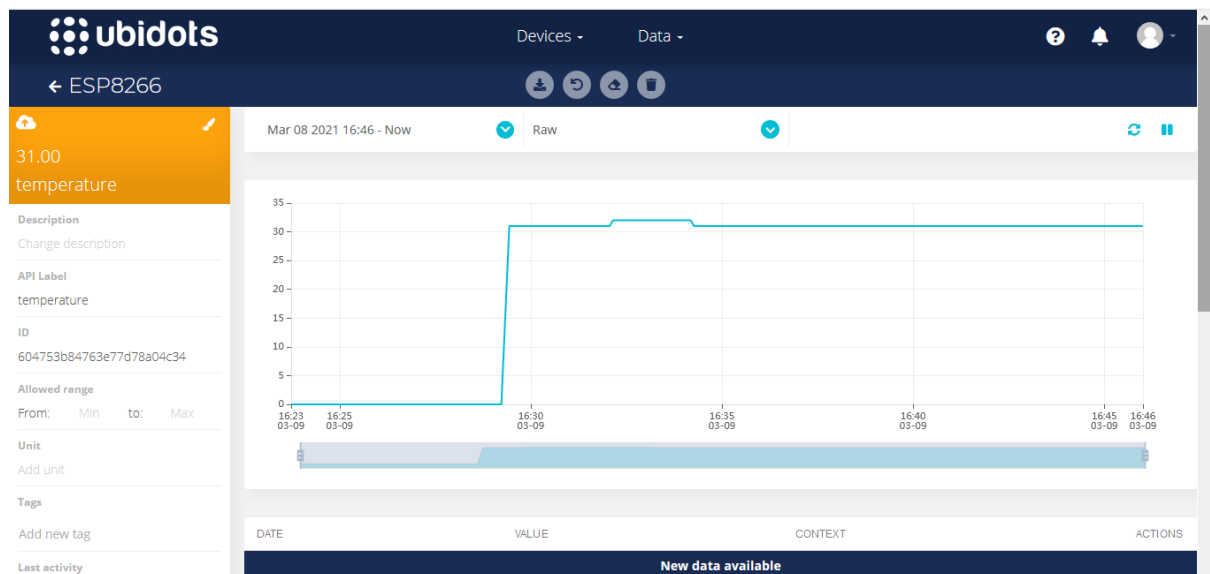
3. In selected device "Add variable"

The screenshot shows the Ubidots interface for a selected device, ESP8266. On the left, there's a sidebar with device details: Description, API Label (6001945e42a8), ID (604753b773efc30d7c39ab7c), Token, Tags, and Last activity (a minute ago). The main area displays a map of Mumbai with a location pin. Below the map, there are two orange boxes showing current values: 37.00 humidity and 28.00 temperature, both with a 'Last activity: a minute ago' label. To the right of these boxes is a large dashed box with a plus sign and the text 'Add Variable'. At the bottom, there's a 'VARIABLES PER PAGE' dropdown set to 30.

4. Now in Data goto Dashboards to add widget

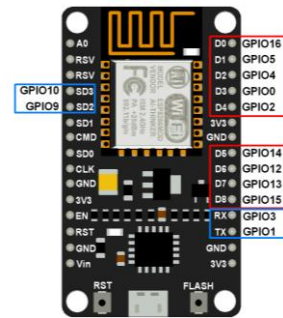
The screenshot shows the Ubidots 'New Dashboard' interface. The top bar includes the Ubidots logo, navigation tabs for 'Devices' and 'Data', and a date range selector set to 'Mar 08 2021 17:49 - Now'. Below the top bar, there are two widgets: a 'Humidity widget' showing a circular gauge at 36.00% and a 'Thermometer widget' showing a thermometer icon with the value 28.00 and 'Last Updated: 03/09/2021 17:50'. A blue plus icon in the top right corner indicates the option to add more widgets.

5. Now goto Explore data to observe reading graphically



COMPONENTS REQUIRED:

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

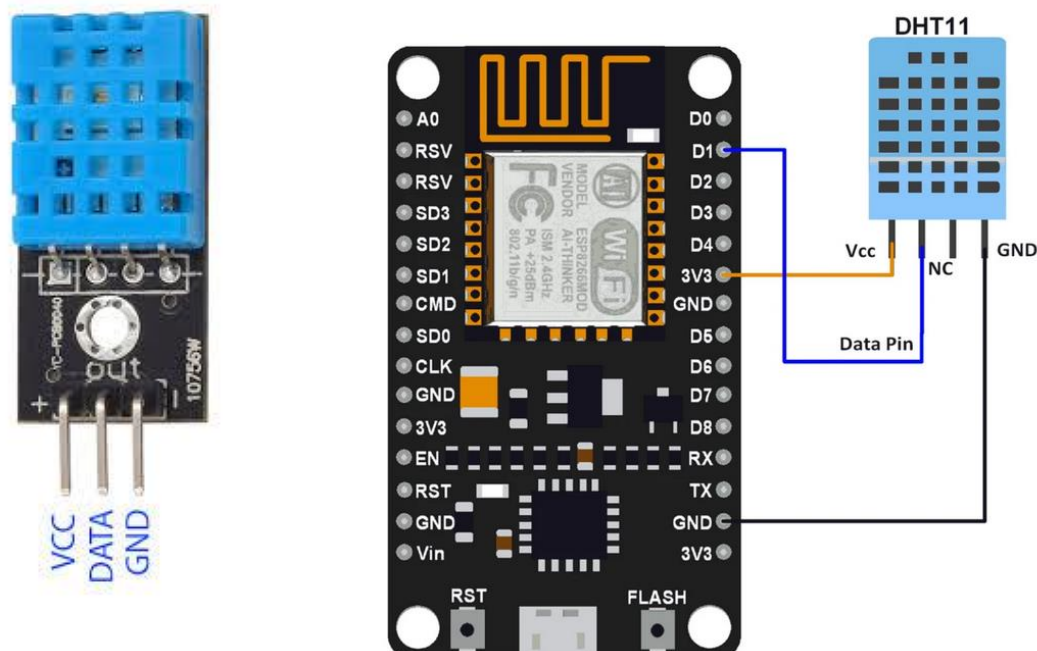
**Circuit Diagram**

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 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:

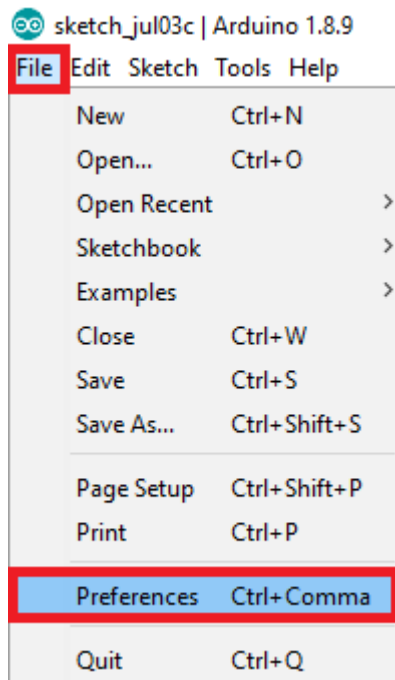
- Connect 5V pin of ESP8266 to first pin of DHT11 (+)
- Connect D0 pin of ESP8266 to second pin of DHT11 (out) in series with 1k resistor
- Connect GND pin of ESP8266 to third pin of DHT11 (-)



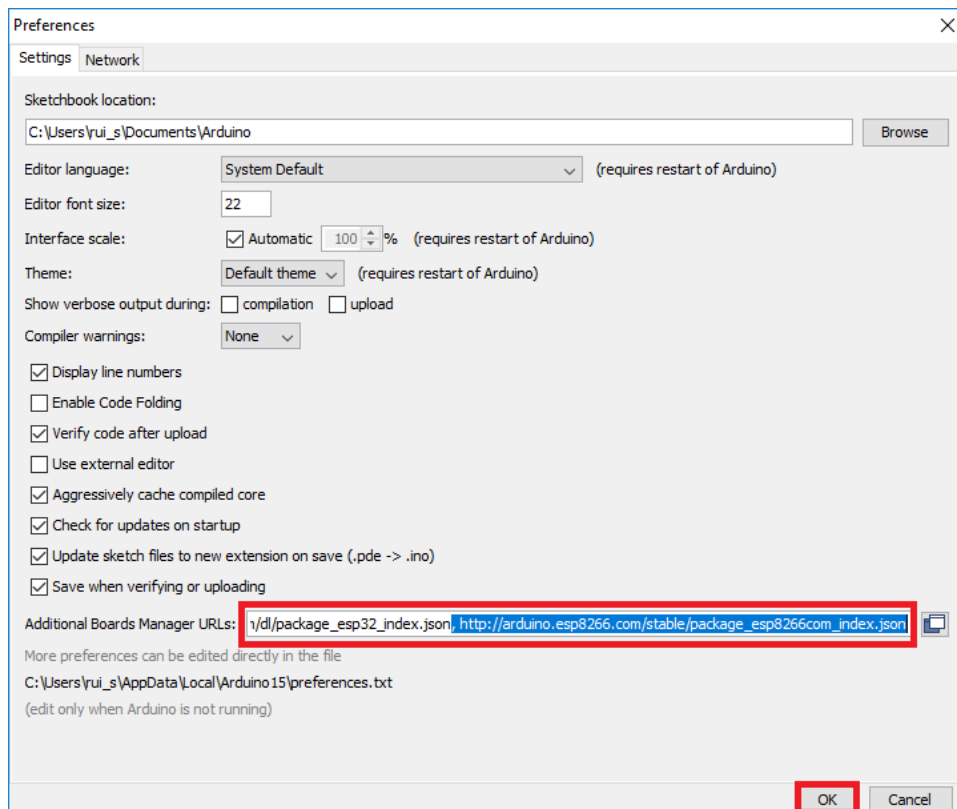
How to Install ESP8266 Add-on in Arduino IDE

To install the ESP8266 board in your Arduino IDE, follow these next instructions:

1. In your Arduino IDE, go to **File> Preferences**



2. Enter http://arduino.esp8266.com/stable/package_esp8266com_index.json into the "Additional Boards Manager URLs" field as shown in the figure below. Then, click the "OK" button:



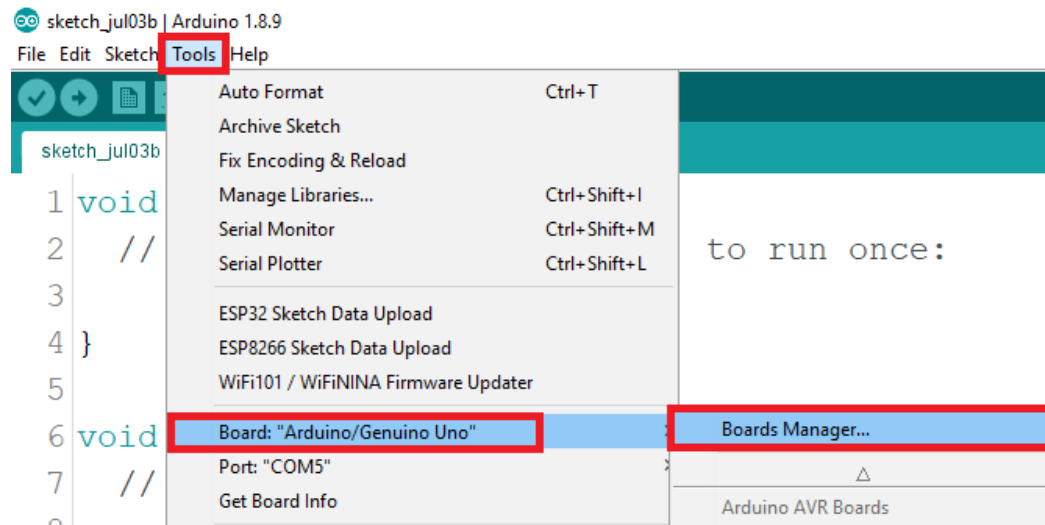
Note: if you already have the ESP32 boards URL, you can separate the URLs with a

comma as follows:

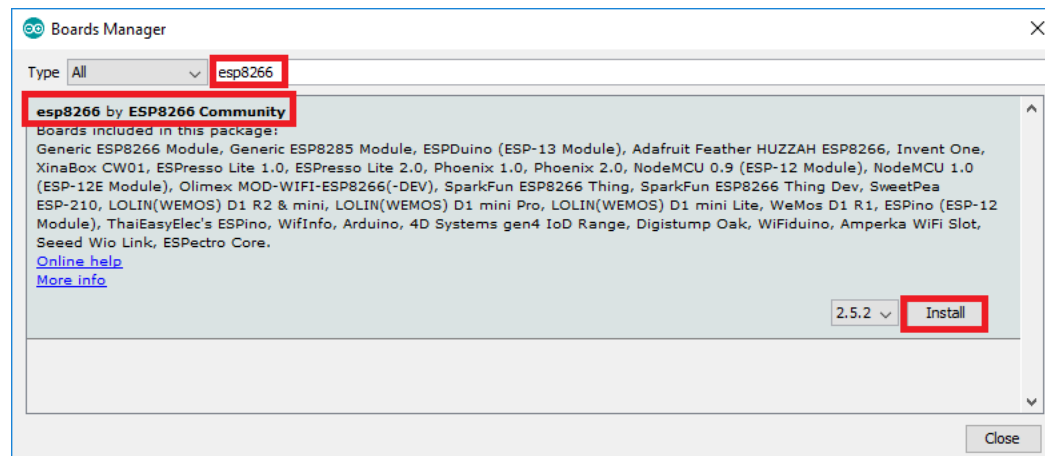
https://dl.espressif.com/dl/package_esp32_index.json,

http://arduino.esp8266.com/stable/package_esp8266com_index.json

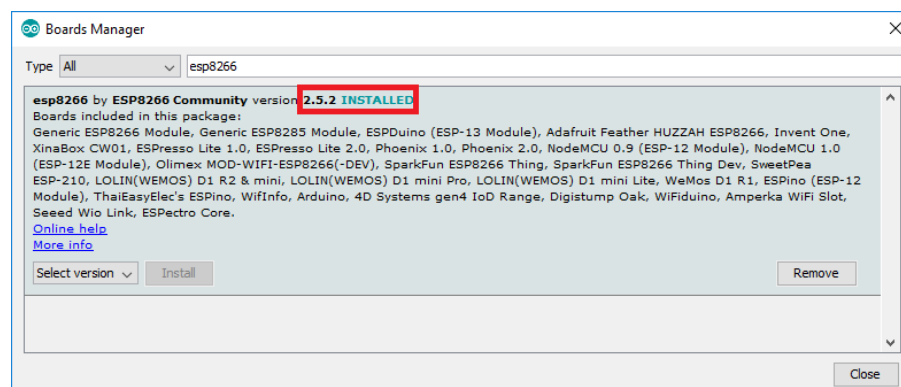
3. Open the Boards Manager. Go to **Tools > Board > Boards Manager...**



4. Search for **ESP8266** and press install button for the “**ESP8266 by ESP8266 Community**”:



5. That's it. It should be installed after a few seconds.



<https://randomnerdtutorials.com/how-to-install-esp8266-board-arduino-ide/>

Program:

```
#include <ESP8266WiFi.h>
#include <DHT.h>
#include "Ubidots.h"
#define TOKEN "BBFF-xqLJHkOXPjhTY0iUGRxLXyWhejABgk" // Put here
your Ubidots TOKEN
#define WIFISSID "OPPO_pcl"
#define PASSWORD "pclatane123"
#define DHTPIN 0 // Digital pin we're connected to D3 on NodeMcu

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

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.wifiConnect(WIFISSID, PASSWORD);}

void setup(){
    Serial.begin(9600);
    dht.begin();
}
```

```
void loop()
{
    float h = dht.readHumidity();
    float t = dht.readTemperature(); // or dht.readTemperature(true) for
    Fahrenheit
    delay(1000);
    Serial.print("Current humidity = ");
    Serial.print(h);
    delay(1000);
    Serial.print("% ");
    Serial.print("temperature = ");
    Serial.print(t);
    Serial.println("C ");

    client.add("temperature", t);
    client.add("humidity", h);
    delay(1000);
    client.send();
    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.