

## Internet

### Milestone 4

**Deadline:** Thursday 28.11.2018 @ 11:59 P.M.

## 1 Introduction

Internet of Things (IoT) strives to connect devices remotely for seamless functioning and ease of operations. An IoT platform bridges the gap between device sensors and data network. It provides an insight into the data used in backend application. This platform is a set of components that allows developers to spread out the applications, remotely collect data, secure connectivity, and execute sensor management. In addition, it manages connectivity of the devices and allows developers to build new mobile software applications. An IoT platform facilitates the collection of data from devices and enables business transformation. It connects different components, ensuring uninterrupted flow of communication between the devices.

In the previous milestones, every team built a simple IoT application. The application was simply used for monitoring the temperature values for a specific area to be stored to a server. The application was consist of a Java Server and ESP8266 Wi-Fi modules connected with a temperature sensor. The temperature values were stored to the Java Server. Also, the Java Server works only in local networks.

Assuming your application has hundreds of IoT sensor modules such as ESP8266 Wi-Fi modules massive volume of data are generated. These data needs to be monitored in good interface. The data should be accessed anywhere and anytime.

## 2 Requirements

### 2.1 Milestone 4 Requirements

The main objective of this milestone is Building a complete IoT system. In this milestone, you are going to connect your ESP module to an IoT platform. You are going to use [Ubidots IoT platform](#) to replace your Java Server.

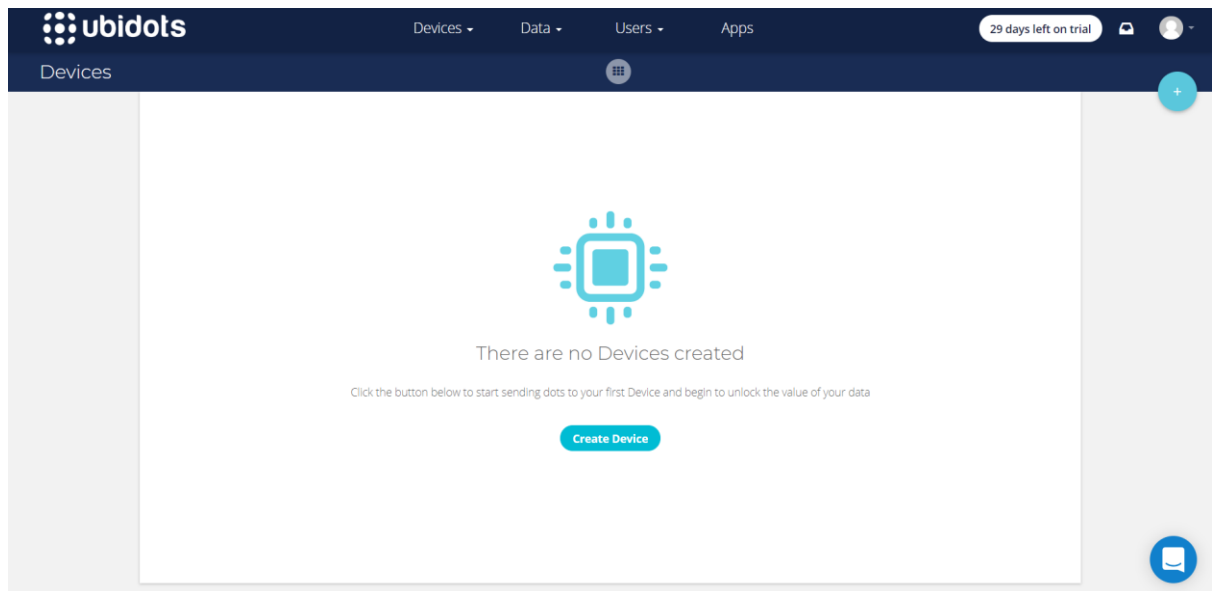
**In this milestone, you are required to deliver the following:**

1. Signup for an account in the Ubidots IoT platform. Link: <https://ubidots.com/>
2. Send the readings obtained by ESP8266 to Ubidots IoT platform every constant time interval. (30 seconds for example).
3. Ubidots IoT platform should monitor the temperature values send from the ESP module.

## 2.2 Connect ESP8266 with Ubidots IoT Platform

You are required to follow the steps below:

1. Signup for an account in the Ubidots IoT platform. Link: <https://ubidots.com/>
2. After Signup, you should **click on Devices Tab** the following window will appears.



3. Click on **create device** -> **wifi** -> **Espressif Systems** -> choose **NodeMCU ESP8266** -> **HTTP**, after this steps [new tab](#) will be opened
4. The [new tab](#) shows a complete guide how to connect your ESP to Ubidots using HTTP.
5. In the Requirements section you need only to download [UbidotsMicroESP8266 library](#) the other requirements already done in Milestone 2
6. Importing [UbidotsMicroESP8266 library](#) to Arduino IDE
  - a. click on **Sketch** -> **Include Library** -> **Add .ZIP Library**.
  - b. Select the .ZIP file of UbidotsMicroESP8266 and then "Accept" or "Choose".
  - c. Restart your Arduino IDE
7. Follow the steps in section **2. Sending (POST) Data to Ubidots** and ignore section 1. Setting up Arduino IDE because it is also done in Milestone 2
8. Update your code with your **access point SSID**, **password** and your **Ubidots Token**

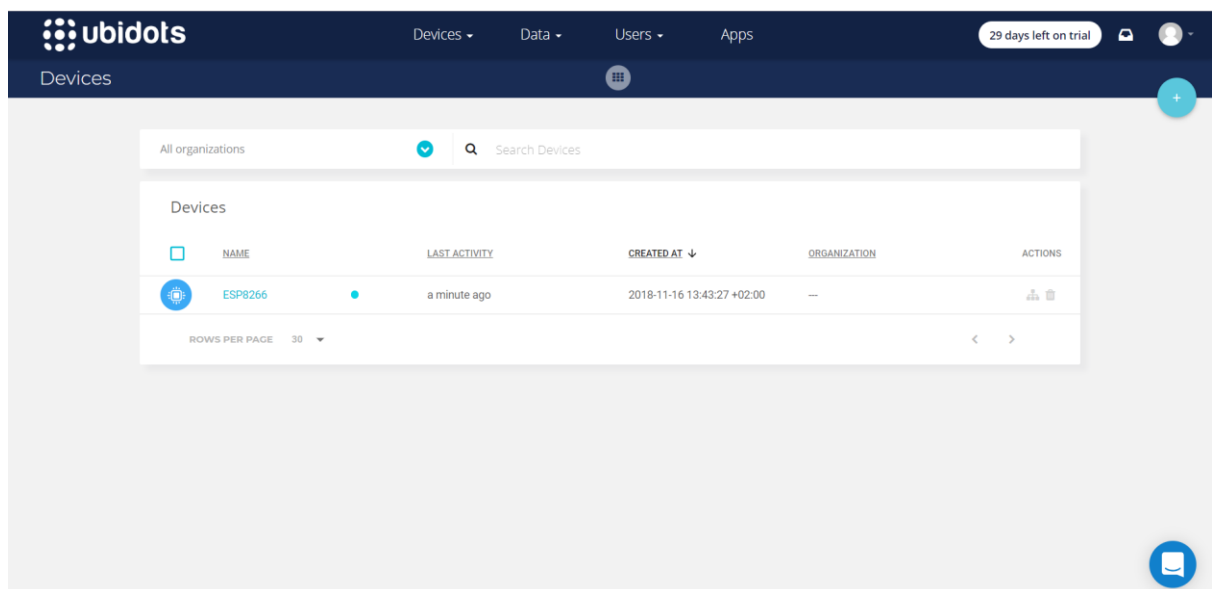
```
#define TOKEN "Your_token_here" // Put here your Ubidots TOKEN
#define WIFISSID "Your_WiFi_SSID" // Put here your Wi-Fi SSID
#define PASSWORD "Your_WiFi_Password" // Put here your Wi-Fi password
```

9. Check what you done in step 7 to know how to obtain the **Ubidots Token**

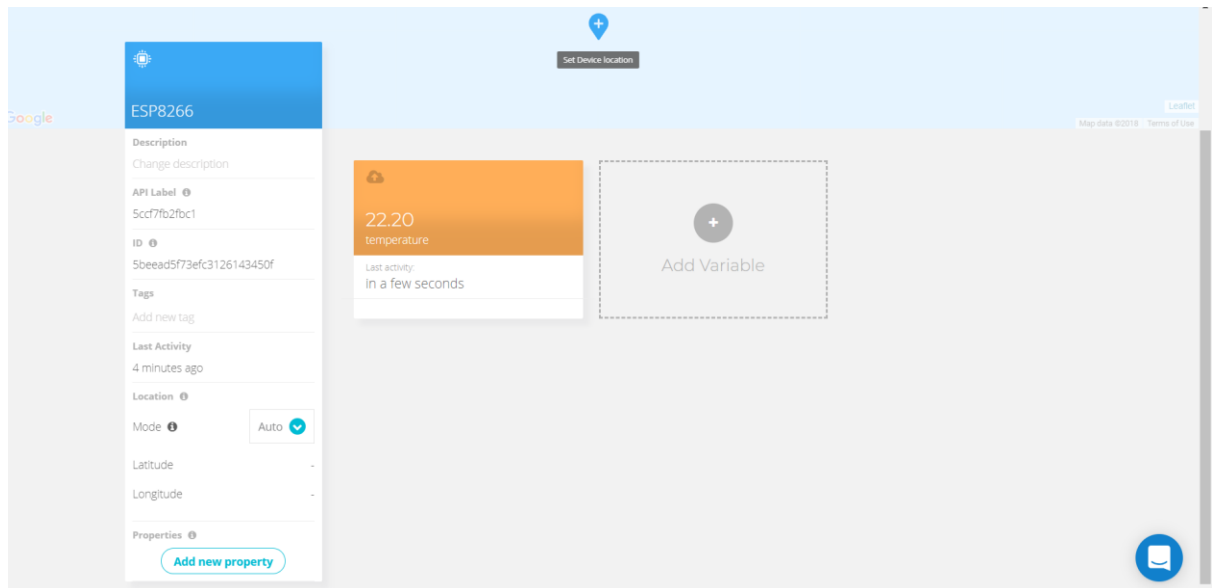
10. You should **update** the **loop method** obtained from step 7 **with the temperature reading code** in milestone 3.

```
void loop() {  
  
    int analogValue = analogRead(A0);  
    float millivolts = (analogValue/1024.0) * 3300;  
    float celsius = millivolts/100;  
    client.add("temperature", celsius);  
    client.sendAll(true);  
    delay(5000);  
}
```

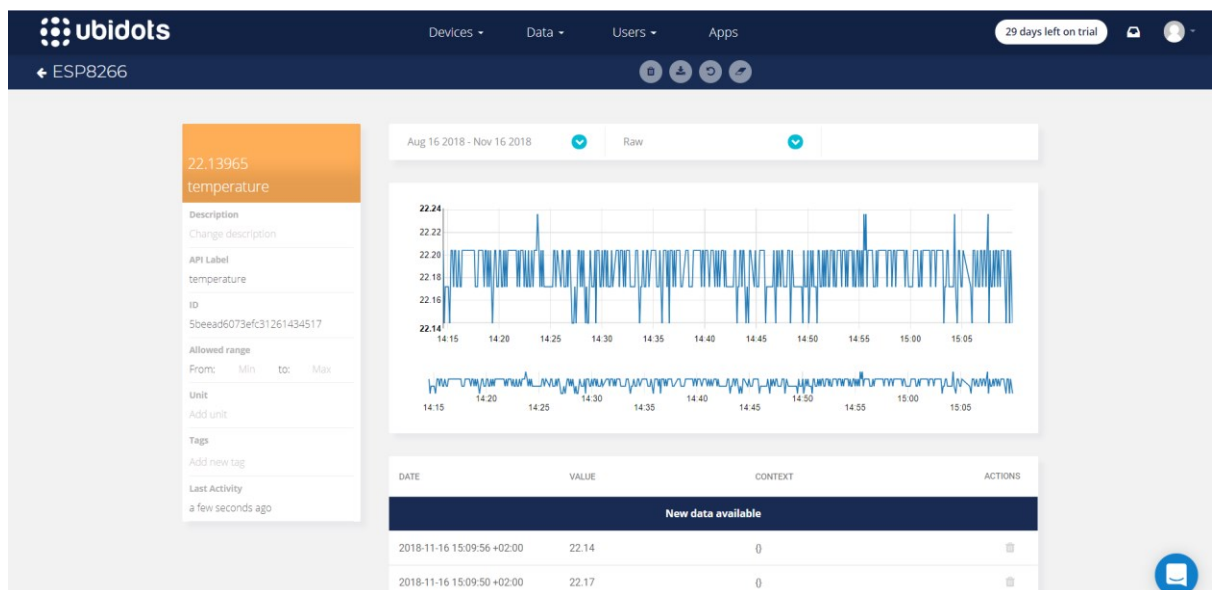
11. Upload the code to your ESP Module. (Note that you will find an error if you do not import the Ubidots library correctly)
12. After the code uploaded successfully, you should open your serial monitor to check that everything is running correctly.
13. You will find your device appeared on the Ubidots Device tab as shown below



14. Click on your device you will find the temperature value and other parameters (such as device location tag add additional variables



15. Click on the temperature value you will monitor the temperature reading values in a graph and the update time.



### 3 Submission

Milestone 4 is a team task. Each team should has **at least 2 members and Maximum 4 members**. Any cheating will be graded **ZERO**.

**You have to submit the:**

- 1- Take screenshots for the
  - a. Serial Monitor in Arduino
  - b. Ubidots Server
- 2- Arduino code + Screenshots as zipped file.

**Note:** *Make sure that the temperature readings in the serial monitor are the same on the Ubidots server.*

The Deadline for submitting **milestone 4** on Thursday 28/11 @ 11:59 P.M.

You should submit your milestone 4 using the following google form via link:

[https://docs.google.com/forms/d/e/1FAIpQLScAbM8aqo3kJxMKJl63\\_J0bVNbBhY3\\_8z4WTOKXDKPh-xLSJA/viewform?usp=pp\\_url](https://docs.google.com/forms/d/e/1FAIpQLScAbM8aqo3kJxMKJl63_J0bVNbBhY3_8z4WTOKXDKPh-xLSJA/viewform?usp=pp_url)