

SWE30011

IoT Programming

Project Instruction- Part 2

By: Armita Zarnegar

SWIN
BUR
* NE *

SWINBURNE
UNIVERSITY OF
TECHNOLOGY

Aims:

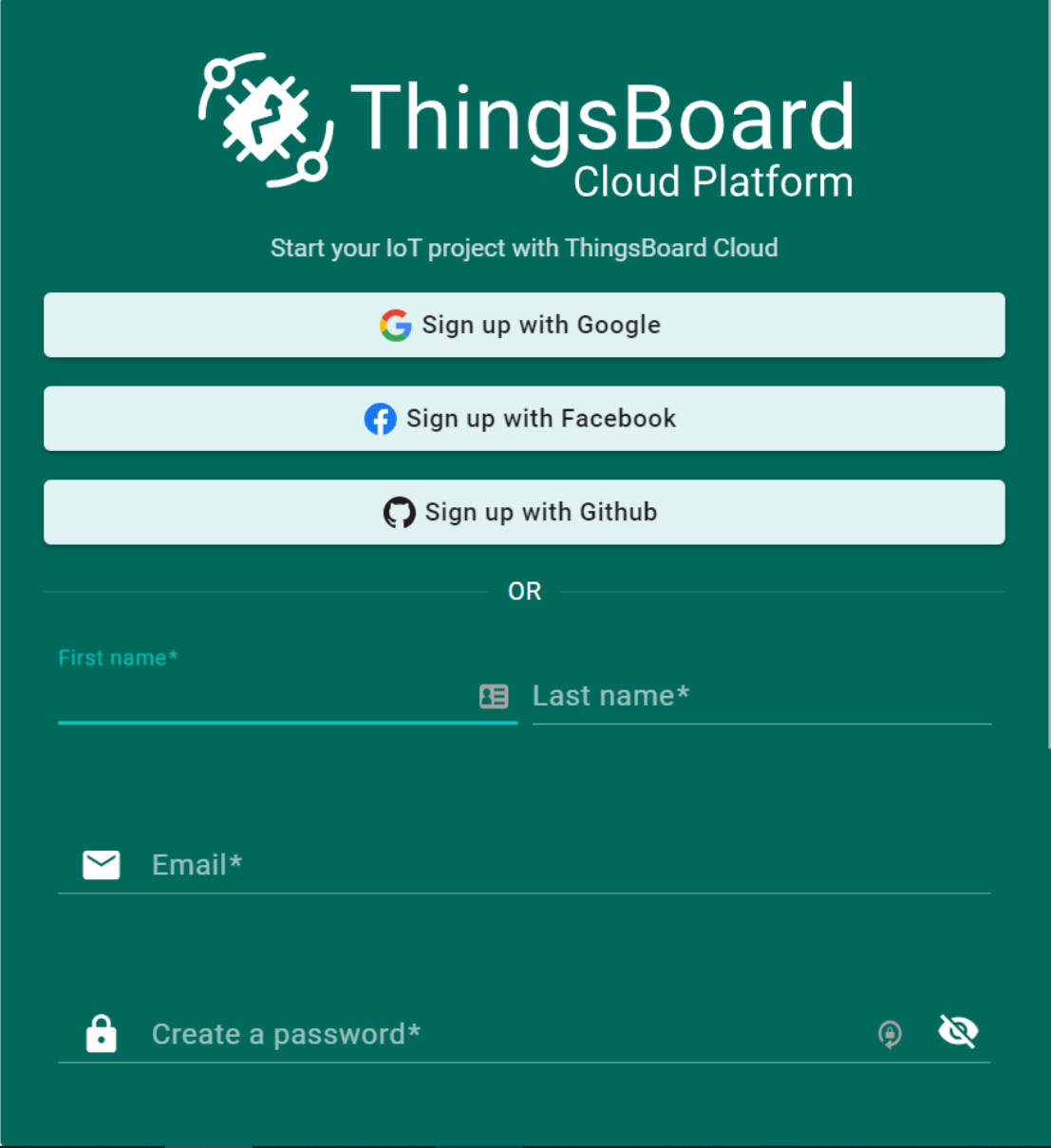
To connect and send data from Arduino to cloud (Thingsboard) via a Raspberry Pi VM.

Requirements and Preliminary Settings:

- Raspberry Pi Virtual Machine with Debian 11,
- Arduino Uno and its IDE ,
- Python version 3 or above installed on the Virtual Machine (refer to the previous assignment's document)
- A given code to run in Arduino IDE
- A given Python script to run in Raspberry Pi VM

Creating a Thingsboard account and device


Visit <https://thingsboard.io/>. From the menu choose "Try It Now" then "Start Free"





The image shows the ThingsBoard Cloud Platform sign-up and login interface. It features a dark teal background with the ThingsBoard logo and name at the top. Below the logo, there is a prompt to start an IoT project. Three light blue buttons offer sign-up options: Google, Facebook, and Github. An 'OR' separator is followed by input fields for 'First name*', 'Last name*', 'Email*', and 'Create a password*'. The password field includes icons for showing and hiding the password.

ThingsBoard Cloud Platform

Start your IoT project with ThingsBoard Cloud


 Sign up with Google


 Sign up with Facebook


 Sign up with Github



OR

First name*

 Last name*

 Email*

 Create a password*

Use your Gmail account or a an email along with a password to sign up. You will need to confirm your email address. After successfully logging in, you will be directed to this dashboard.

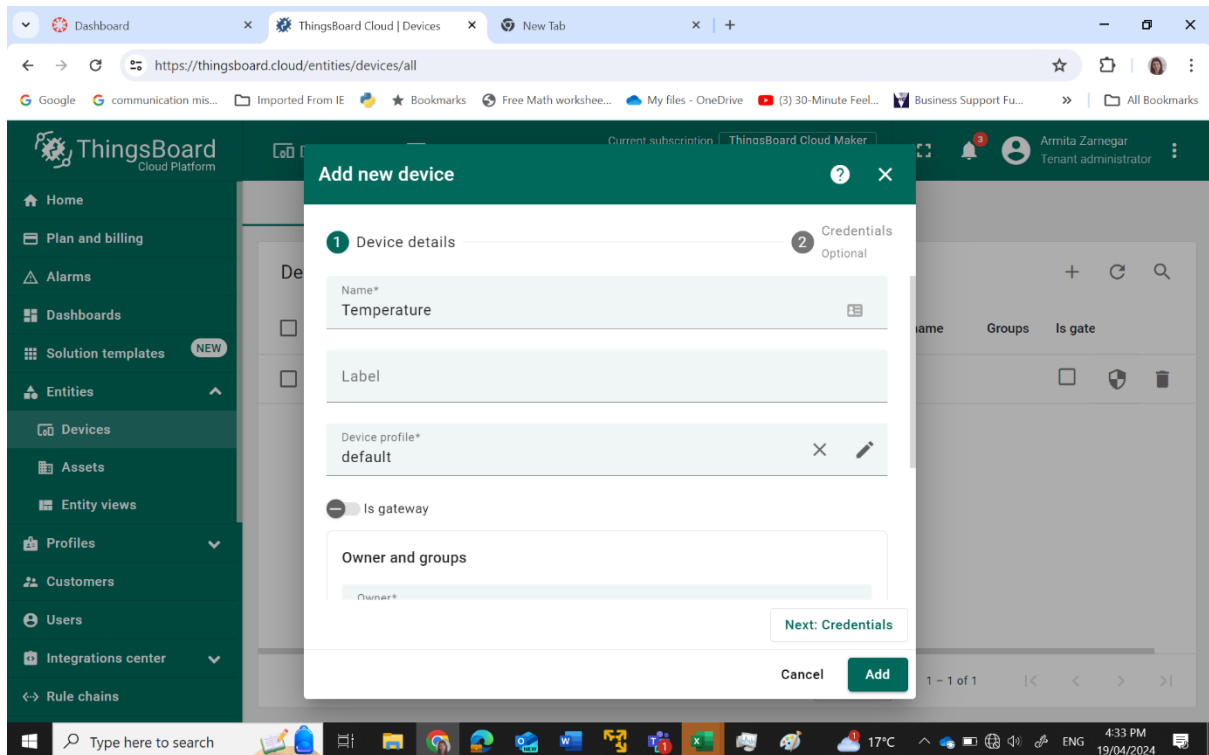
IoT Programming

The screenshot shows the ThingsBoard Cloud Platform home dashboard. The left sidebar contains navigation links: Home, Plan and billing, Alarms, Dashboards, Solution templates (marked as NEW), Entities, Devices, Assets, Entity views, Profiles, Customers, Users, Integrations center, and Rule chains. The main content area displays various widgets: Solution templates (Temperature & Humidity, Smart office, Fleet tracking, Fuel Level Monitoring, Air Quality Monitoring), Devices (Inactive: 1, Active: 0, Total: 1), Alarms (Critical: 0, Assigned to me: 0, Total: 0), Dashboards (Last viewed: 0, Add dashboard), Activity (History - last 30 days), Quick links (Alarms, Dashboards, Devices), Documentation (Getting started, Rule engine, API, Device profiles), Usage (Entities: 1/30, Devices: 0/30, Assets: 1/5, Users: 1/25, Dashboards: 0/5, Customers: 0/5), and a Get started guide with steps: 1. Create device, 2. Connect device, 3. Create dashboard, 4. Configure alarm rules, 5. Create alarm, 6. Create customer and share dashboard.

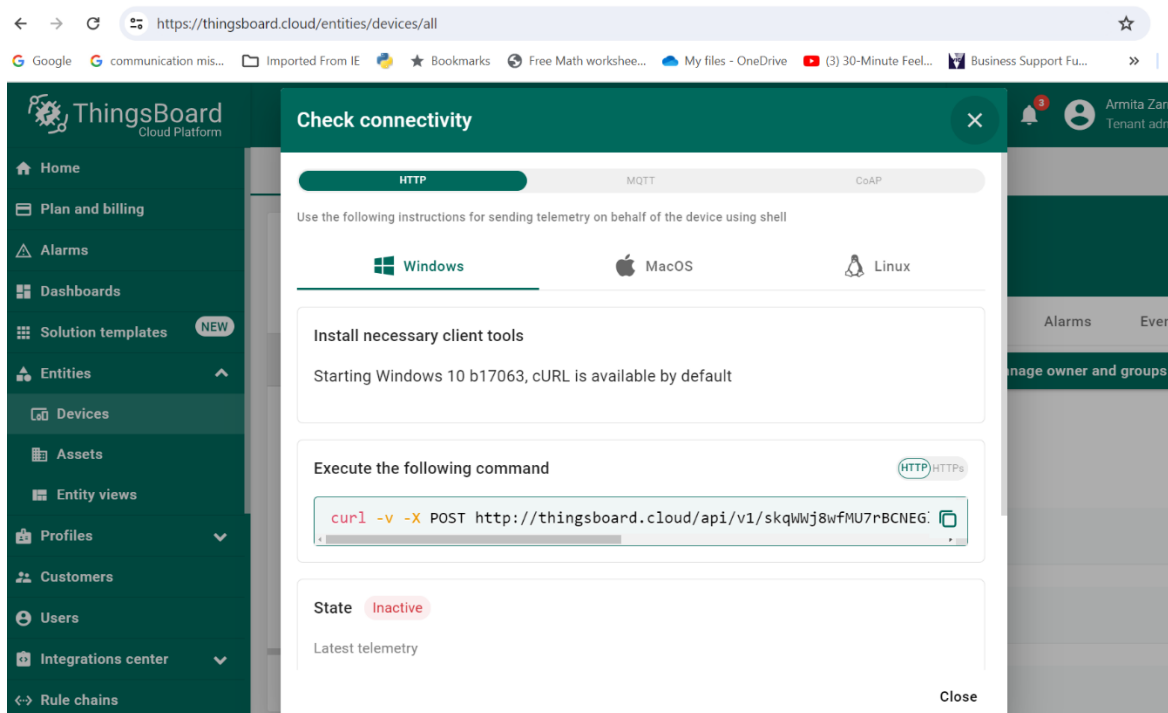
Then navigate to Entities and add a new device.

The screenshot shows the 'All Devices' page in the ThingsBoard Cloud Platform. The left sidebar is the same as the previous screenshot. The main content area shows a table of devices with columns: Created time, Name, Device profile, Label, State, and Customer. A 'Device Filter' button is present. A dropdown menu is open, showing 'Add new device' and 'Import device' options. The table contains one device entry: Created time: 2024-04-09 11:40:26, Name: Temp, Device profile: default, State: Inactive. The bottom of the page shows a Windows taskbar with the search bar and various application icons.

Choose a name for the device then leave all settings intact and add the device.

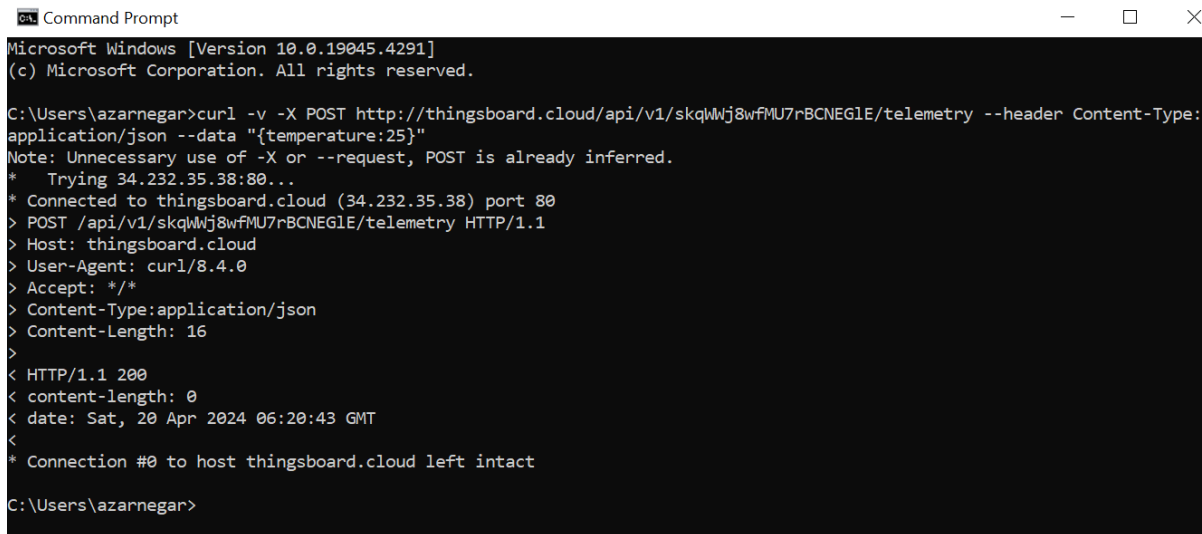


After creating the device, you need to connect to it to activate it. You can do this by clicking on the device and selecting 'Check Connectivity'. This will bring up the following menu.



Select the command to run based on your client operating system. If you are using Windows, copy the command provided and run it in Command Prompt. To open Command Prompt (cmd), type it into the search box next to the Windows icon and

double-click on it. After running the command, you will see the following messages confirming the successful connection.



```

Microsoft Windows [Version 10.0.19045.4291]
(c) Microsoft Corporation. All rights reserved.

C:\Users\azarnegar>curl -v -X POST http://thingsboard.cloud/api/v1/skqWwJ8wfMU7rBCNEGLE/telemetry --header Content-Type:
application/json --data "{temperature:25}"
Note: Unnecessary use of -X or --request, POST is already inferred.
* Trying 34.232.35.38:80...
* Connected to thingsboard.cloud (34.232.35.38) port 80
> POST /api/v1/skqWwJ8wfMU7rBCNEGLE/telemetry HTTP/1.1
> Host: thingsboard.cloud
> User-Agent: curl/8.4.0
> Accept: */*
> Content-Type:application/json
> Content-Length: 16
>
< HTTP/1.1 200
< content-length: 0
< date: Sat, 20 Apr 2024 06:20:43 GMT
<
* Connection #0 to host thingsboard.cloud left intact

C:\Users\azarnegar>

```

Now, you'll need to transfer the attached Python code to yourself via email or place it in the shared drive between the host and the VM for easy transfer to your VM. Once you have the Python code accessible on your VM, open a terminal and enter the following command:

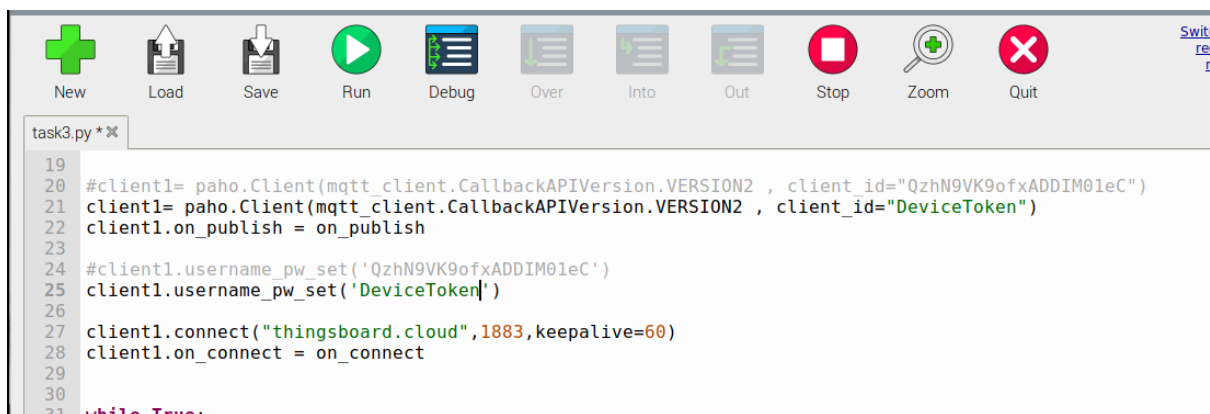


```

azarnegar@raspberrypi:~ $ sudo pip install paho-mqtt
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting paho-mqtt
  Downloading https://www.piwheels.org/simple/paho-mqtt/paho_mqtt-2.0.0-py3-none-any.whl (66 kB)
    | 66 kB 89 kB/s
Installing collected packages: paho-mqtt
Successfully installed paho-mqtt-2.0.0
azarnegar@raspberrypi:~ $

```

Next, open the provided Python code in the Thonny Python IDE. To do this, click on the Raspberry icon, then navigate to 'Programming' and select 'Thonny'. In the code, replace the placeholder token with your specific token. To obtain your token, proceed to the next step.

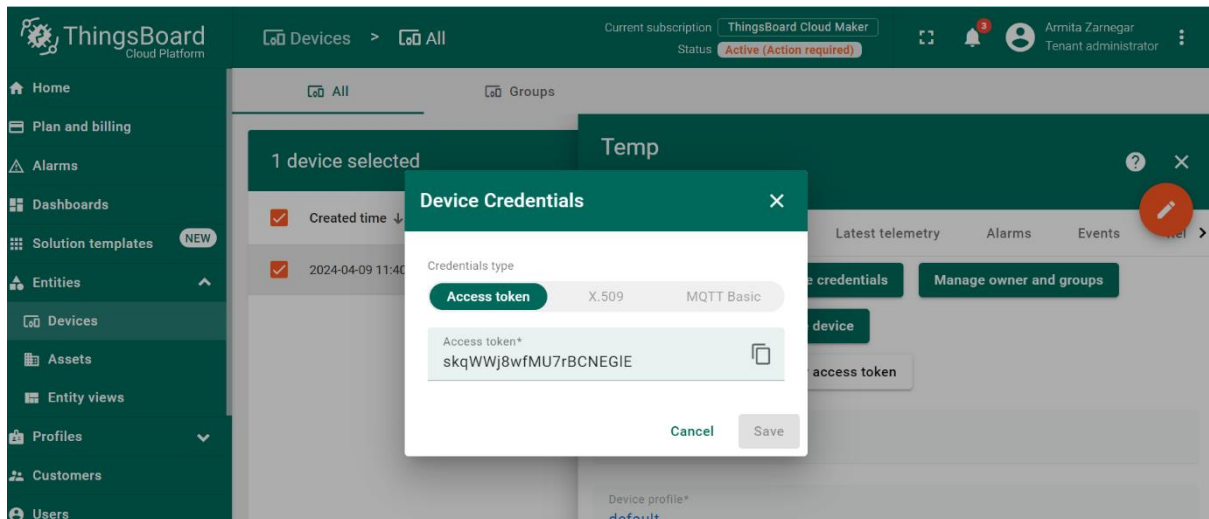


```

task3.py *
19
20 #client1= paho.Client(mqtt_client.CallbackAPIVersion.VERSION2 , client_id="QzhN9VK9ofxADDIM01eC")
21 client1= paho.Client(mqtt_client.CallbackAPIVersion.VERSION2 , client_id="DeviceToken")
22 client1.on_publish = on_publish
23
24 #client1.username_pw_set('QzhN9VK9ofxADDIM01eC')
25 client1.username_pw_set('DeviceToken')
26
27 client1.connect("thingsboard.cloud",1883,keepalive=60)
28 client1.on_connect = on_connect
29
30
31 while True:

```

To obtain the access token, log in to your ThingsBoard account and navigate to 'Devices'. Select the device you have just created, then click on the device to open a menu. From the menu, select 'Manage Credentials'.



Firstly, run the initial piece of code provided in the Arduino IDE. Next, execute the Python code on the Raspberry Pi VM. Upon successful execution, navigate to the 'Device' tab and click on 'Latest Telemetry'. You should then observe the temperature reading that the Arduino sent (in our code, it is 34) displayed there.

