Name: Hemant Harekrishna Mahto SUID: 597044038

IOT ASSIGNMENT 02

We need to create two stations that collect data from different sensors such as temperature and rainfall and publish their status using an MQTT broker hosted by ThingsBoard dashboard on a cloud platform.

To view the dashboard, you should ensure that the port number and access token for the second station are open and properly set up. Additionally, you should verify that the broker where the dashboard is hosted is functioning correctly and accessible. Finally, compare the access token you have with the token in the code to ensure they match.

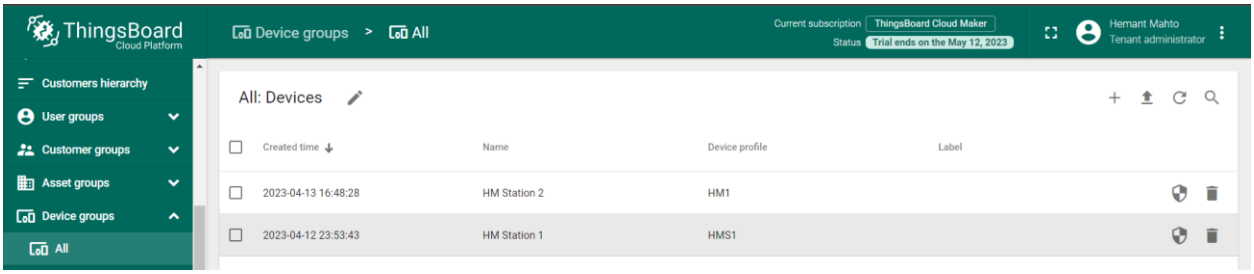

Figure 1: Devices

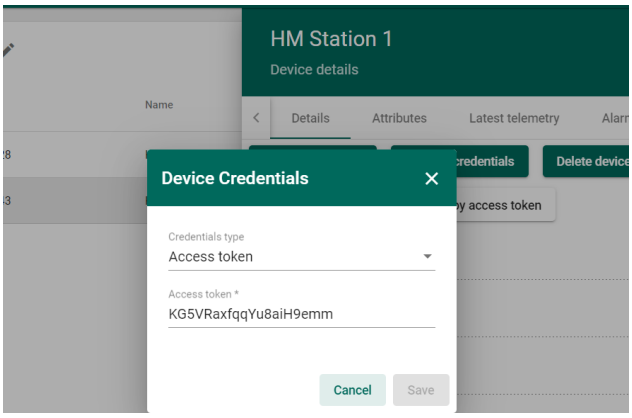

Figure 2.1: Secret Token of Station 1

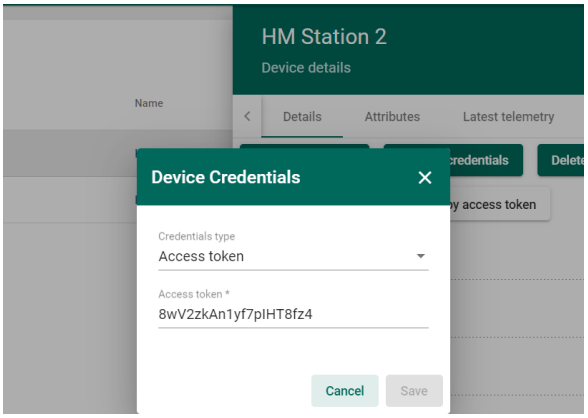

Figure 2.2: Secret Token of Station 2

```
# MQTT client definition
ACCESS_TOKEN1 = 'KG5VRaxfqqYu8aiH9emm'  # AccessToken of Virtual Environment Station 1 in ThingsBoard
ACCESS_TOKEN2 = '8wV2zkAn1yf7pIHT8fz4'  # AccessToken of Virtual Environment Station 2 in ThingsBoard

broker = "demo.thingsboard.io"  # Host Name
port = 1883                     # Data Listening port
```

Figure 3: Access Token in Python Code

Create a payload, then release it: We may establish a connection between the client and our broker using host:"demo.thingsboard.io", port:1883 and topic:"v1/devices/me/telemetry."

It is constructed using Python and sent in JSON format. We must enter our access token in the "username" field on the connection form to complete it.

Remote connectivity is crucial since only then will the payload display live data and dashboard readings. Additionally, download the required libraries to speed up the connection to the cloud service.

First, we download httpie and use the command line tool httpie to make HTTP requests with JSON payloads (For Windows).

Real-time modifications require the WebSocket API and its authorization, which may be deduced from: Using this bash script and an X-Authorization token, http POST http://demo.thingsboard.io/api/auth/login 'Content-Type:application/json' 'Accept:application/json' username=iotassignment2@thingsboard.org password=hemant

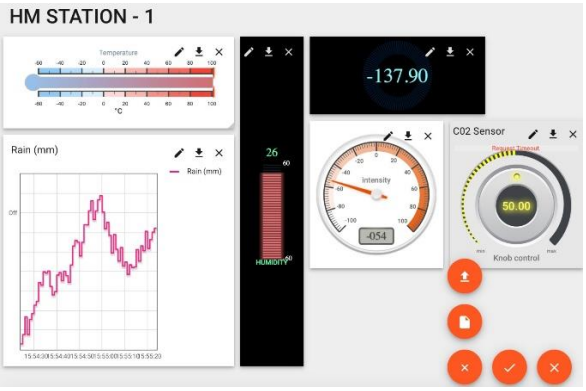

Figure 4.1: WorkStation 1

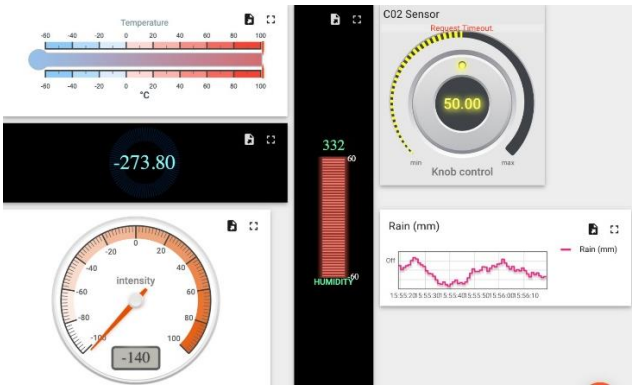

Figure 4.2: WorkStation 2

We did the same thing for WorkStation 2.

- GitHub Repository contains the Python script, a program that sends fake telemetry data to two virtual environment stations in the ThingsBoard IoT platform.
- It uses the paho-mqtt library for establishing MQTT connections and sending the data as payloads to the ThingsBoard cloud.
- First, it imports necessary libraries like paho-mqtt, time, and random for generating random telemetry data.
- Then, it defines functions to generate random values for temperature, humidity, wind direction, wind intensity, and rain height, which are called by the get_payload function to generate a JSON payload of telemetry data.
- The script creates two MQTT clients and connects them to ThingsBoard cloud using access tokens specific to each virtual environment station.
- The client objects also have a callback function assigned to handle publishing events.
- The script then enters a loop that generates new payloads every five seconds and sends them to their respective virtual environment stations using MQTT clients.
- The payloads are printed to the console, and the loop continues indefinitely until the script is stopped.

GitHub Repository: https://github.com/HemantMahto/IOT-Assignment-02