

HO CHI MINH CITY, UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEER



Application Based Internet of Things Report - LAB 3

Student: Vu Minh Quang

ID: 1852699

HỒ CHÍ MINH CITY



Content

1	Introduction	2
2	Requirements	2
2.1	Thingsboard server	2
2.2	Python gateway	2
2.3	Microbit sensor	3
3	Report	3
3.1	Thingsboard dashboard	3
3.2	Python source code	4
3.2.1	Publish data to Thingsboard	4
3.2.2	Command detector from 2 buttons	5
3.2.3	Github link	6
4	Microbit sensor	6



1 Introduction

In this second LAB, students are proposed to develop a simple application having the sensory data shown in a Thingsboard dashboard and a 2 buttons to illustrate the controlling functions.

The Microbit board is used in this LAB, connected to your PC via USB connector. In the microbit, 2 sensors can be used in this LAB, including the **TEMPERATURE** and the **LIGHT** sensor.

Every source code of this LAB is also required to publish in your Github. The details are described in the next section of this report.

2 Requirements

2.1 Thingsboard server

In order to present the sensory data, two UIs are required:

- A graph to display the sensory data
- A label or a Gauge view to display the current data

Students can create 2 different graphs for this requirement. However, one graph to display 2 different sensory data is also allowed.

Finally, two different buttons, labled **LED** and **FAN** are also required in the dashboard.

2.2 Python gateway

A python source code, running on your PC to receive the sensory data and publish to the server. Moreover, a command when a button is clicked, is needed to send to the Microbit platform.

A skeleton of the source code is published in this link:

https://github.com/npnlab-vn/code-manager/blob/ThingsBoard_Lab3/IoT_Lab.py

Following package are required to install for this source code:

- pip install paho-client
- pip install pyserial

Students are proposed to change the **THINGS_BOARD_ACCESS_TOKEN** and the name of the COM port (line 13 and 14), in the case it is connected to the Microbit.

2.3 Microbit sensor

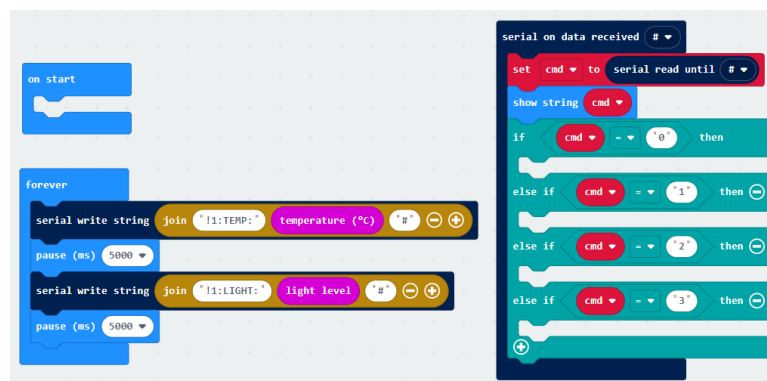
The temperature value of the microbit is sent to the gateway first. After that 5 seconds, the light sensor value is sent. The command is received at the microbit through serial connection. A reference source code can be found at:

https://makecode.microbit.org/_Cj3JUFXij6bu

In the case there is an error when downloading the hex file to the microbit platform, following steps are required to fix the error:

- Step 1: Create a new project in MakeCode
- Step 2: Switch to JavaScript view
- Step 3: From the reference source code, switch to JavaScript view as well
- Step 4: Copy the java script source code from the reference link to your project
- Step 5: Switch to Block view

If the error can not be fixed, students can implement the source code from the beginning. The image of the source code is depicted bellow:



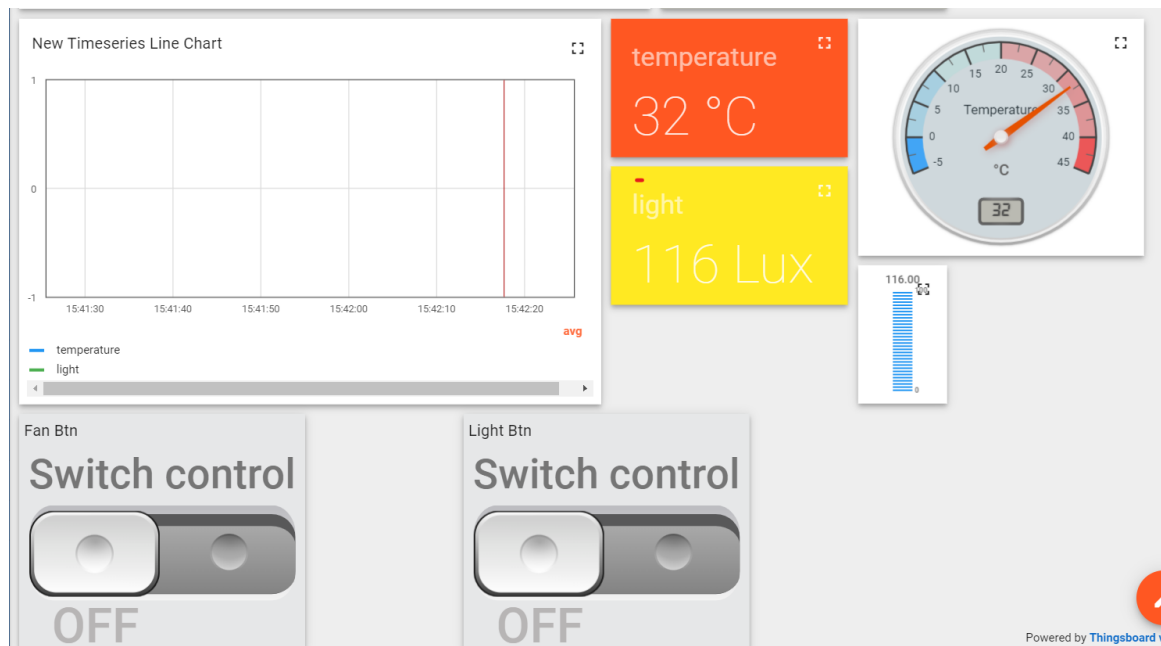
Hình 1: A simple program in BBC Microbit platform

3 Report

3.1 Thingsboard dashboard

A picture as well as the public link your your dashboard are required to present in this report.

In the case an UI is import from third party library (e.g. from github), a brief description to use this UI can be provided here. **These advance UIs can be considered as an extra point for this lab.**



Hình 2: Dashboard setup

3.2 Python source code

Students are proposed to present some part of the source code, as listed following.

3.2.1 Publish data to Thingsboard

In order to publish the data to the **Thingsboard server**, students have to add more source code to the **processData** function. Please fill your implementation in this report.

Listing 1: Publishing data to Thingsboard

```
1 def processData(data):
2     data = data.replace("!", "")
3     data = data.replace("#", "")
4     splitData = data.split(":")
5     print(splitData)
6     # TODO: Add your source code to publish data to the server
7     if (splitData[1] == 'LIGHT'):
8         data = {
9             'light': splitData[2]
10        }
11    if (splitData[1] == 'TEMP'):
12        data = {
13            'temperature': splitData[2]
14        }
15
16    client.publish('v1/devices/me/telemetry', json.dumps(data), 1)
```

3.2.2 Command detector from 2 buttons

In this part, the python gateway needs to send 4 different values to the microbit. Please add your implementation to finalize this part.

Listing 2: Command processing at the gateway

```
1 FAN = False
2 LED = False
3 def recv_message(client, userdata, message):
4     print("Received: ", message.payload.decode("utf-8"))
5     temp_data = {'value': True}
6     cmd = 1
7     # TODO: Update the cmd to control 2 devices
8     global FAN, LED
9     try:
10         jsonobj = json.loads(message.payload)
11         if jsonobj['method'] == "setLED":
12             temp_data['value'] = jsonobj['params']
13             LED = temp_data['value']
14             client.publish('v1/devices/me/attributes',
15                             json.dumps(temp_data), 1)
16         if jsonobj['method'] == "setFAN":
17             temp_data['value'] = jsonobj['params']
18             FAN = temp_data['value']
19             client.publish('v1/devices/me/attributes',
20                             json.dumps(temp_data), 1)
21     except:
22         pass
23
24     if FAN and LED:
25         cmd = 0
26     if not FAN and LED:
27         cmd = 1
28     if FAN and not LED:
29         cmd = 2
30     if not FAN and not LED:
31         cmd = 3
32
33     if len(bbc_port) > 0:
34         ser.write((str(cmd) + "#").encode())
```

Your source code is mostly in the **try except** block. The **cmd** variable is required to update

here.

3.2.3 Github link

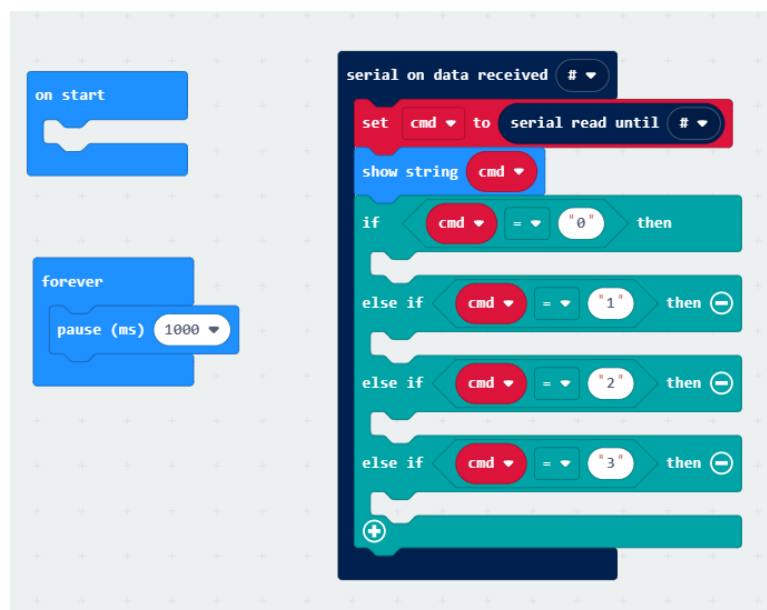
Finally, the github link of your python source code is provided in this part.

4 Microbit sensor

Finalize your source code and display 4 different images (using show icons block, in the Basic group). Then, capture your source code in MakeCode and present it in this report.

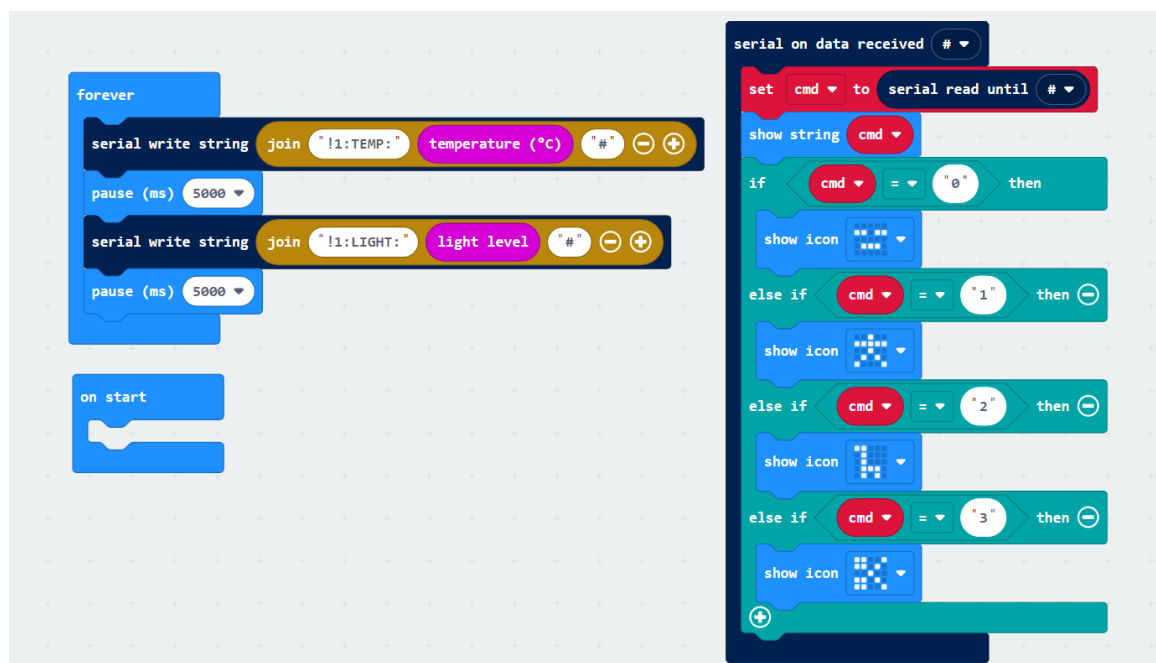
An extra point for this part which is described as follows:

- Modify the source code in forever, which is used to send sensory data to the gateway.
- There is only one delay instruction (pause(1000)) at the end of the forever
- The temperature is sent first. After that 5 seconds, the light is sent. This process is repeated again after 5 seconds



Hình 3: Structure of the program in Microbit

The picture of your source code is also required to present in this report.



Hình 4: *My structure of microbit*