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

Application Based Internet of Things Report - LAB 5

Student: Student Name
ID: 123456

HỒ CHÍ MINH CITY

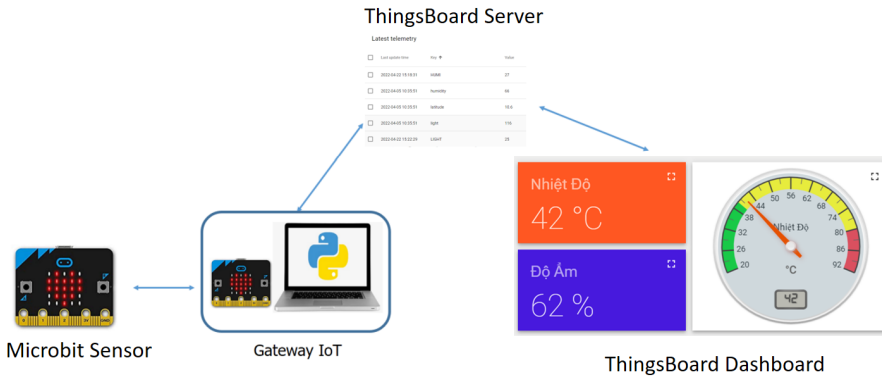
Content

1 Introduction

In this lab, students are supposed to implement an advanced feature for your IoT Gateway. Proposed options are decribed in following subsections.

1.1 Stop and Wait Protocol

Every communications in an IoT application should follow the Stop and Wait protocol. However, because of the Thingsboard server, some connections are impossible to apply this protocol, such as the connection betwen the python gateway and the Thingsboard server.



Hình 1: *Structure of the wireless sensor network*

However, there are 2 connections can be improved by the Stop and Wait, which are the connection between the main Microbit and the Python gateway (connected by serial) and the wireless communications between the Microbit sensors and the main Microbit. Students can improve one of them.

1.2 Simple AI Inference

A PC webcam can be used as an advanced sensor in your system. A simple AI model can be implemented at the gateway and the detected results are uploaded to Thingsboard server, then displayed on the Dashboard.

A reference for simple AI manual can be found in the link following (focus on the first 3 chapters):

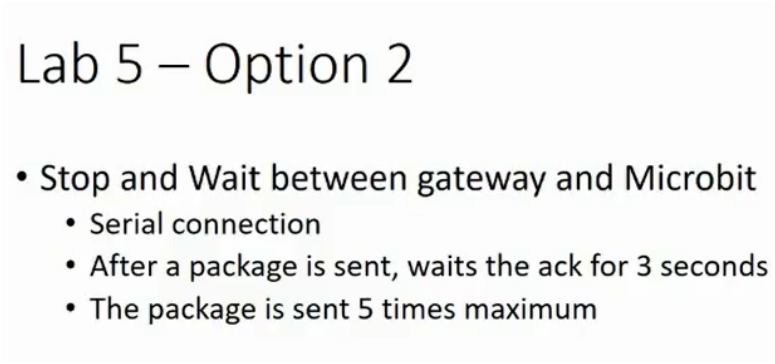
`https://drive.google.com/file/d/1d-VGlM5m_jFh9WkA38kg6v0o8zmF85v5/view`

2 Report

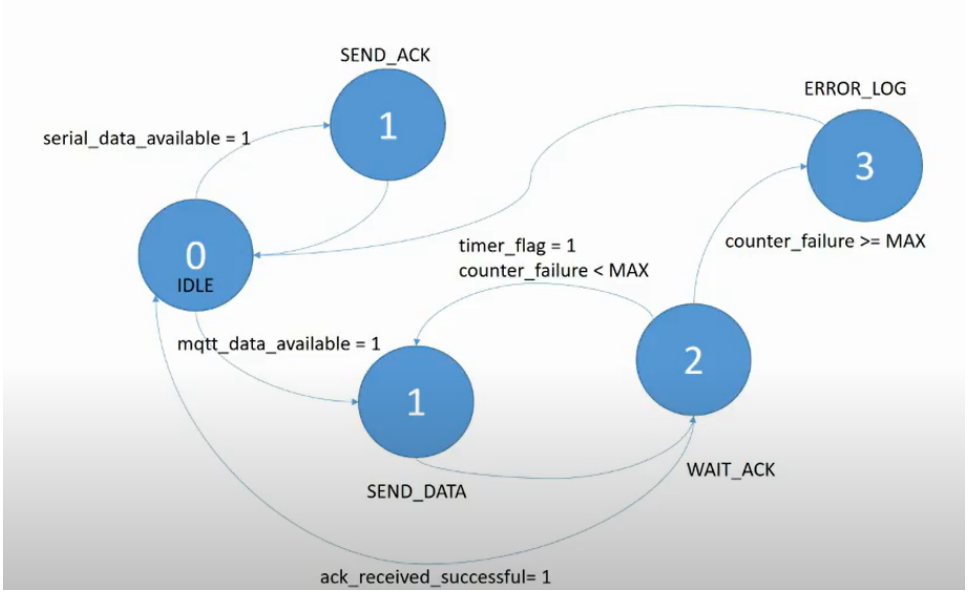
Please explain your solution in this report and provide the source code by a shareable link from MakeCode or github.

For the first option, explain the state machine, which is used to implement the Stop and Wait protocol.

Firstly, my option is the Stop and Wait protocol which State Machine and descriptions are as below:



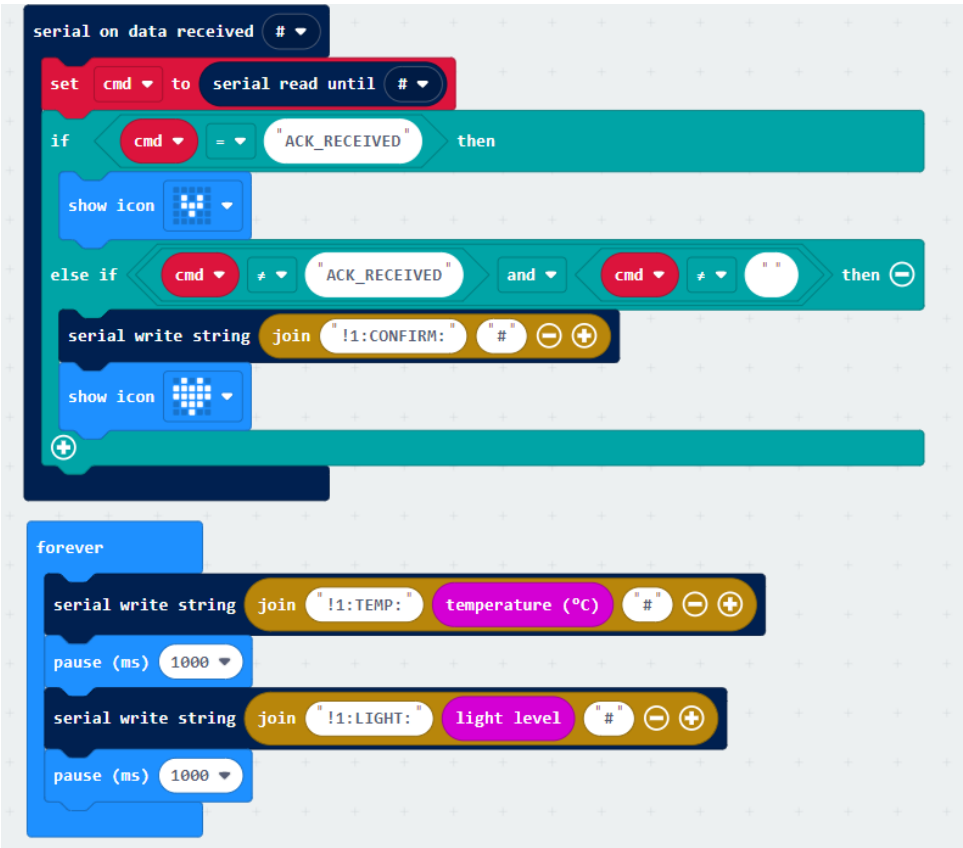
Hình 2: *Description*



Hình 3: *State Machine*

Lets begin with the Main-Microbit which receives data sensor from Sensor-Microbit and then send to Gateway by serial connection. What I do here is to make it as simple as possible: I

will assume that we already have data sensor and apply Stop and Wait protocol as picture below:



Hình 4: *Microbit MakeCode Blocks*

The shareable link from MakeCode:

`https://makecode.microbit.org/_6diRxy85tdHe`

I will explain the detail why the blocks look like above:

- The sensor data is sent to Gateway continuously (which I assume that we have data from Sensor Microbit).
- Whenever Main Mirobit receive data through serial connection, cmd will hold the String that Gateway transmit before the symbol #.
  - If (cmd == "ACK_RECEIVE") which indicate that Microbit receives ACK from Gateway to confirm that Gateway receives the sensor data, then show out the Image for User to know it.
  - If (cmd != "ACK_RECEIVE" and cmd != "") that means Microbit receives data from Gateway and will send ACK to Gateway to confirm and also show out another Image for user to know it.

About Python Gateway, i will explain detail the State Machine:

```
if status == 0:
    if serial_data_available == 1:
        serial_data_available = 0
        status = 1
    if mqtt_data_available == 1:
        mqtt_data_available = 0
        status = 2
```

Hình 5: *State 0*

State 0:

- Check the serial data if available then jump to State 1. The variable "serial_data_available" is checked whenever Gateway receives sensor data from Microbit:

```
def processData(data):
    global ack_available
    global serial_data_available
    serial_data_available = 1
    data = data.replace("!", "")
    data = data.replace("#", "")
    splitData = data.split(":")
    #print(splitData)
    # TODO: Add your source code to publish data to the server
    temp = 0
    light = 0
    if splitData[1] == "TEMP":
        temp = splitData[2]
        global temperature
        temperature = {'temperature': temp}
        client.publish('v1/devices/me/attributes', json.dumps(temperature), 1)

    if splitData[1] == "LIGHT":
        light = splitData[2]
        global lightning
        lightning = {'light': light}
        client.publish('v1/devices/me/attributes', json.dumps(lightning), 1)

    if splitData[1] == "CONFIRM":
        ack_available = 1
```

Hình 6: *Serial Data Available Check*

- Check the mqtt data if available then jump to State 2. The variable "mqtt_data_available" is checked whenever Gateway receives data from Server:

```python
def recv_message(client, userdata, message):
    global mqtt_data_available
    mqtt_data_available = 1
    print("Received: ", message.payload.decode("utf-8"))
    global temp_data
    temp_data = {'value': True}
    global cmd
    # TODO: Update the cmd to control 2 devices
    try:
        jsonobj = json.loads(message.payload)
        if jsonobj['method'] == "setLed" and jsonobj['params'] is True:
            cmd = 1
        if jsonobj['method'] == "setLed" and jsonobj['params'] is False:
            cmd = 2
        if jsonobj['method'] == "setFan" and jsonobj['params'] is True:
            cmd = 3
        if jsonobj['method'] == "setFan" and jsonobj['params'] is False:
            cmd = 4
        temp_data['value'] = jsonobj['params']
        client.publish('v1/devices/me/attributes', json.dumps(temp_data), 1)
```

Hình 7: *Mqtt Data Available Check*

State 1:

- Send_ACK to Microbit to confirm Serial data available then roll back to State 0.

```python
elif status == 1:
    send_ack()
    status = 0
```

Hình 8: *State 1*

```python
def send_ack():
    ser.write(("ACK_RECEIVED" + "#").encode())
```

Hình 9: *Send ACK*

State 2:

- Send_data get from Server which are encoded in Python Gateway through serial connection to Main Microbit and also set timer to count down for 3 seconds then jump to State 3. What is "check" does here is to prevent loop setTimer forever overide all other conditions.

```
elif status == 2:
    #readSerial()
    send_data()
    print("Send data: ", counter_failure + 1, "times")
    if check == 0:
        setTimer(3)
        check = 1
    status = 3
```

Hình 10: *State 2*

```
def send_data():
    ser.write((str(cmd) + "#").encode())
```

Hình 11: *Send Data*

```
def setTimer(counter):
    global timer_counter, timer_flag
    timer_counter = counter
    timer_flag = 0
```

Hình 12: *Set Timer*

```
def cancelTimer():
    global timer_counter, timer_flag
    timer_counter = 0
    timer_flag = 0
```

Hình 13: *Cancel Timer*

```
def runTimer():
    global timer_counter, timer_flag
    if timer_counter > 0:
        timer_counter = timer_counter - 1
        if timer_counter <= 0:
            timer_flag = 1
```

Hình 14: *Run Timer*

State 3:

- If Gateway receives ACK from Microbit then cancel timer reset everything and roll back to State 0.
- If timer_flag is checked that means finish waiting ACK 3 seconds once and also counter_failure increase 1.

- If counter_failure smaller than MAX = 5, roll back to state 2 to send_data the next time.
- If counter_failure reach MAX=5 that means we done 5 times sending package and jumps to State 4 which is used to reset everything, notify the Failure and roll back to State Idle(0).

```
elif status == 3:
    if ack_available == 1:
        print("ACK_RECEIVED_SUCCESSFUL")
        cancelTimer()
        counter_failure = 0
        status = 0
    if timer_flag == 1:
        counter_failure += 1
        check = 0
        if counter_failure < MAX:
            status = 2
        if counter_failure >= MAX:
            status = 4
```

Hình 15: *State 3*

State 4: Reset everything, show out the Failure and roll back to State 0(idle).

```
elif status == 4:
    status = 0
    counter_failure = 0
    print("!!!Failure NO ACK_RECEIVED!!!")
```

Hình 16: *State 4*

Link Github for this Lab:

`https://github.com/TieuLong98/Lab5_IOT.git`

Here are some results:

- As if there is no ACK from Microbit:

```
Current Status:  0
Thingsboard connected successfully!!
Subscribed...
Current Status:  0
Current Status:  1
Current Status:  0
Current Status:  1
Received:  {"method":"setFan","params":true}
Current Status:  0
Current Status:  2
Send data:  1 times
Current Status:  3
Current Status:  3
Current Status:  3
Current Status:  2
Send data:  2 times
Current Status:  3
Current Status:  3
Current Status:  3
Current Status:  2
Send data:  3 times
Current Status:  3
Current Status:  3
Current Status:  3
Current Status:  2
Send data:  4 times
Current Status:  3
Current Status:  3
Current Status:  3
Current Status:  2
Send data:  5 times
Current Status:  3
Current Status:  3
Current Status:  3
Current Status:  4
!!!Failure NO ACK_RECEIVED!!!
```

Hình 17: *Failure, No ACK received*

- If receiving ACK from Microbit:

```
Thingsboard connected successfully!!
Subscribed...
Current Status:  0
Current Status:  1
Current Status:  0
Current Status:  1
Current Status:  0
Current Status:  1
Current Status:  0
Current Status:  1
Received:  {"method":"setFan","params":false}
Current Status:  0
Current Status:  2
Send data:  1 times
Current Status:  3
ACK_RECEIVED_SUCCESSFUL
Current Status:  0
Current Status:  1
```

Hình 18: *Successful, ACK received*