

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



PHÁT TRIỂN ỨNG DỤNG INTERNET OF THINGS THỰC HÀNH
(CO3038)

LAB03 Gửi lệnh điều khiển tới thiết bị

BỘ MÔN: KỸ THUẬT MÁY TÍNH

GVHD: VŨ TRỌNG THIÊN

NHÓM LỚP: L01

—o0o—

SVTH: MAI THỊNH PHÁT (1914590)

TP. HỒ CHÍ MINH, 3/2024

Mục lục

1	Tổng quan	1
1.1	Giới thiệu sơ lược về mục đích của bài lab này	1
1.2	Cấu trúc bài báo cáo	1
2	Giả lập thiết bị phần cứng trên WSL	2
2.1	Cài đặt phần mềm giả lập	2
3	Hiện thực chương trình	3
3.1	khởi tạo kết nối	3
3.2	Kết quả của chương trình	4
3.3	Repo mã nguồn	5

Danh sách hình vẽ

2.1	Chuẩn bị dữ liệu	2
3.1	Bật đèn ở giao diện	5
3.2	Tín hiệu bật đèn được ghi nhận tại chương trình python	5
3.3	Lệnh mở đèn đã được gửi đến thiết bị phần cứng	5

Chương 1

Tổng quan

1.1 Giới thiệu sơ lược về mục đích của bài lab này

Bài thực hành này nhằm mục đích giúp hiểu rõ hơn và ứng dụng các kiến thức đã học ở giờ lý thuyết để hiện thực một chương trình đơn giản gửi lệnh điều khiển từ phía người dùng tới thiết bị phần cứng. Việc tương tác với các thiết bị phần cứng thông qua các lệnh điều khiển là một khía cạnh quan trọng của các hệ thống IOT.

Trong bài thực hành này em sẽ tập trung vào cách xây dựng một chương trình bằng python để nhận lệnh điều khiển thiết bị từ phía người dùng (ở đây là dashBoard được tích hợp sẵn trên Adafruit IO) và gửi các lệnh điều khiển này tới thiết bị phần cứng để điều khiển các thiết bị. Do sự giới hạn về nhân lực và vật lực nên thiết bị phần cứng ở đây sẽ được giả lập bằng phần mềm. Do đó các lệnh mà thiết bị phần cứng nhận được sẽ được hiển thị ra màn hình terminal của chương trình giả lập phần cứng.

1.2 Cấu trúc bài báo cáo

Cấu trúc của bài báo cáo này gồm có các chương sau:

- **Chương 1 Tổng quan:** Giới thiệu về bài thực hành.
- **Chương 2 Giả lập thiết bị phần cứng trên WSL:** Trình bày các bước giả lập phần cứng trên WSL
- **Chương 3 Hiện thực chương trình:** Trình bày các bước thực hiện chương trình bằng ngôn ngữ *python*

Chương 2

Giả lập thiết bị phần cứng trên WSL

Do chương trình được hiện thực trên Windows Subsystem for Linux (WSL) do đó việc truy cập vào các thiết bị thật của tài nguyên phần cứng hệ thống là rất khó khăn và hạn chế, vì vậy em sẽ tiến hành giả lập thiết bị phần cứng bằng phần mềm để từ đó có thể đọc và ghi trên thiết bị giả lập này

1. **Chuẩn bị dữ liệu:** Dữ liệu sẽ được chụp trực tiếp từ camera cho các lớp tương ứng.
2. **Huấn luyện:** Lựa chọn các siêu tham số phù hợp cho mô hình và tiến hành huấn luyện.
3. **Kiểm tra lại mô hình:** Sử dụng webcam và đánh giá mô hình trực tiếp trên ứng dụng Teachable Machine.

2.1 Cài đặt phần mềm giả lập

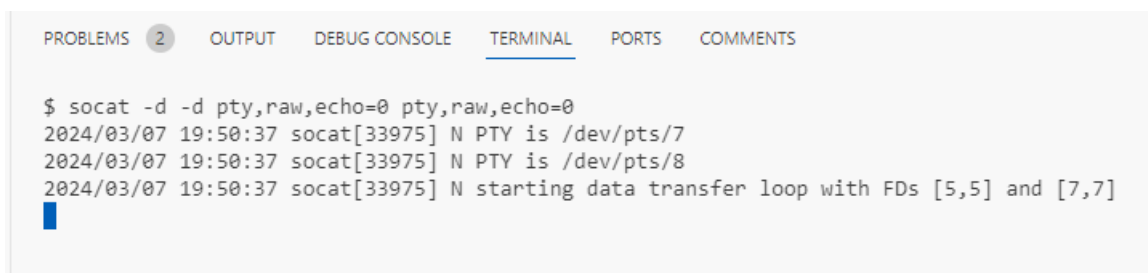
Để giả lập thiết bị phần cứng em đã cài đặt phần mềm còn tên là **socat** trên WSL bằng lệnh sau:

```
sudo apt-get install -y socat
```

Sau khi đã chạy cài xong tiến hành mở chương trình giả lập bằng lệnh sau:

```
socat -d -d pty,raw,echo=0 pty,raw,echo=0
```

Dưới đây là kết quả chạy được ở trên máy tính cá nhân, Tùy vào thiết bị của mỗi người mà con số trả về có thể sẽ khác nhau:



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
$ socat -d -d pty,raw,echo=0 pty,raw,echo=0
2024/03/07 19:50:37 socat[33975] N PTY is /dev/pts/7
2024/03/07 19:50:37 socat[33975] N PTY is /dev/pts/8
2024/03/07 19:50:37 socat[33975] N starting data transfer loop with FDs [5,5] and [7,7]
```

Hình 2.1: Chuẩn bị dữ liệu

Chương 3

Hiện thực chương trình

Trong phần này em sẽ trình bày các bước từ tạo kết nối với thiết bị phần cứng giả lập, viết hàm lắng nghe sự kiện từ Adafruit IO server, cho đến việc gửi lệnh đến thiết bị phần cứng giả lập đã tạo kết nối trước đó.

3.1 khởi tạo kết nối

Sau khi giả lập phần cứng trên *WSL* xong chúng ta sẽ tìm và kết nối với thiết bị phần cứng bằng đoạn mã sau:

```
1 import serial.tools.list_ports
2
3 def getPort():
4     ports = serial.tools.list_ports.comports()
5     N = len(ports)
6     commPort = "None"
7     for i in range(0, N):
8         port = ports[i]
9         strPort = str(port)
10        if "USB Serial Device" in strPort:
11            splitPort = strPort.split(" ")
12            commPort = (splitPort[0])
13        # return commPort
14        # return "COM3"
15        return '/dev/pts/3'
16
17 if getPort()!="None":
18     ser = serial.Serial( port=getPort(), baudrate=115200)
19     print(ser)
20
21
```

Listing 3.1: Khởi tạo kết nối

Trong hàm **getPort** được định nghĩa để nhằm tìm kiếm các cổng kết nối đang mở. Tuy nhiên do việc hiện thực trên *WSL* để truy xuất vào các thành phần vật lý rất phức tạp nên trong bài thực hành này em sẽ gán trực tiếp cổng tại **/dev/pts/3** cho chương trình python. Câu lệnh ở dòng 17 đến 18 là việc kiểm tra và kết nối đến cổng đã trả ra ở hàm *getPort*.

để có thể gửi được lệnh xuống thiết bị phần cứng ta sẽ định nghĩa một hàm để thực hiện vấn đề này:

```
1 def writeData(data):
2     ser.write(str(data).encode())
3
```

Listing 3.2: định nghĩa hàm gửi lệnh xuống thiết bị

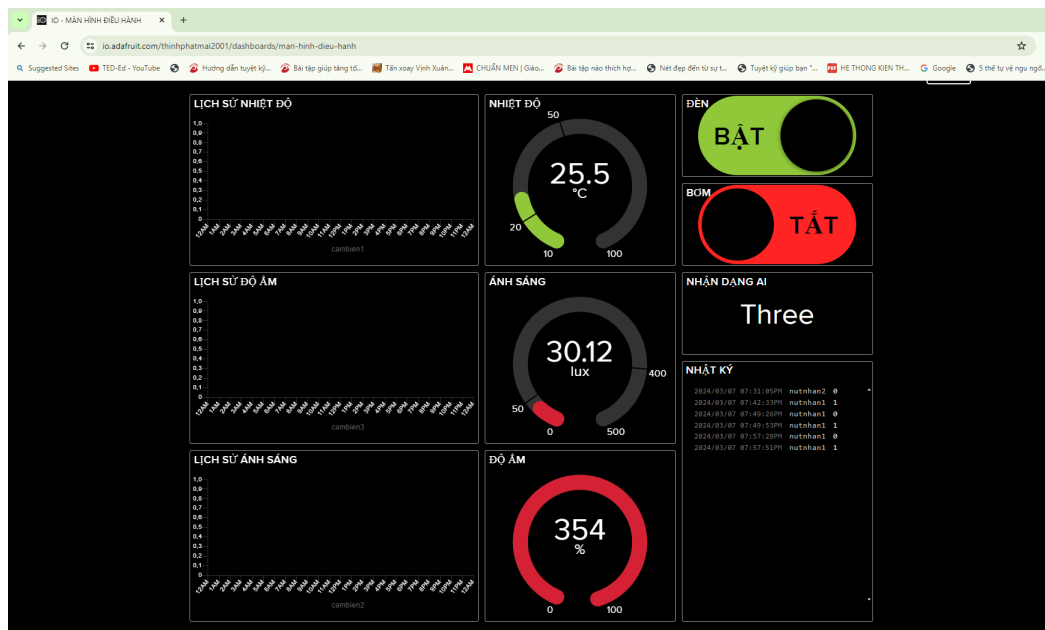
Bên phía file **main.py** ta sẽ định nghĩa một hàm callback dùng để lắng nghe dữ liệu từ server Adafruit IO và gửi lệnh xuống thiết bị phần cứng.

```
1 def encodeData(deviceID, data):
2     return {
3         "deviceID": deviceID,
4         "signal": data,
5     }
6
7 def message(client, feed_id, payload):
8     print("Nhan du lieu: " + payload + " , feed id: ", feed_id)
9     data = None
10    if feed_id == 'nutnhan1' or feed_id == 'nutnhan2':
11        if payload == '0':
12            data = encodeData(feed_id, "OFF")
13            # writeData(1)
14        else:
15            data = encodeData(feed_id, "ON")
16            # writeData(2)
17    if data != None:
18        writeData(data)
19
```

Listing 3.3: định nghĩa hàm gửi lệnh xuống thiết bị

3.2 Kết quả của chương trình

Sau khi đã thiết lập xong tất cả và tiến hành chạy chương trình ta sẽ thu được các kết quả sau:



Hình 3.1: Bật đèn ở giao diện

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
(project_env) IOT_project/lab03/ $ python main.py
Serial<id=0x7f433a96d540, open=True>(port="/dev/pts/3", baudrate=115200, bytesize=8, parity='N', stopbits=1, timeout=None, xonxoff=False, rtscts=False, dsrdtr=False)
Connected to Adafruit IO!
Ket noi thanh cong ...
Subscribe thanh cong ...
Subscribe thanh cong ...
Nhan du lieu: 1, feed id: nutnhan1
█
```

Hình 3.2: Tín hiệu bật đèn được ghi nhận tại chương trình python

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
$ socat -d -d pty,raw,echo=0 pty,raw,echo=0
2024/03/07 19:59:08 socat[37209] N PTY is /dev/pts/5
2024/03/07 19:59:08 socat[37209] N PTY is /dev/pts/7
2024/03/07 19:59:08 socat[37209] N starting data transfer loop with FDs [5,5] and [7,7]
{'deviceId': 'nutnhan1', 'signal': 'ON'}█
```

Hình 3.3: Lệnh mở đèn đã được gửi đến thiết bị phần cứng

3.3 Repo mã nguồn

Mã nguồn và báo cáo đều được lưu trữ tại: https://github.com/LeoPkm2-1/Lab_IOT_-C03038-/tree/master/IOT_LAB_FOR_WINDOWNS/LAB_03

Tài liệu tham khảo

- [1] The Python Tutorial
<https://docs.python.org/3/tutorial/index.html> [Truy cập: 20-02-2024]
- [2] Python Tutorial
<https://www.w3schools.com/python/> [Truy cập: 20-02-2024]
- [3] adafruit-io python
<https://pypi.org/project/adafruit-io/> [Truy cập: 20-02-2024]