



IOT LAB
REPORT

2024

(CO3038) Phát triển Ứng dụng
Internet Of Things
Bài thí nghiệm số 1

GVHD: VŨ TRỌNG THIÊN

Người viết: Tô Hoàng Phong - 2112012

Ho Chi Minh University of Technology (HCMUT)

Khoa khoa học và kỹ thuật máy tính

Mục lục

1	Đường dẫn github và một số đường dẫn khác (nếu có)	3
2	Xây dựng môi trường và kết quả	3
2.1	Các yêu cầu	3
2.2	Một số kết quả	4

1 Đường dẫn github và một số đường dẫn khác (nếu có)

- Đường dẫn github của bài thí nghiệm: [github](#)
- Đường dẫn dashboard của bài thí nghiệm: [dashboard](#)
- Đường dẫn video demo (nếu có)

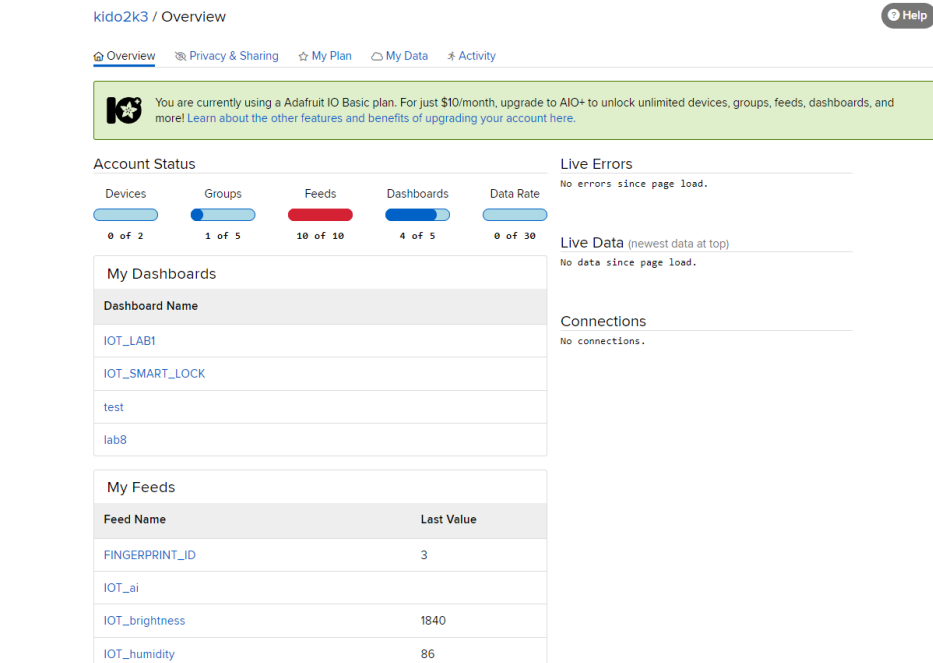
2 Xây dựng môi trường và kết quả

2.1 Các yêu cầu

Các yêu cầu đã thực hiện được trong phần thí nghiệm này:

- Đăng ký - đăng nhập tài khoản *Adafruit.io*.
- Tạo và cài đặt *feeds* trên *Adafruit.io*.
- Tạo và cài đặt *dashboards* trên *Adafruit.io*.
- Cài đặt *vscode* và *WSL*.
- Kết nối, *subscribe*, *publish* một số dữ liệu từ *local computer* (máy tính cá nhân) lên *Adafruit server*.
- Chạy code trên môi trường Window.
- Chạy code trên môi trường Linux.

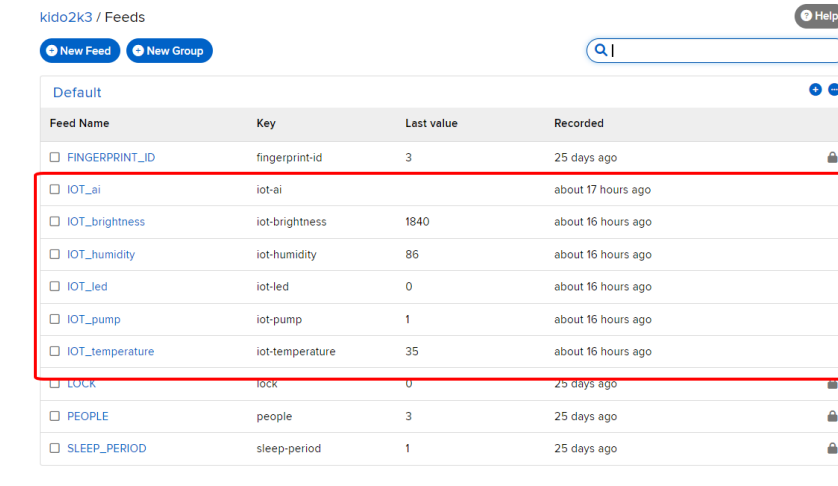
2.2 Một số kết quả



Hình 1: Tạo tài khoản trên *Adafruit.io*

Nhận xét: ở Hình 1 tab ‘Overview’, có tất cả thông tin tài khoản trên *Adafruit.io*.

Ví dụ: tên ‘kido2k3’, số lượng *feeds* và *dashboards*, danh sách *feeds* và *dashboards* đã tạo.



Default

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> FINGERPRINT_ID	fingerprint-id	3	25 days ago
<input type="checkbox"/> IOT_ai	iot-ai		about 17 hours ago
<input type="checkbox"/> IOT_brightness	iot-brightness	1840	about 16 hours ago
<input type="checkbox"/> IOT_humidity	iot-humidity	86	about 16 hours ago
<input type="checkbox"/> IOT_led	iot-led	0	about 16 hours ago
<input type="checkbox"/> IOT_pump	iot-pump	1	about 16 hours ago
<input type="checkbox"/> IOT_temperature	iot-temperature	35	about 16 hours ago
<input type="checkbox"/> LOCK	lock	0	25 days ago
<input type="checkbox"/> PEOPLE	people	3	25 days ago
<input type="checkbox"/> SLEEP_PERIOD	sleep-period	1	25 days ago

Hình 2: Danh sách các *feeds*

Nhận xét: Hình 2 là danh sách tất cả các *feeds* hiện có trên server. Ở trong thí nghiệm này (và tất cả thí nghiệm sau), tên của *feeds* sẽ có tiền tố ‘IOT’ ở phía trước nhằm phân biệt với các dự án khác.

kido2k3 / Dashboards

New Dashboard

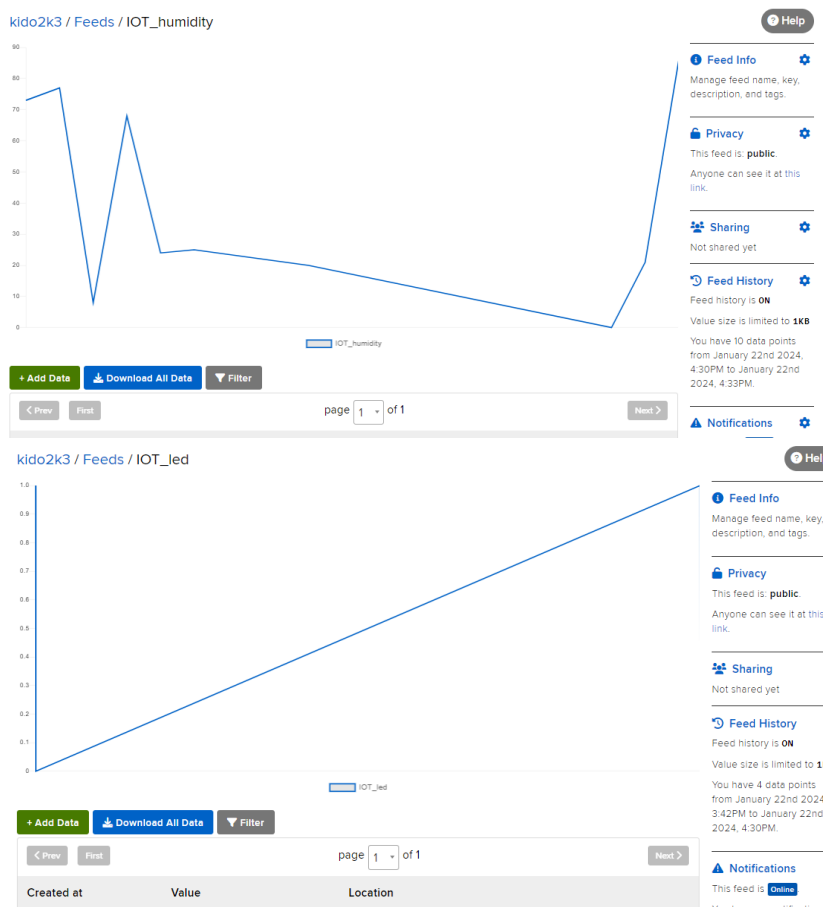
Search

Name	Key	Created At
IOT_LAB1	iot-lab1	January 22, 2024
IOT_SMART_LOCK	iot-smart-lock	December 18, 2023
lab8	lab8	November 27, 2023
test	test	December 16, 2023

Loaded in 0.87 seconds.

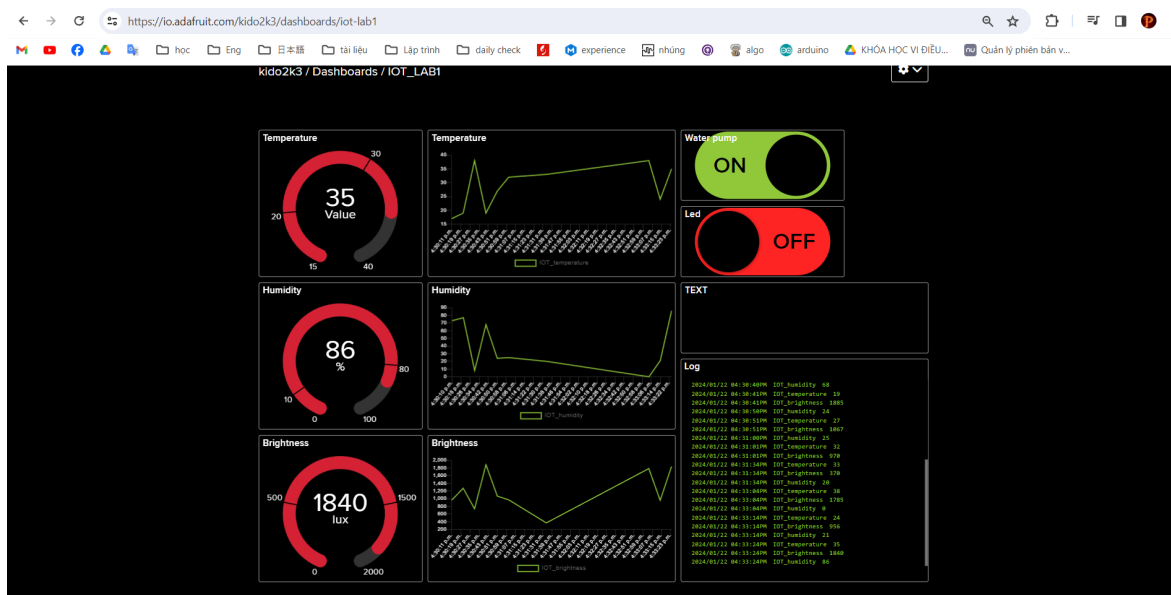
Hình 3: Danh sách các *dashboards*

Nhận xét: Hình 3 là danh sách tất cả các *dashboards* hiện có trên server. Ở trong thí nghiệm này (và tất cả thí nghiệm sau), tên của *dashboards* sẽ là 'IOT_LAB<no.lab>', ví dụ: IOT_LAB1.



Hình 4: IOT_humidity và IOT_led feeds

Nhận xét: Hình 4 chỉ ra một số feeds tiêu biểu. IOT_humidity (độ ẩm, thường sử dụng phương thức *public*), tương tự cho IOT_temperature (nhiệt độ), IOT_brightness (độ sáng), IOT_ai (trí tuệ nhân tạo) IOT_led (đèn, thường sử dụng phương thức *subscribe*), tương tự cho IOT_pump (máy bơm)



Hình 5: IOT_LAB1 dashboard

Nhận xét: Dashboard được bố trí theo 3 cột.
 Ở bên trái là giá trị gần nhất tại thời điểm lấy mẫu.
 Biểu đồ đường ghi lại tất cả giá trị trong vòng 24 giờ được đặt ở giữa.
 Bên phải chứa một số nút nhấn, nhật ký (log),...

```
C:\Users\HP>wsl --version
WSL version: 2.0.9.0
Kernel version: 5.15.133.1-1
WSLg version: 1.0.59
MSRDC version: 1.2.4677
Direct3D version: 1.611.1-81528511
DXCore version: 10.0.25131.1002-220531-1700.rs-onecore-base2-hyp
Windows version: 10.0.19045.3930

C:\Users\HP>_
```

Hình 6: Cài đặt WSL

Nhận xét: phiên bản của WSL trên máy là 2.0.9.0

```

main.py M X
main.py > ...
You, 1 second ago | 1 author (You)
1 # Import standard python modules.
2 import random
3 import sys
4 import time
5
6 # Import Adafruit IO MQTT client.
7 from Adafruit_IO import MQTTClient
8
9
10 ADAFRUIT_IO_KEY = "aio_kHTu8402Xlf274wB8cmdf4b2UKHA"
11 ADAFRUIT_IO_USERNAME = "kido2k3"
12 FEED_KEY = ["iot-brightness", "iot-ai", "iot-humidity", "iot-led", "iot-pump", "iot-temperature"]
13
14 You, 23 hours ago • finished most...
15 # Define callback functions which will be called when certain events happen.
16 def connected(client):
17     # Connected function will be called when the client is connected to Adafruit IO.
18     # This is a good place to subscribe to feed changes. The client parameter
19     # passed to this function is the Adafruit IO MQTT client so you can make
20     # calls against it easily.
21     print('Connected to Adafruit IO! Listening for DemoFeed changes...')
22
23
24 PS E:\dai_hoc\IoTApplicationDesign\IOT_LABS> python .\main.py
25 Publishing a new message every 10 seconds (press Ctrl-C to quit)...
26 Connected to Adafruit IO!
27 Connected to Adafruit IO! Listening for DemoFeed changes...
28 Subscribe thanh cong.....
29 Subscribe thanh cong.....
30 Subscribe thanh cong.....
31 Subscribe thanh cong.....
32 Subscribe thanh cong.....
33 Subscribe thanh cong.....
34

```

Hình 7: Chạy code python trên môi trường Window

Nhận xét: Đoạn code trên đã subscribe thành công tất cả 6 feeds và in lên terminal 6 dòng ‘Subscribe thanh cong.....’

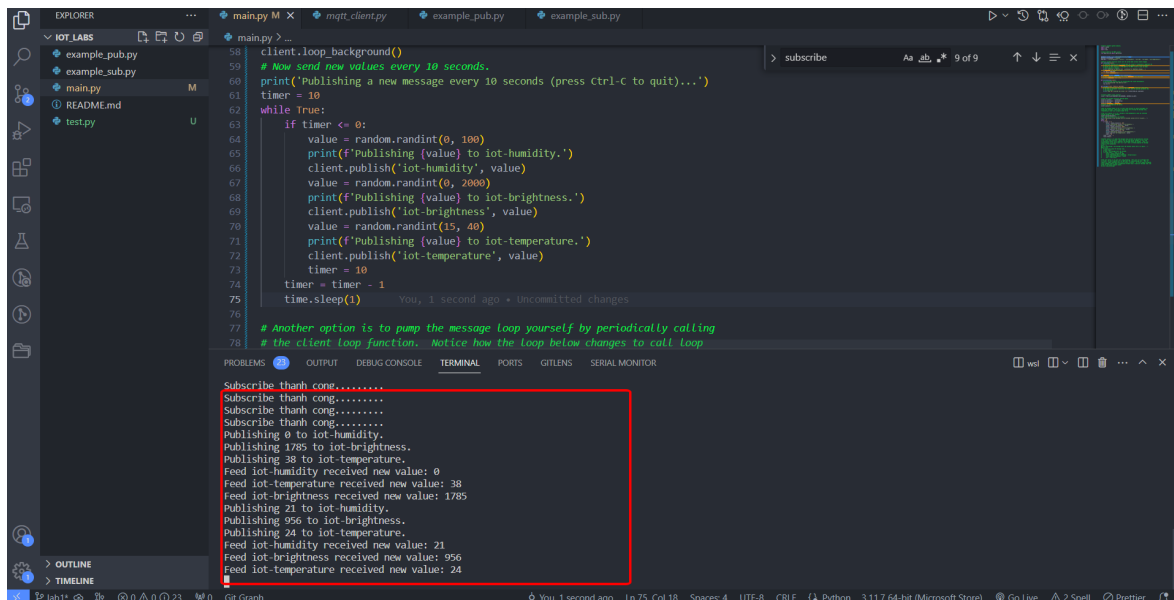
```

IOT_LABS
main.py > ...
58 client.loop_background()
59 # Now send new values every 10 seconds.
60 print('Publishing a new message every 10 seconds (press Ctrl-C to quit)...')
61 timer = 10
62 while True:
63     if timer <= 0:
64         value = random.randint(0, 100)
65         print(f'Publishing {value} to iot-humidity.')
66         client.publish('iot-humidity', value)
67         value = random.randint(0, 2000)
68         print(f'Publishing {value} to iot-brightness.')
69         client.publish('iot-brightness', value)
70         value = random.randint(15, 40)
71         print(f'Publishing {value} to iot-temperature.')
72         client.publish('iot-temperature', value)
73         timer = 10
74         timer = timer - 1
75         time.sleep(1)
76
77 You, 1 second ago • uncommitted changes
78 # Another option is to pump the message loop yourself by periodically calling
79 # the client loop function. Notice how the loop below changes to call loop
80
81
82 PS E:\dai_hoc\IoTApplicationDesign\IOT_LABS> wsl
83 kido@DESKTOP-IQT1ABB:/mnt/e/dai_hoc/IOTApplicationDesign/IOT_LABS$ python3 main.py
84 Publishing a new message every 10 seconds (press Ctrl-C to quit)...
85 Connected to Adafruit IO!
86 Connected to Adafruit IO! Listening for DemoFeed changes...
87 Subscribe thanh cong.....
88 Subscribe thanh cong.....
89 Subscribe thanh cong.....
90 Subscribe thanh cong.....
91 Subscribe thanh cong.....
92 Subscribe thanh cong.....
93

```

Hình 8: Chạy code python trên môi trường Linux

Nhận xét: Tương tự như môi trường Window, nhưng trước khi chạy code, phải chuyển sang môi trường linux bằng lệnh ‘wsl’



```

58 client.loop_background()
59 # Now send new values every 10 seconds.
60 print('Publishing a new message every 10 seconds (press Ctrl-C to quit)...')
61 timer = 10
62 while True:
63     if timer <= 0:
64         value = random.randint(0, 100)
65         print(f'Publishing {value} to iot-humidity.')
66         client.publish("iot-humidity", value)
67         value = random.randint(0, 2000)
68         print(f'Publishing {value} to iot-brightness.')
69         client.publish("iot-brightness", value)
70         value = random.randint(15, 40)
71         print(f'Publishing {value} to iot-temperature.')
72         client.publish("iot-temperature", value)
73         timer = 10
74     timer = timer - 1
75     time.sleep(1)
76
77 # Another option is to pump the message loop yourself by periodically calling
78 # the client loop function. Notice how the loop below changes to call loop.

```

```

Subscribe thành công.....
Subscribe thành công.....
Subscribe thành công.....
Publishing 0 to iot-humidity.
Publishing 1785 to iot-brightness.
Publishing 38 to iot-temperature.
Feed iot-humidity received new value: 0
Feed iot-temperature received new value: 38
Feed iot-brightness received new value: 1785
Publishing 21 to iot-humidity.
Publishing 956 to iot-brightness.
Publishing 24 to iot-temperature.
Feed iot-humidity received new value: 21
Feed iot-brightness received new value: 956
Feed iot-temperature received new value: 24

```

Hình 9: Kiểm tra subscribe và publish dữ liệu

Nhận xét: sau mỗi 10s sẽ tạo ngẫu nhiên 1 khối dữ liệu rồi gửi (publish) lên server.

Khi có sự thay đổi dữ liệu trên server (thay đổi giá trị nút nhấn), chương trình sẽ tải xuống (subscribe) về local computer.

Kết quả được in trên màn hình.