

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



PHÁT TRIỂN ỨNG DỤNG INTERNET OF THINGS (CO3038)

**LAB 1 : INTRODUCE TO ADAFRUIT PLATFORM
CONNECT TO ADAFRUIT USING PYTHON ON LINUX ENVIRONMENT**

SV thực hiện:

Nguyễn Trọng Tín

2012215

Contents

1	Giới thiệu	2
1.1	ADAFRUIT	2
1.2	MQTT	2
1.3	PYTHON	3
1.4	LINUX	4
1.5	VSCODE	5
2	Hiện thực	6
2.1	Kiến trúc hệ thống	6
2.2	Adafruit IO	7
2.3	Môi trường lập trình	8
3	REFERENCES	8

List of Figures

1	Adafruit's Logo	2
2	MQTT Logo	3
3	Python Logo	4
4	Ubuntu Logo	5
5	VSCode	5
6	Kiến trúc hệ thống	6
7	Feeds on Adafruit IO	7
8	Dashboard on Adafruit Io	7
9	VScode on Linux	8
10	Dashboard on Adafruit IO	8

1 Giới thiệu

1.1 ADAFRUIT

Adafruit là một công ty nổi tiếng trong lĩnh vực sản xuất và cung cấp các sản phẩm và giải pháp công nghệ sáng tạo, chuyên đặc biệt trong lĩnh vực Internet of Things (IoT) và điện tử sáng tạo.

Adafruit cung cấp một loạt các sản phẩm bao gồm các mô-đun và linh kiện điện tử, các kit phát triển, và các bảng mạch đa dạng để hỗ trợ những người sáng tạo, lập trình viên, và những người muốn thám hiểm thế giới của công nghệ. Công ty này còn nổi tiếng với việc cung cấp tài liệu hướng dẫn rõ ràng và chất lượng cao, giúp cộng đồng tận dụng tối đa các sản phẩm của họ.

Adafruit không chỉ tập trung vào việc cung cấp sản phẩm mà còn thúc đẩy cộng đồng thông qua việc hỗ trợ các dự án mã nguồn mở và chia sẻ kiến thức thông qua các bài hướng dẫn và video giáo dục. Điều này đã làm cho Adafruit trở thành một nguồn tài nguyên quan trọng và trung tâm cộng đồng cho những người đam mê điện tử và IoT.



Figure 1: Adafruit's Logo

Hơn thế nữa, Adafruit cung cấp một nền tảng IoT mạnh mẽ, **Adafruit IO**, mang lại giải pháp đơn giản để kết nối, giám sát và kiểm soát các thiết bị từ xa. Với Adafruit IO, bạn có thể dễ dàng tạo ra ứng dụng IoT đa dạng thông qua giao diện đồ họa thân thiện và hỗ trợ nhiều loại thiết bị và giao thức. Nền tảng này cung cấp các công cụ quản lý dữ liệu và tương tác linh hoạt, làm cho việc phát triển ứng dụng IoT trở nên linh hoạt và hiệu quả.

Mỗi người dùng sẽ được cấp một tài khoản, được xác minh bằng **USERNAME** và **KEY**. Dữ liệu sẽ được lưu trữ tại các **FEEDS**. Thông qua **DASHBOARD**, chúng ta có thể trực quan hóa dữ liệu, giám sát, cũng như điều khiển từ xa các thiết bị. Ngoài ra, Adafruit IO còn hỗ trợ các **ACTION** có các chức năng như **REACT** lại các tín hiệu gửi lên, ví dụ như gửi email, thông báo qua Telegram, **SCHEDULED** và **TIMER** các hành động được thiết lập trước.

1.2 MQTT

MQTT, Message Queuing Telemetry Transport, là một giao thức truyền thông nhẹ được phát triển để hỗ trợ việc truyền tải dữ liệu giữa các thiết bị trong môi trường Internet of Things (IoT). Giao thức

này tập trung vào hiệu suất, tiết kiệm băng thông, và độ tin cậy.

Publish (Xuất bản): Là quá trình mà một thiết bị gửi thông điệp (message) đến một địa chỉ trên broker (một máy chủ MQTT). Thông điệp này có thể là dữ liệu cảm biến, trạng thái thiết bị, hoặc bất kỳ thông tin nào khác. Cụ thể khi sử dụng platform Adafruit IO, chúng ta sẽ publish lên các Feed.

Subscribe (Đăng ký): Là quá trình mà một thiết bị yêu cầu nhận thông điệp từ một địa chỉ cụ thể trên broker. Khi thông điệp được xuất bản đến địa chỉ đó, thiết bị đăng ký sẽ nhận và xử lý thông điệp đó. Chúng ta có thể Subscribe vào các Feed trên Adafruit IO.

MQTT tạo ra một mô hình giao tiếp linh hoạt giữa các thiết bị, cho phép chúng xuất bản và đăng ký thông tin một cách dễ dàng, tối ưu hóa quá trình truyền thông trong mạng IoT.



Figure 2: MQTT Logo

1.3 PYTHON

Python là một ngôn ngữ lập trình bậc cao, đa năng, dễ học và có thể chạy trên nhiều nền tảng khác nhau. Python được sử dụng rộng rãi trong các lĩnh vực như phát triển web, phân tích dữ liệu, học máy và trí tuệ nhân tạo, thậm chí là phát triển Hệ thống Nhúng, IoT...

Python ra mắt lần đầu tiên vào năm 1991 và đã có nhiều phiên bản cải tiến kể từ đó. Hiện nay, Python có hai phiên bản chính là Python 2 và Python 3, trong đó Python 3 là phiên bản mới nhất và được khuyến khích sử dụng với các ưu điểm như là:

- **Dễ Đọc và Viết:** Python có cú pháp rõ ràng và gần với ngôn ngữ tự nhiên, giúp người lập trình viết mã một cách nhanh chóng và dễ đọc.
- **Đa Nhiệm và Linh Hoạt:** Python hỗ trợ đa nhiệm và có thư viện đa dạng, điều này làm cho nó linh hoạt và phù hợp với nhiều ứng dụng khác nhau.
- **Cộng Đồng Lớn:** Python có một cộng đồng phát triển đông đảo, với nhiều thư viện và framework hỗ trợ, giúp giảm độ phức tạp của việc phát triển phần mềm.
- **Hỗ Trợ Đa Nền Tảng:** Python có thể chạy trên nhiều hệ điều hành và kiến trúc, từ máy tính cá nhân đến thiết bị nhúng.

- Thư Viện Đa Dạng: Python có nhiều thư viện và framework phổ biến như Flask, Django, và Twisted, giúp xây dựng các ứng dụng nhúng một cách dễ dàng.
- Hỗ Trợ MQTT và IoT: Python có các thư viện như Paho MQTT cho việc triển khai các giao thức IoT như MQTT, làm cho nó trở thành một lựa chọn tốt cho các dự án gateway nhúng kết nối thiết bị IoT.

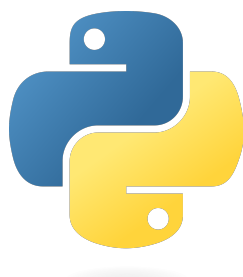


Figure 3: *Python Logo*

Python chủ yếu được sử dụng để phát triển các chương trình nhúng trong các gateway ví dụ như:

- Raspberry Pi: Raspberry Pi là một máy tính nhúng rất phổ biến và thường sử dụng hệ điều hành Raspbian, một phiên bản tùy chỉnh của Linux có sẵn sẽ hỗ trợ Python.
- Arduino (với MicroPython): Arduino thường sử dụng C/C++, nhưng có một phiên bản MicroPython được phát triển dành cho môi trường nhúng, cho phép bạn sử dụng Python trực tiếp.
- Jetson Nano: Jetson Nano của NVIDIA là một mô-đun máy tính nhúng mạnh mẽ, có thể chạy Python và hỗ trợ nhiều công nghệ AI.
- Orange Pi: Một loạt các bo mạch nhúng của Orange Pi chạy trên nền tảng Linux và hỗ trợ Python.

Đa số, các gateway, máy tính nhúng sẽ sử dụng hệ điều hành Linux, một nền tảng hệ điều hành mở, được phát triển rộng rãi sẽ được giới thiệu sau đây.

1.4 LINUX

Linux là một hệ điều hành tương tự Unix, dựa trên hạt nhân Linux, được phát triển bởi Linus Torvalds vào năm 1991. Linux là một hệ điều hành mã nguồn mở, có nghĩa là có thể tải xuống, sửa đổi và phân phối mã nguồn của nó theo các điều khoản của giấy phép GNU General Public License. Linux có thể chạy trên nhiều nền tảng khác nhau, từ máy tính cá nhân, máy chủ, điện thoại thông minh, đến các thiết bị nhúng.

Linux thường được sử dụng trong các thiết bị nhúng vì nhiều lý do, chẳng hạn như:

- Linux có chi phí thấp và dễ tùy chỉnh. Có thể chọn các thành phần phù hợp với nhu cầu và tài nguyên của thiết bị nhúng, và loại bỏ những thứ không cần thiết. Ngoài ra, cho phép sửa đổi mã nguồn để tối ưu hóa hiệu năng và bảo mật.
- Linux là môi trường lý tưởng cho các công nghệ mã nguồn mở. Dễ dàng tận dụng các thư viện, công cụ và ứng dụng mã nguồn mở sẵn có để phát triển các chức năng cho thiết bị nhúng. Ưu tiên từ sự hỗ trợ và cộng tác của cộng đồng mã nguồn mở.
- Linux có ứng dụng CLI (command-line interface) mạnh mẽ. Sử dụng các lệnh để điều khiển và quản lý thiết bị nhúng một cách nhanh chóng và hiệu quả. Cũng có thể viết các script để tự động hóa các tác vụ thường xuyên.

Trong học phần thí nghiệm này, tôi lựa chọn Hệ điều hành Ubuntu, một bản phân phối ổn định và phổ biến của Linux.



Figure 4: *Ubuntu Logo*

1.5 VSCODE

VScode là một trình biên tập mã nguồn có thể sử dụng với nhiều ngôn ngữ lập trình khác nhau, bao gồm C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust và Julia.



Figure 5: *VSCode*

VScode có nhiều tính năng nổi bật, chẳng hạn như:

- IntelliSense: Hỗ trợ hoàn thiện thông minh dựa trên các loại biến, định nghĩa hàm và các mô-đun được nhập.
- Chạy và gỡ lỗi: Cho phép gỡ lỗi mã nguồn ngay từ trình biên tập. Khởi chạy hoặc kết nối với các ứng dụng đang chạy và gỡ lỗi với các điểm dừng, ngăn xếp cuộc gọi và bảng điều khiển tương tác.
- Git tích hợp: Hỗ trợ các lệnh Git cơ bản đến nâng cao.

- Mở rộng và tùy biến: Có thể cài đặt các tiện ích mở rộng để thêm các ngôn ngữ, chủ đề, trình gỡ lỗi và kết nối với các dịch vụ bổ sung. Các tiện ích mở rộng chạy trong các quy trình riêng biệt, đảm bảo chúng sẽ không làm chậm trình biên tập.
- VScode là miễn phí và có sẵn trên nền tảng phổ biến - Linux, macOS và Windows.

2 Hiện thực

2.1 Kiến trúc hệ thống

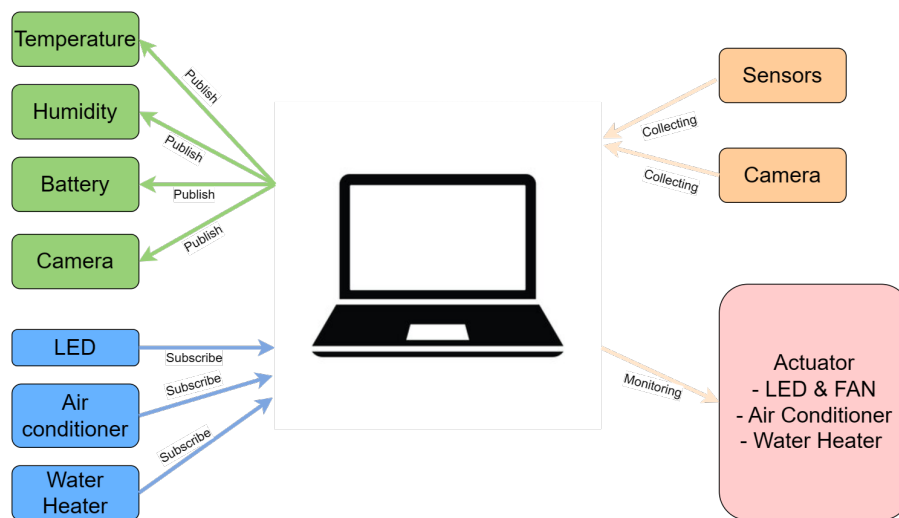


Figure 6: Kiến trúc hệ thống

Kiến trúc hệ thống tổng quát như hình trên, trong đó PC, Laptop, hoặc một máy tính nhúng sẽ hoạt động như 1 gateway chạy một chương trình Python.

Phía Platform Adafruit, bao gồm 7 FEEDS. Trong đó, 4 FEEDS dùng để **lưu trữ dữ liệu quan trắc**: Nhiệt độ, độ ẩm, dung lượng Pin, và Hình ảnh từ Camera. 3 FEEDS còn lại dùng để **điều khiển các thiết bị**: Đèn và quạt, Máy điều hòa, Máy nước nóng.

Gateway sẽ **thu thập dữ liệu từ các Sensors**, sau đó **publish** lên 4 Feeds quan trắc tương ứng. Người dùng sẽ sử dụng **Dashboard** để tạo dữ liệu điều khiển xuống 3 Feeds điều khiển, Gateway sẽ **subscribe** vào các Feeds này để **nhận tín hiệu và điều khiển các thiết bị** tương ứng.

2.2 Adafruit IO

Feed Name	Key	Last value	Recorded
<input type="checkbox"/> Air-conditioner	air-conditioner	20	about 18 hours ago
<input type="checkbox"/> Battery	battery	54.2	about 13 hours ago
<input type="checkbox"/> Camera	camera		1 day ago
<input type="checkbox"/> Humi	humi	66.51	about 13 hours ago
<input type="checkbox"/> LED	led	0	about 19 hours ago
<input type="checkbox"/> Temp	temp	24.83	about 13 hours ago
<input type="checkbox"/> Water-Heater	water-heater	1	about 18 hours ago

Figure 7: Feeds on Adafruit IO

Bao gồm 7 FEEDS được tạo trên Adafruit IO, trong đó Temperature (Nhiệt độ), Humidity (Độ ẩm) và Dung lượng Pin (Battery) sẽ được cập nhật theo thứ tự mỗi 5 giây. Chỉ số Nhiệt độ sẽ dao động quanh Nhiệt độ được điều chỉnh bởi Air-conditioner (Máy điều hòa). Dung lượng Pin sẽ giảm nhanh hoặc chậm dựa vào việc LED và Máy nước nóng có mở hay không.

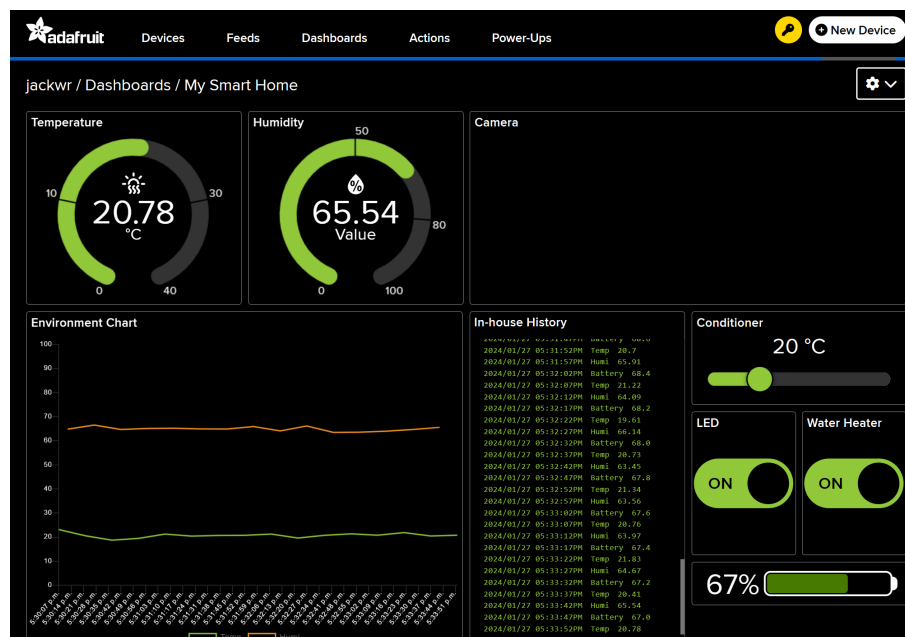


Figure 8: Dashboard on Adafruit Io

Dashboard trực quan trên Adafruit IO, trong đó Nhiệt độ (ổn định trong khoảng 10 - 30 °C) và Độ ẩm (ổn định trong khoảng 50 - 80 %) được theo dõi trên Gauge chart. Line chart dùng để theo dõi trực quan lịch sử dao động của Nhiệt độ và độ ẩm trong 1 giờ gần nhất. Ngoài ra, còn có bảng Lịch sử giá trị tất cả 7 Feeds trong 1 giờ gần nhất, và Dung lượng Pin.

Có thể điều khiển máy điều hòa trong khoảng 16 đến 30 °C bằng slider, công tắc bật tắt Đèn và Máy nước nóng

2.3 Môi trường lập trình

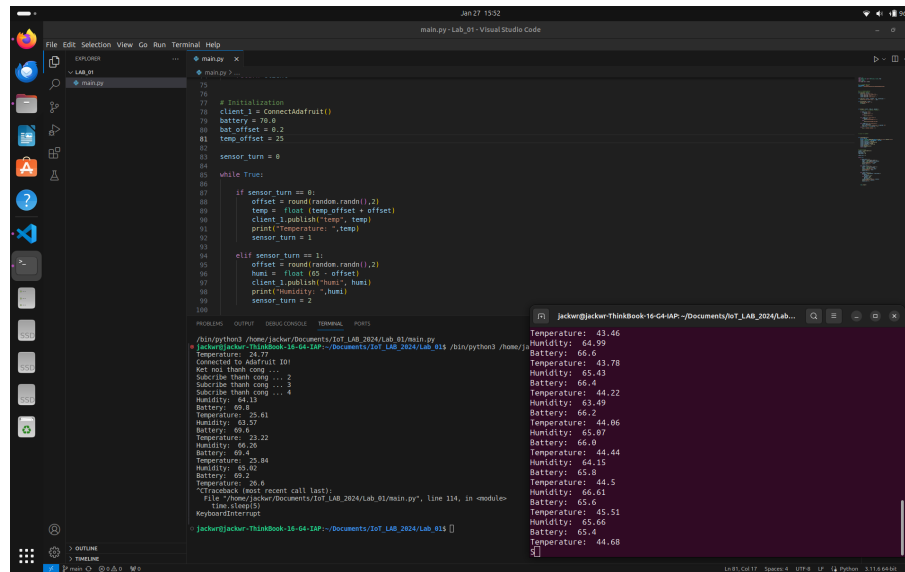


Figure 9: VScode on Linux

Lập trình bằng VScode Editor, môi trường Python 3.11, trong bản phân phối Ubuntu 22 dùng kernel Linux.

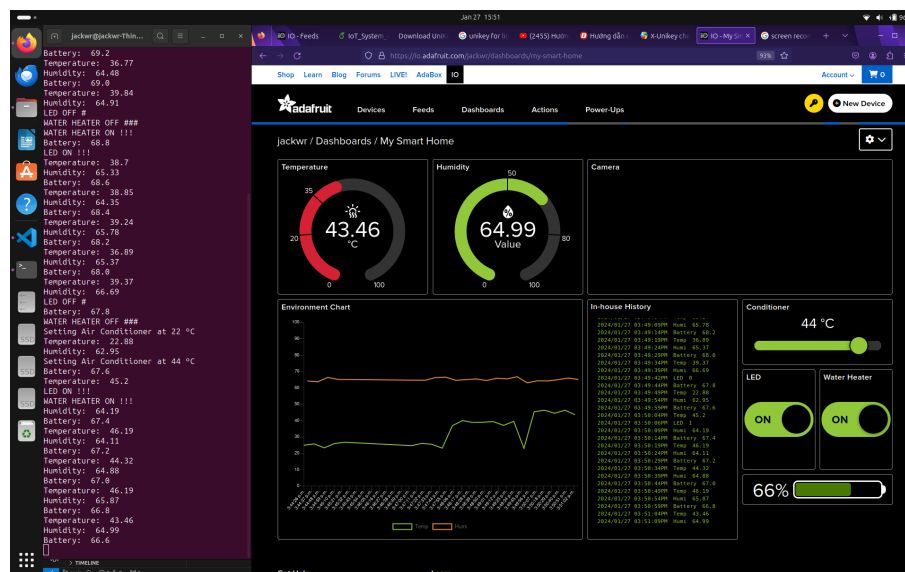


Figure 10: Dashboard on Adafruit IO

3 REFERENCES

Video Demo upload here: <https://youtu.be/UySeJY34C8g>

Please check my github for more information about source code and building problems

https://github.com/JackWrion/IoT_LAB_2024/tree/main/Lab_01