

VIET NAM NATIONAL UNIVERSITY HO CHI MINH CITY  
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY  
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



Thực tập đồ án đa ngành

---

Báo cáo Đồ án đa ngành

# Dự án hệ thống cạnh tác thông minh

---

Advisor: Th.S Mai Đức Trung

HO CHI MINH CITY, JANUARY 2024



## Danh sách thành viên & Đóng góp

No.	Họ và Tên	MSSV	Đóng góp
1	Nguyễn Công Minh	1752347	25%
2	Trương Huỳnh Phát	2033076	25%
3	Nguyễn Duy Tân	1713069	25%
4	Nguyễn Quang Minh	1813088	25%



## Contents

<b>1</b>	<b>Giới thiệu</b>	<b>4</b>
<b>2</b>	<b>Tổng quan hệ thống</b>	<b>4</b>
2.1	Cảm biến nhiệt độ và độ ẩm đất . . . . .	4
2.2	Cảm biến ánh sáng . . . . .	4
2.3	Hệ thống Relay và máy bơm . . . . .	4
2.4	Hệ thống đèn LED chiếu sáng . . . . .	5
<b>3</b>	<b>Các công nghệ sử dụng</b>	<b>5</b>
3.1	Adafruit IO . . . . .	5
3.2	Python MQTT Gateway/PyCharm . . . . .	6
3.3	Machine Learning . . . . .	7
<b>4</b>	<b>Thiết kế hệ thống</b>	<b>8</b>
4.1	Yêu cầu chức năng . . . . .	8
4.2	Yêu cầu phi chức năng . . . . .	8
4.3	Lược đồ Use-case . . . . .	9
4.4	Đặc tả Use-case . . . . .	10
4.4.1	Điều khiển máy hệ thống đèn LED . . . . .	10
4.4.2	Điều khiển máy bơm nước . . . . .	10
4.4.3	Ghi dữ liệu lên hệ thống . . . . .	11
4.4.4	Tạo lịch sử dữ liệu . . . . .	11
4.4.5	Hiển thị dữ liệu . . . . .	11
4.4.6	Kiểm tra dữ liệu và đưa ra cảnh báo . . . . .	12
4.5	Sơ đồ hoạt động . . . . .	12
4.5.1	Bật tắt hệ thống đèn và máy bơm . . . . .	12
4.5.2	Ghi dữ liệu lên hệ thống . . . . .	13
4.5.3	Hiển thị dữ liệu trên hệ thống . . . . .	13
4.5.4	Kiểm tra dữ liệu và gửi cảnh báo . . . . .	14
<b>5</b>	<b>Hiện thực hệ thống</b>	<b>14</b>
5.1	Tổng quan hệ thống . . . . .	14
5.2	IoT Gateway . . . . .	15
5.3	COM Serial Simulation . . . . .	16
5.4	Adafruit Dashboard . . . . .	17
5.5	Machine Learning . . . . .	18
<b>6</b>	<b>Kết luận</b>	<b>19</b>

## 1 Giới thiệu

Nông nghiệp luôn là một trong những thế mạnh của đất nước ta, với sản lượng nông nghiệp hàng năm, chúng ta luôn dẫn đầu trong khu vực về khối lượng nông nghiệp xuất khẩu cũng như tiêu thụ trong và ngoài nước. Hơn nữa ngày nay với thời đại công nghệ thông tin đang ngày càng phát triển, càng nhiều ngành nghề đang tìm những giải pháp số hóa cũng như tự động hóa nhằm cải tiến sản lượng cũng như là giảm thiểu khối lượng công việc lao động của con người, cho nên với dự án canh tác thông minh, nhóm chúng em xin đem đến một giải pháp hệ thống canh tác thông minh có thể giúp người nông dân có thể kiểm soát cũng như là theo dõi những điều kiện canh tác một cách dễ dàng nhằm việc tạo ra nhiều sự thuận lợi cho công việc chăm sóc cây trồng, hơn nữa quy mô của giải này không chỉ dừng lại ở việc canh tác trong nông nghiệp, chúng ta cũng có thể ứng dụng trong cả những hộ gia đình có nhu cầu canh tác tại gia những loại rau sạch hoặc những cây cảnh một cách tiện lợi mà không cần tiêu tốn nhiều thời gian.

## 2 Tổng quan hệ thống

Sau quá trình xem xét cũng như tìm hiểu chúng em quyết định sẽ sử dụng những thiết bị như sau để phục vụ cho việc hiện thực hệ thống.

### 2.1 Cảm biến nhiệt độ và độ ẩm đất

Độ ẩm cũng như là nhiệt độ là một trong những yếu tố quyết định đến năng suất cũng như khả năng sinh trưởng của cây trồng rất nhiều, cho nên việc sử dụng cảm biến để theo dõi độ ẩm và nhiệt độ cần phải chính xác cũng như là kịp thời. Theo như chúng em khảo sát cũng như là nghiên cứu thì độ ẩm tối thiểu đất để cây trồng có thể phát triển ổn định sẽ vào khoảng từ 50 - 60% và nhiệt độ không được vượt quá 30 độ C.

### 2.2 Cảm biến ánh sáng

Trong điều kiện những loại cây trồng khác nhau sẽ có nhu cầu chiếu sáng khác nhau, ví dụ như những loại cây trồng ưa nắng có thể cần đến cường độ ánh sáng lên tới 20.000 lux, còn những loại cây trồng ưa bóng râm thì cường độ ánh sáng cần thiết chỉ rơi vào khoảng từ 5.000 - 7.000 lux cho nên việc theo dõi cường độ ánh sáng để phù hợp với nhu cầu chiếu sáng của từng loại cây trồng là rất quan trọng.

### 2.3 Hệ thống Relay và máy bơm

Việc bật tắt từng máy bơm có thể bất tiện cũng như là khó kiểm soát, nên ở đây chúng em sẽ kiểm soát hệ thống máy bơm qua một relay, việc sử dụng relay cũng khiến cho hệ thống trở nên an toàn và tiện dụng hơn trong việc kiểm soát cũng như là sửa chữa trong trường hợp có sự cố xảy ra.

## 2.4 Hệ thống đèn LED chiếu sáng

Hệ thống đèn chiếu sáng giúp hỗ trợ tăng khả năng chiếu sáng để đảm bảo việc cung cấp đầy đủ ánh sáng cho cây quang hợp một cách thuận lợi, việc sử dụng LED thay cho bóng đèn truyền thống cũng mang lại độ hiệu quả cao vì đèn LED có hiệu suất hoạt động cao, tiết kiệm điện cũng như ít xảy ra tình trạng hao phí so với việc sử dụng đèn huỳnh quang hay đèn dây tóc truyền thống. Nhược điểm duy nhất của việc sử dụng đèn LED là chi phí đầu tư ban đầu có thể cao hơn so với các giải pháp chiếu sáng khác.

## 3 Các công nghệ sử dụng

Việc tìm hiểu cũng như hiểu biết về các công nghệ được sử dụng trong một dự án rất quan trọng vì nó cung cấp cho người dùng cũng như người vận hành những thông tin cần thiết để có thể áp dụng cũng như vận hành hệ thống một cách trơn tru mang lại hiệu quả cao hơn cũng như là để khắc phục những biến cố phát sinh trong quá trình vận hành và sử dụng.

### 3.1 Adafruit IO

Adafruit IO là một dịch vụ Internet of Things (IoT) do Adafruit Industries cung cấp. Adafruit Industries là một công ty chuyên về thiết kế và sản xuất các sản phẩm điện tử và phụ kiện để hỗ trợ cộng đồng sáng tạo và những người làm thủ công.

Adafruit IO giúp kết nối các thiết bị IoT với Internet, cho phép chúng ta theo dõi và điều khiển chúng từ bất kỳ đâu trên thế giới thông qua mạng Internet. Dịch vụ này được xây dựng trên nền tảng đám mây và cung cấp một loạt các tính năng hữu ích cho phát triển ứng dụng IoT. Adafruit cung cấp cho người dùng những tiện ích được tích hợp trên nền tảng như sau:

- **Feeds (Nguồn):** Adafruit IO sử dụng khái niệm "feeds" để lưu trữ và quản lý dữ liệu từ các thiết bị IoT của bạn. Mỗi feed là một dòng dữ liệu theo dõi một loại thông tin cụ thể.
- **Dashboards (Bảng điều khiển):** Chúng ta có thể tạo các bảng điều khiển tùy chỉnh để theo dõi và kiểm soát các thiết bị của mình một cách thuận tiện. Bảng điều khiển này giúp chúng ta tổ chức thông tin và thao tác với thiết bị một cách dễ dàng.
- **Triggers (Kích hoạt):** Adafruit IO hỗ trợ triggers, giúp chúng ta định cấu hình các sự kiện tự động dựa trên dữ liệu từ thiết bị. Khi điều kiện được đáp ứng, triggers có thể kích hoạt các hành động xác định.
- **API (Giao diện lập trình ứng dụng):** Adafruit IO cung cấp một API mạnh mẽ, cho phép bạn tương tác với dịch vụ từ các ứng dụng tự xây dựng của mình. Điều này mở ra nhiều cơ hội để tích hợp Adafruit IO với các dự án và ứng dụng khác.
- **Community Support (Hỗ trợ từ cộng đồng):** Adafruit có một cộng đồng mạnh mẽ với các diễn đàn, tài liệu và hướng dẫn giúp đỡ người dùng trong việc sử dụng Adafruit IO và các sản phẩm khác của họ.

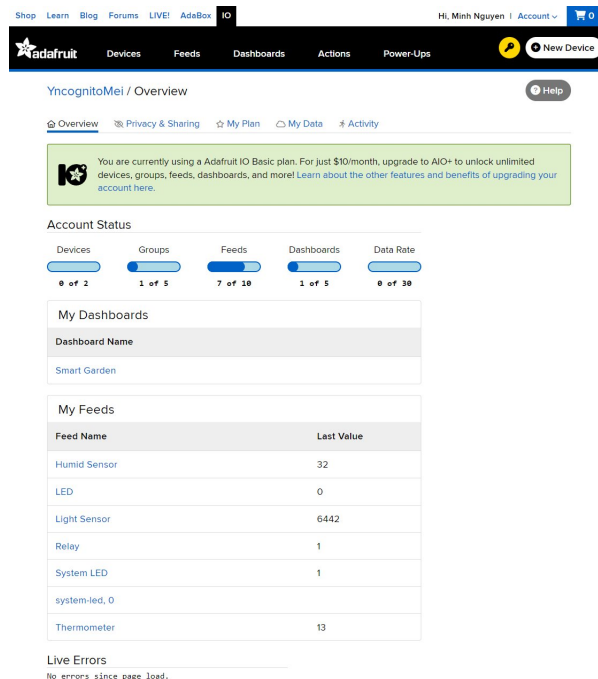


Figure 1: AdafruitIO

### 3.2 Python MQTT Gateway/PyCharm

MQTT là viết tắt của Message Queuing Telemetry Transport, một giao thức dùng để kết nối giữa các thiết bị IoT với server, MQTT Gateway giúp thu thập dữ liệu và truyền tải thông tin giữa các thiết bị trong hệ thống.

Toàn bộ dự án này được chúng em lập trình bằng ngôn ngữ lập trình Python sử dụng PyCharm một nền tảng môi trường tích hợp lập trình cho ngôn ngữ Python được phát triển bởi JetBrains. Được thiết kế để cung cấp một loạt các tính năng hữu ích để hỗ trợ các nhà phát triển Python trong quá trình phát triển ứng dụng Python. Một số tính năng nổi trội của PyCharm cung cấp như:

- **Editor Nâng Cao:** PyCharm có một trình soạn thảo mã nguồn mạnh mẽ với nhiều tính năng như kiểm tra lỗi cú pháp, hoàn thành mã tự động, refactoring, và hỗ trợ cho nhiều ngôn ngữ lập trình liên quan đến Python.
- **Hỗ Trợ Dự Án Lớn:** PyCharm hỗ trợ phát triển ứng dụng Python với kích thước lớn, bao gồm cả hệ thống quản lý phiên bản như Git và Mercurial. Nó cung cấp khả năng làm việc hiệu quả với các dự án lớn và phức tạp.
- **Debug Console:** PyCharm cung cấp công cụ debugging mạnh mẽ với khả năng tìm kiếm lỗi, theo dõi biến, và xem các biểu đồ giúp hiểu rõ hơn về quá trình thực thi của ứng dụng.
- **Unit Testing:** Nó tích hợp với các framework kiểm thử như pytest và unittest, giúp người phát triển viết

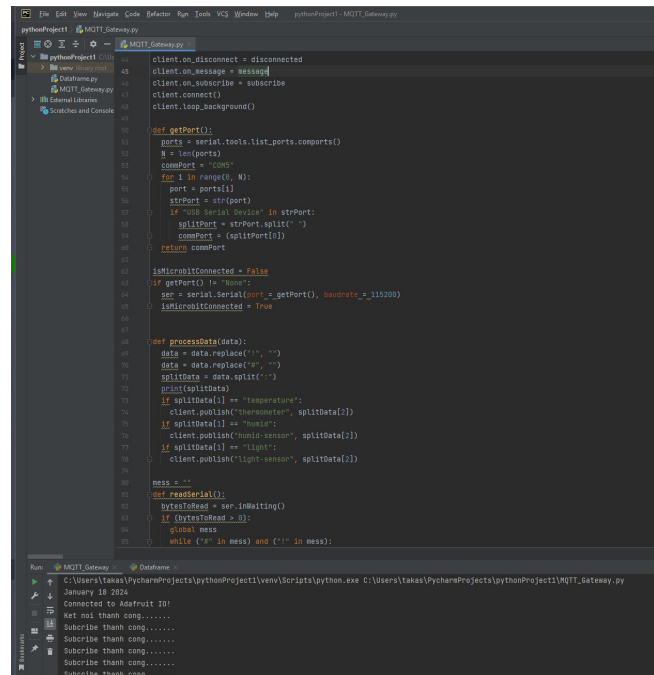


Figure 2: PyCharm IDE

và chạy các bài kiểm thử đơn vị một cách thuận tiện.

- **Quản lý Môi Trường Ứng Dụng:** PyCharm hỗ trợ quản lý môi trường ảo và tích hợp với các trình quản lý gói như pip, giúp tạo và duy trì môi trường phát triển riêng biệt cho từng dự án.
- **Tích Hợp Nâng Cao:** Cung cấp tích hợp với các công cụ như IPython, Anaconda, Jupyter Notebook, Django, Flask, và nhiều framework và thư viện Python khác.
- **Cộng Đồng và Hỗ Trợ:** PyCharm có một cộng đồng sôi động, và JetBrains cung cấp hỗ trợ chất lượng cao thông qua diễn đàn và tài liệu trực tuyến.

### 3.3 Machine Learning

Machine Learning hay Học máy là một lĩnh vực của trí tuệ nhân tạo (AI) mà nói chung nghiên cứu và xây dựng các thuật toán và mô hình có khả năng tự học và cải thiện từ dữ liệu. Mục tiêu của Machine Learning là phát triển các phương pháp để máy tính có thể tự động học hỏi từ dữ liệu mà không cần được lập trình một cách tường minh. Việc ứng dụng Machine Learning có thể giúp chúng ta đưa ra những phương án tối ưu cũng như những kế hoạch cạnh tranh phù hợp với từng thời điểm mà không cần tới sự can thiệp trực tiếp của con người.

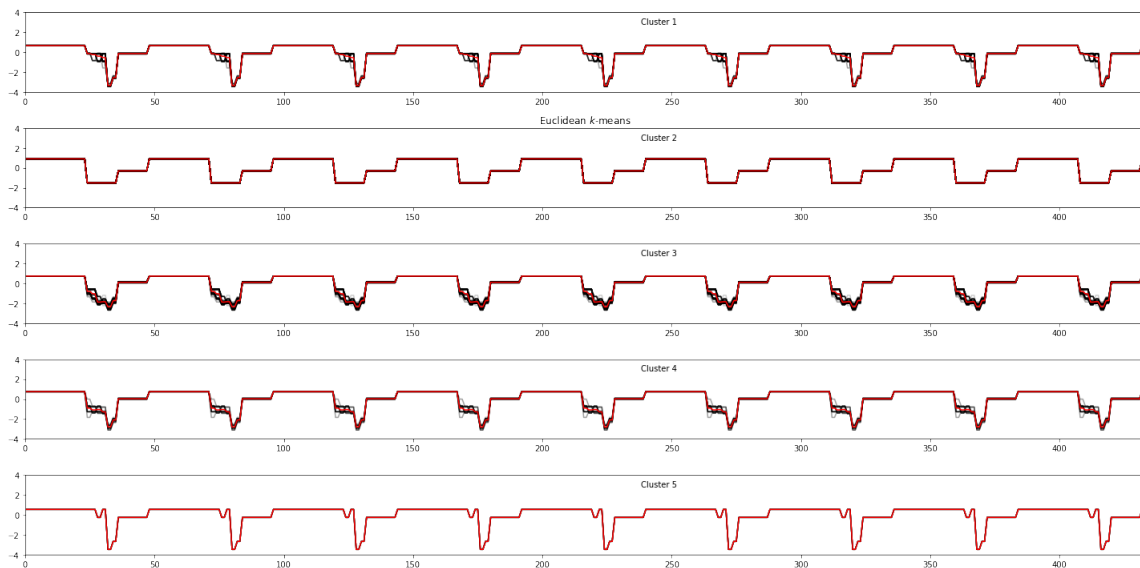


Figure 3: K-means Clustering, một thuật toán phân cụm trong Học máy

## 4 Thiết kế hệ thống

### 4.1 Yêu cầu chức năng

Hệ thống có những yêu cầu về chức năng như sau đây:

- **Cập nhật và hiển thị thông tin cảm biến nhiệt độ, độ ẩm, ánh sáng:** Những dữ liệu cảm biến đọc được phải được ghi nhận trực tiếp lên hệ thống và được thể hiện thông qua các dashboard trên server.
- **Kiểm tra lịch sử hệ thống:** Những dữ liệu lịch sử phải được lưu lại trên hệ thống để kiểm tra cũng như là được sử dụng cho mục đích nghiên cứu.
- **Cảnh báo về các dữ kiện:** Khi những giá trị chạm mức cảnh báo (được đặt ra sẵn) hệ thống sẽ hiển thị cảnh báo.
- **Bật tắt máy bơm, đèn tự động dựa trên dữ liệu thu thập:** Máy bơm và đèn sẽ tự động bật khi hệ thống cảm biến ghi nhận dữ liệu chạm ngưỡng kích hoạt (trigger threshold).
- **Bật tắt máy bơm, đèn thủ công:** Người dùng có thể bật tắt thủ công các thiết bị bằng nút bấm trên hệ thống.

### 4.2 Yêu cầu phi chức năng

- Hệ thống phải ghi nhận dữ liệu theo thời gian thực.
- Các thao tác trên hệ thống có độ trễ thấp.



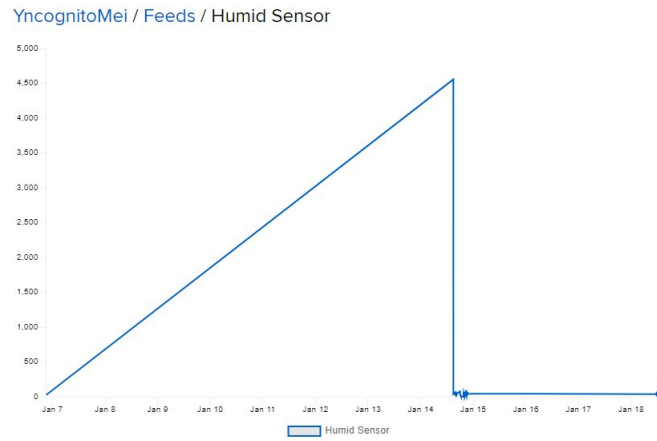
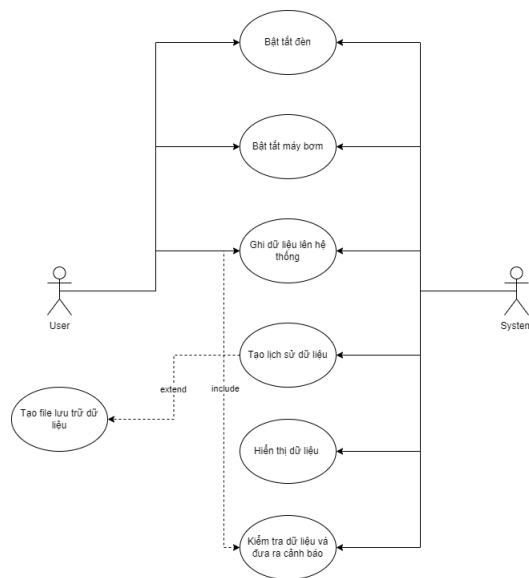


Figure 4: Dữ liệu độ ẩm theo thời gian thực

- Giao diện thân thiện với người dùng.
- Tính ổn định cao

### 4.3 Lược đồ Use-case



## 4.4 Đặc tả Use-case

### 4.4.1 Điều khiển máy hệ thống đèn LED

Tên Use-case	Bật tắt đèn
Chủ thể	Người dùng, Hệ thống
Mô tả	Bật tắt hệ thống đèn LED
Tiền điều kiện	Giá trị cường độ ánh sáng $\leq 1.000(lux)$
Kích hoạt	Hệ thống hoặc người dùng bật công tắc
Điều hướng cơ bản	1. Cường độ ánh sáng $\leq 1.000(lux)$ 2. Hệ thống nhận dữ liệu từ cảm biến 3. Hệ thống bật công tắc đèn
Điều hướng khác	Người dùng bật công tắc đèn
Hậu điều kiện	Hệ thống đèn bật và tự động tắt khi cường độ ánh sáng $> 10.000 (lux)$
Ngoài lệ	None
Ghi chú	None

### 4.4.2 Điều khiển máy bơm nước

Tên Use-case	Bật tắt máy bơm nước
Chủ thể	Người dùng, Hệ thống
Mô tả	Bật tắt hệ thống máy bơm
Tiền điều kiện	Giá trị độ ẩm $\leq 50\%$
Kích hoạt	Hệ thống hoặc người dùng bật công tắc
Điều hướng cơ bản	1. Độ ẩm đất $\leq 50\%$ 2. Hệ thống nhận dữ liệu từ cảm biến 3. Hệ thống bật công tắc máy bơm
Điều hướng khác	Người dùng bật công tắc máy bơm
Hậu điều kiện	Hệ thống máy bơm bật và tự động tắt khi độ ẩm $> 60\%$
Ngoài lệ	Cảnh báo người dùng bật máy bơm khi độ ẩm đất cao
Ghi chú	None

#### 4.4.3 Ghi dữ liệu lên hệ thống

Tên Use-case	Ghi dữ liệu lên hệ thống
Chủ thể	Người dùng, Hệ thống
Mô tả	Chèn và hiển thị dữ liệu đã chèn trên hệ thống
Tiền điều kiện	Người dùng được cấp quyền hoặc hệ thống ghi nhận dữ liệu từ cảm biến
Kích hoạt	Bật tắt đèn hoặc máy bơm nếu dữ liệu chèn chạm ngưỡng kích hoạt
Điều hướng cơ bản	1. Người dùng nhập dữ liệu trên feed 2. Hệ thống nhận dữ liệu từ cảm biến
Điều hướng khác	None
Hậu điều kiện	None
Ngoài lệ	None
Ghi chú	None

#### 4.4.4 Tạo lịch sử dữ liệu

Tên Use-case	Tạo lịch sử dữ liệu
Chủ thể	Hệ thống
Mô tả	Lưu trữ dữ liệu trên hệ thống
Tiền điều kiện	Dữ liệu đã được thu thập trước
Kích hoạt	None
Điều hướng cơ bản	Hệ thống lưu trữ dữ liệu cảm biến
Điều hướng khác	None
Hậu điều kiện	Lưu trữ xong dữ liệu
Ngoài lệ	Không nhận được dữ liệu gì từ cảm biến
Ghi chú	None

#### 4.4.5 Hiển thị dữ liệu

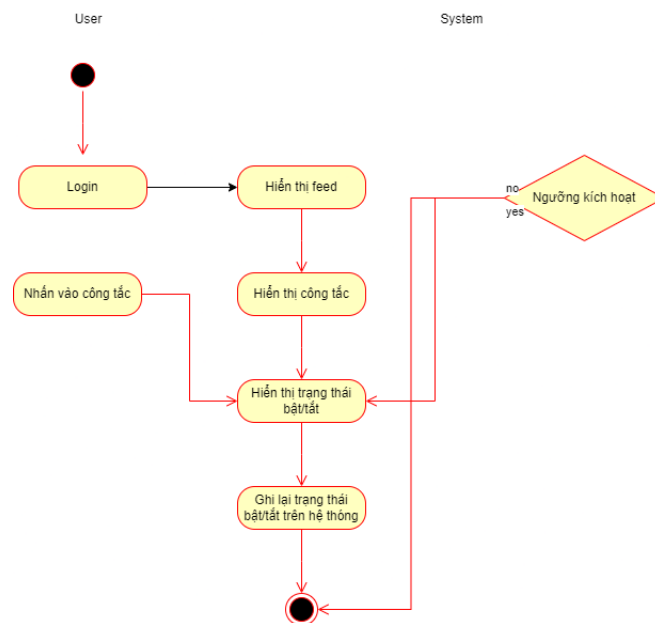
Tên Use-case	Hiển thị dữ liệu lên hệ thống
Chủ thể	Hệ thống
Mô tả	Hiển thị dữ liệu nhận được trên hệ thống
Tiền điều kiện	Dữ liệu đã được thu thập từ cảm biến
Kích hoạt	None
Điều hướng cơ bản	Hiển thị dữ liệu đã thu thập trên dashboard
Điều hướng khác	None
Hậu điều kiện	Dữ liệu được hiển thị
Ngoài lệ	Không nhận được dữ liệu gì từ cảm biến
Ghi chú	None

#### 4.4.6 Kiểm tra dữ liệu và đưa ra cảnh báo

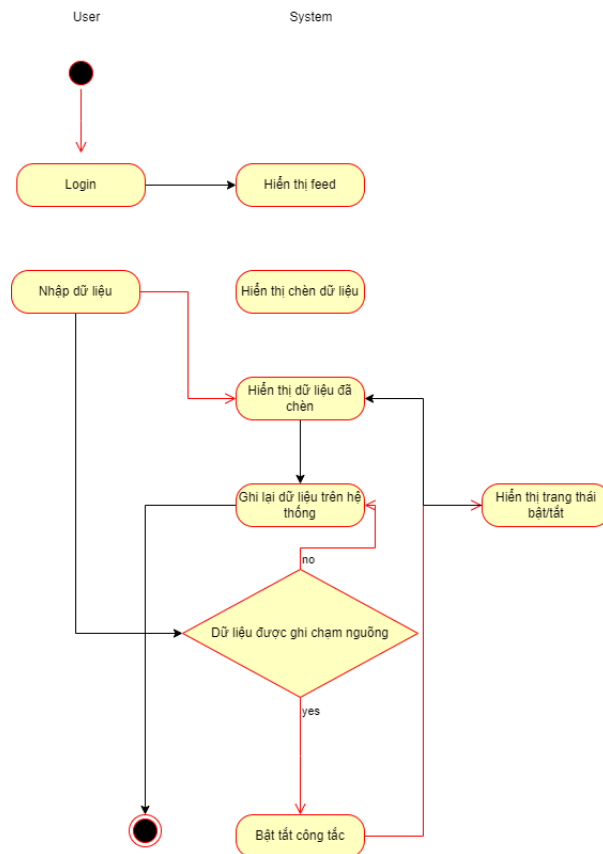
Tên Use-case	Kiểm tra dữ liệu và đưa ra cảnh báo
Chủ thể	Hệ thống
Mô tả	Đối chiếu dữ liệu nhận được trên hệ thống
Tiền điều kiện	Dữ liệu đã được thu thập từ cảm biến
Kích hoạt	None
Điều hướng cơ bản	Hiển thị dữ liệu tới ngưỡng và đưa ra cảnh báo
Điều hướng khác	None
Hậu điều kiện	Cảnh báo được hiển thị
Ngoài lệ	Không nhận được dữ liệu gì từ cảm biến hoặc dữ liệu chưa đến ngưỡng
Ghi chú	None

### 4.5 Sơ đồ hoạt động

#### 4.5.1 Bật tắt hệ thống đèn và máy bơm



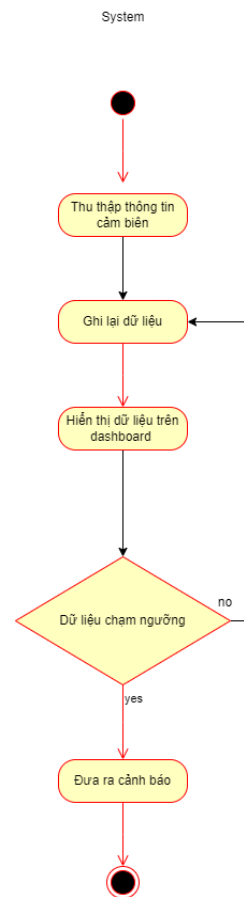
#### 4.5.2 Ghi dữ liệu lên hệ thống



#### 4.5.3 Hiển thị dữ liệu trên hệ thống

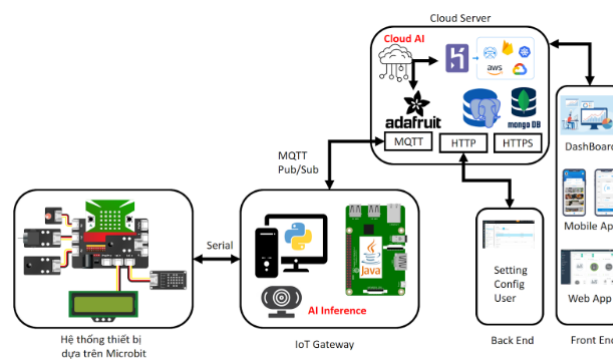


#### 4.5.4 Kiểm tra dữ liệu và gửi cảnh báo



## 5 Hiện thực hệ thống

### 5.1 Tổng quan hệ thống



## 5.2 IoT Gateway

Gateway trong IoT (Internet of Things) là hoặc phần mềm giữa thiết bị cảm biến và hệ thống mạng, giúp kết nối, gửi dữ liệu và thực hiện các chức năng xử lý trước khi dữ liệu được chuyển đến các điểm cuối, như máy chủ IoT hoặc các dịch vụ đám mây. Một IoT Gateway có nhiệm vụ chính là thu thập, gửi và quản lý dữ liệu từ các thiết bị IoT khác nhau.

Gateway cung cấp cho chúng ta những tính năng như:

- **Thu thập Dữ liệu:** IoT Gateway thu thập dữ liệu từ các thiết bị cảm biến, bộ điều khiển hoặc thiết bị IoT khác trong hệ thống, thông thường các thiết bị sẽ được kết nối với gateway thông qua cổng COM/USB, rồi qua gateway mới kết nối được với server.
- **Giao Tiếp và Kết Nối:** Nó hỗ trợ nhiều giao thức kết nối để tương tác với các thiết bị IoT. Các giao thức phổ biến bao gồm MQTT, CoAP, HTTP, và Modbus trong đề án này chúng ta sử dụng giao thức MQTT.
- **Quản Lý Thiết Bị:** IoT Gateway thường có khả năng quản lý thiết bị, bao gồm đăng ký thiết bị, quản lý thư mục, và theo dõi trạng thái của các thiết bị trong mạng.
- **Bảo mật:** Đảm bảo an toàn và bảo mật dữ liệu là một chức năng quan trọng của IoT Gateway. Nó có thể thực hiện mã hóa dữ liệu, xác thực thiết bị và người dùng, và giữ an toàn cho quá trình truyền tải dữ liệu.
- **Xử Lý Dữ Liệu Tại Chỗ (Edge Processing):** IoT Gateway thường có khả năng xử lý dữ liệu tại chỗ (edge processing), giúp giảm độ trễ và giảm lượng dữ liệu cần chuyển đến các trung tâm xử lý trung tâm.
- **Gửi Dữ Liệu Đến Điểm Cuối:** Sau khi thu thập và xử lý dữ liệu, IoT Gateway chuyển dữ liệu đến các điểm cuối như máy chủ IoT, dịch vụ đám mây hoặc hệ thống quản lý.
- **Tích Hợp và Mở Rộng:** Có thể tích hợp với các hệ thống và dịch vụ khác trong hệ sinh thái IoT. Một số IoT Gateway hỗ trợ cổng kết nối mở rộng cho các module và plugin bổ sung.

IoT Gateway đóng vai trò quan trọng trong việc tạo nên một hệ thống IoT linh hoạt và hiệu quả bằng cách giữa các thiết bị IoT và hệ thống mạng.

The screenshot displays a Jupyter Notebook environment with a Python script titled 'MQTT\_Gateway.py' and its execution results.

**Python Script (MQTT\_Gateway.py):**

```

1 import random
2 import time
3 import sys
4 import pandas as pd
5 import matplotlib.pyplot
6 import serial.tools.list_ports
7 from datetime import date
8 from Adafruit_I2C import MQTTClient
9
10 today = date.today()
11 strTextToday = today.strftime("%B %d %Y")
12 print(strTextToday)
13
14 AIO_FEED_ID_1 = "Lieu"
15 AIO_FEED_ID_2 = "Humid-sensor"
16 AIO_FEED_ID_3 = "Light-sensor"
17 AIO_FEED_ID_4 = "thermometer"
18 AIO_FEED_ID_5 = "relay"
19 AIO_USERNAME = "ynccapitolme1"
20 AIO_KEY = "aio_B1f88bvcK0SMZnc3Kz1SLTPE60n"
21
22 def connected(client):
23     print("Wat met team comp.....")
24     client.subscribe(AIO_FEED_ID_1)
25     client.subscribe(AIO_FEED_ID_2)
26     client.subscribe(AIO_FEED_ID_3)
27     client.subscribe(AIO_FEED_ID_4)
28     client.subscribe(AIO_FEED_ID_5)
29
30 def subscribe(client_userdata_mid_granted_qos):
31     print("Subscribe team comp.....")
32
33 def disconnected(client):
34     print("hugi ket noi.....")
35     sys.exit(0)
36
37 def message(client, feed_id, payload):
38     print("Nhan du lieu : " + payload)
39     if isMicrobitConnected:
40         ser.write((str(payload) + "\r").encode())
41
42 client = MQTTClient(AIO_USERNAME, AIO_KEY)

```

**Execution Output:**

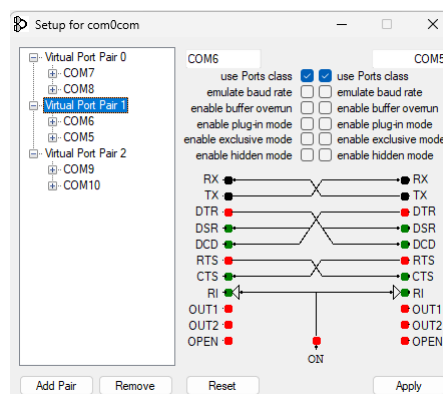
```

Run MQTT_Gateway DataFrame <
Nhan du lieu : 12
Hay bom da bat
Nhan du lieu : 1
Cap nhât do sang: 5782
Den da tat
Nhan du lieu: 5782
Nhan du lieu: 0
Cap nhât nhiet do: 3
Nhan du lieu: 3
Cap nhât do am: 52
Nhan du lieu: 52
Hay bom da tat

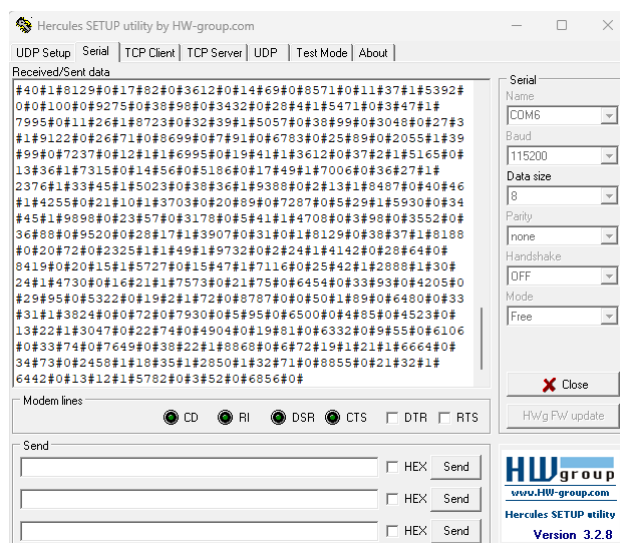
```

### 5.3 COM Serial Simulation

Việc sử dụng giả lập cổng COM ảo rất tiện lợi cho quá trình phát triển cũng như kiểm tra hệ thống, vì chúng ta không phải liên tục kết nối hoặc thiết lập những thiết bị để kiểm tra cũng như là sửa lỗi thường xuyên.// Trong dự án này chúng ta sẽ sử dụng **Hercules** và **com0com serial virtual COM** để mô phỏng hệ thống cảm biến thực tế phục vụ cho quá trình phát triển.





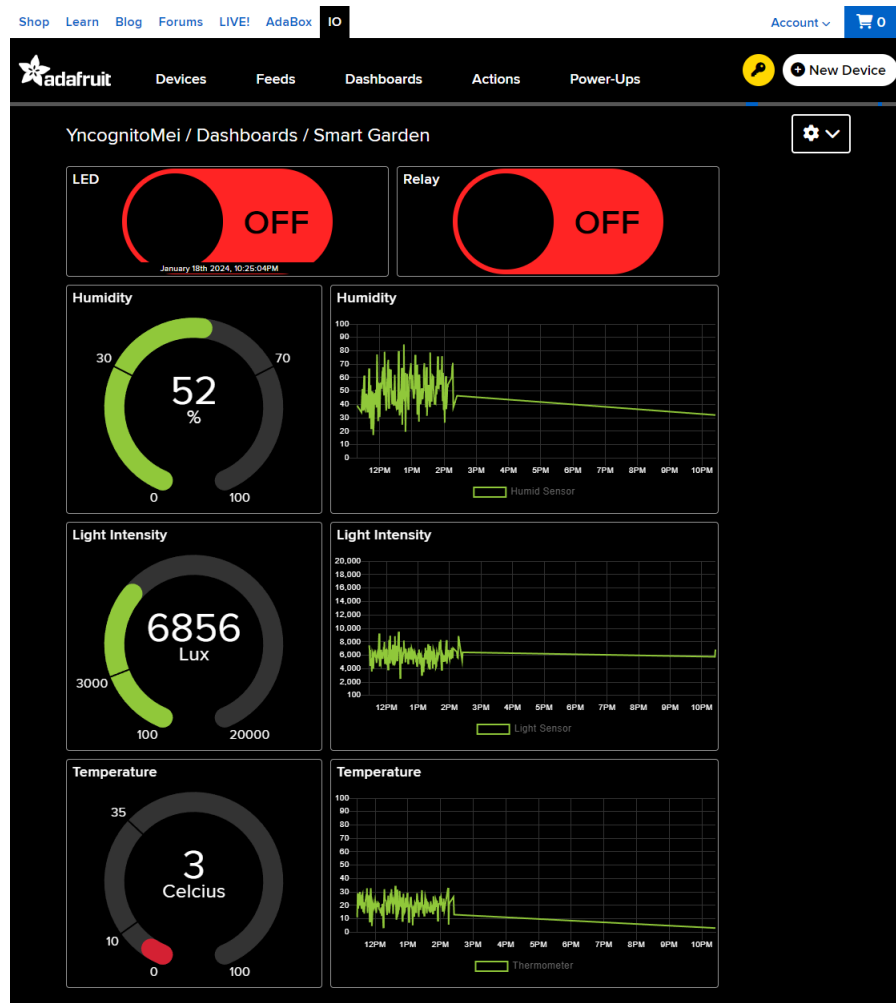


Như trên hình, phần mềm đã tự tạo giả lập ra cặp cổng COM ảo là COM5 và COM6, thiết bị của chúng ta sẽ kết nối vào cổng COM6 và cổng còn lại sẽ kết nối vào Gateway Python. Cổng COM sẽ kết nối với tốc độ 115200 (8 bits data, 1 stop bit, no parity).

Khi chúng ta gửi dữ liệu từ gateway hoặc từ server thì chương trình Hercules sẽ nhận được và sẽ được hiển thị ở trong serial console như trên hình và ngược lại.

## 5.4 Adafruit Dashboard

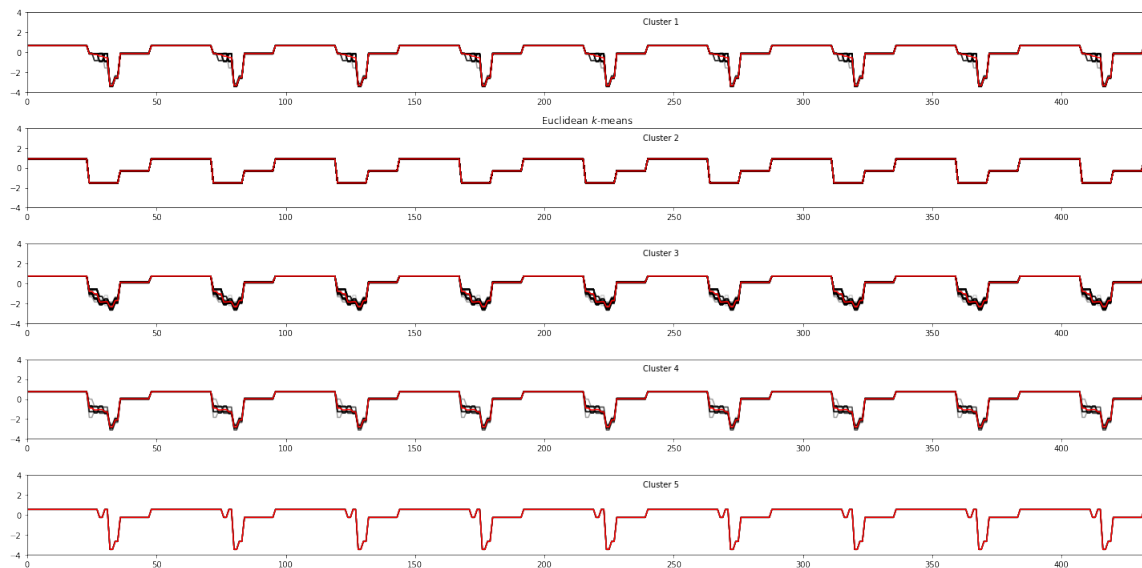
Adafruit Dashboard cho phép chúng ta trực quan hóa dữ liệu và kiểm soát các thiết bị đã kết nối. Các tiện ích như biểu đồ, thanh trượt và nút có sẵn giúp chúng ta nhanh chóng thiết lập và chạy dự án IoT mà không cần bất kỳ mã tùy chỉnh nào. Chúng ta hoàn toàn có thể tùy chỉnh dễ dàng bằng phương pháp kéo thả.



## 5.5 Machine Learning

Sử dụng Học máy để nghiên cứu những dữ liệu được ghi lại từ cảm biến có thể giúp tự động hóa vì mô hình dữ liệu có thể tự học và đưa ra những phương thức canh tác hợp lý cũng như là chính xác nhằm tăng hiệu quả của hệ thống. Trong đồ án lần này chúng em sử dụng thuật toán học không giám sát (Unsupervised-learning) là K-means Clustering nhằm phân cụm dữ liệu tại những thời điểm cụ thể hoặc những thời điểm nằm trong ngưỡng báo động. Các cụm dữ liệu phân cụm sẽ được dùng để dự đoán những thời điểm để hệ thống có thể tự động bật tắt đèn hoặc máy bơm nhằm đảm bảo độ ẩm và độ sáng cho vườn.

Nhằm đáp ứng cho nhu cầu đó, những dữ liệu trên hệ thống sẽ được ghi lại dưới dạng dataframe và lưu trữ trong hệ thống file csv.



## 6 Kết luận

Qua việc thực hiện đồ án lần này, chúng em đã học được rất nhiều kĩ năng cũng như là kinh nghiệm trong việc thiết kế, lên kế hoạch cũng như là hiện thực một hệ thống IoT. Trong tương lai chúng em sẽ tiếp tục cải tiến cũng như là tiến hành nâng cấp hệ thống để có thể theo sát hơn với nhu cầu thực tế nhằm nâng cao năng suất sản xuất nông nghiệp và cũng như mang lại một giải pháp tuyệt vời cho những người có nhu cầu. Xin cảm ơn thầy Mai Đức Trung đã nhiệt tình giúp đỡ cũng như là hướng dẫn chúng em trong suốt quá trình thực hiện đồ án lần này.



The End.