



BÁO CÁO ĐỀ CƯƠNG THỰC HIỆN ĐỒ ÁN CUỐI KỲ

Vật lý cho Công nghệ thông tin-PHY00007_20CLC03



NHÓM 15

*Nguyễn Mạnh Cường 20127456
Nguyễn Trương Hoàng Thái 20127625
Trương Quốc Huân 20127172*

Mục lục

1. Thông tin nhóm.....	2
2. Vai trò và công việc của các thành viên	2
3. Tổng quát ý tưởng sản phẩm.....	4
4. Chức năng của sản phẩm	5
• Chế độ ánh sáng.....	6
• Chế độ thủ công	6
• Chế độ chuyển động	6
• Chức năng dành cho tất cả chế độ	6
5. Các thiết bị	7
6. Bảng vẽ phác thảo sản phẩm.....	8
• Bản vẽ 2D	8
• Bản vẽ 3D	8
7. Sơ đồ truyền và nhận dữ liệu của hệ thống.....	11
8. Giao diện trang web	12
9. Giải thích flow NodeRED.....	14
• Nhóm 1: Hiển thị dữ liệu từ Wokwi lên UI.....	15
• Nhóm 2: Hiển thị thời gian thực lên UI.....	16
• Nhóm 3: Điều khiển hệ thống bằng UI.....	17

1. Thông tin nhóm

- Mã số nhóm: 15
- Thành viên

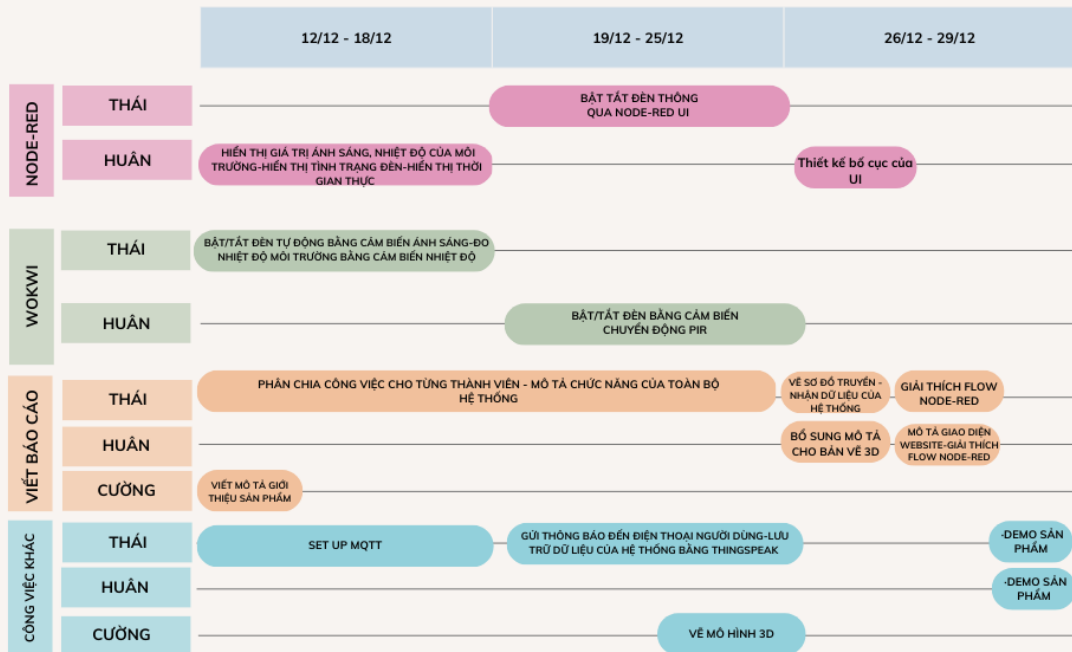
<i>Họ và tên</i>	<i>MSSV</i>
Nguyễn Trương Hoàng Thái	20127625
Trương Quốc Huân	20127172
Nguyễn Mạnh Cường	20127456

2. Vai trò và công việc của các thành viên

STT	Người phụ trách	Công việc	Phần trăm
1	Nguyễn Trương Hoàng Thái	<ul style="list-style-type: none">• Set up MQTT (chuẩn bị kết nối internet cho hệ thống)• Gửi thông báo đến điện thoại người dùng bằng IFTTT (điện thoại được set up trước)• Lưu trữ dữ liệu của hệ thống bằng ThingSpeak• Wokwi<ul style="list-style-type: none">○ Bật/tắt đèn tự động bằng cảm biến ánh sáng○ Đo nhiệt độ môi trường bằng cảm biến nhiệt độ• Node-RED<ul style="list-style-type: none">○ Bật tắt đèn thông qua Node-RED UI• Viết báo cáo<ul style="list-style-type: none">○ Mô tả chức năng của toàn bộ hệ thống○ Phân chia công việc cho từng thành viên○ Vẽ sơ đồ truyền nhận dữ liệu của hệ thống○ Giải thích flow Node-RED	49%
2	Trương Quốc Huân	<ul style="list-style-type: none">• Demo sản phẩm• Wokwi<ul style="list-style-type: none">○ Bật/tắt đèn bằng cảm biến chuyển động PIR• Node-RED<ul style="list-style-type: none">○ Thiết kế bố cục của UI○ Hiển thị giá trị ánh sáng, nhiệt độ của môi trường○ Hiển thị tình trạng đèn○ Hiển thị thời gian thực (đồng hồ số và đồng hồ kim)• Viết báo cáo<ul style="list-style-type: none">○ Bổ sung mô tả cho bản vẽ 3D○ Mô tả giao diện website○ Giải thích flow Node-RED	49%
3	Nguyễn Mạnh Cường	<ul style="list-style-type: none">• Vẽ mô hình 3D• Viết mô tả giới thiệu sản phẩm	2%

GANTT CHART

ĐỒ ÁN CUỐI KÌ VẬT LÝ CHO CÔNG NGHỆ THÔNG TIN-
PHY00007_20CLC03



Gantt chart mô tả quá trình hoàn thiện sản phẩm (vẽ bằng canva)

3. Tổng quát ý tưởng sản phẩm



a. Tên sản phẩm: hệ thống đèn tự động.

b. Giới thiệu: Ánh sáng của các loại đèn như đèn pin, đèn đường... là nhu cầu cơ bản trong đời sống hiện đại để đảm bảo an toàn và tránh những tai nạn đáng tiếc về đêm. Xã hội ngày càng phát triển, con người ngày càng bận rộn dẫn đến việc quản lý các hệ thống đèn một cách thủ công trở nên vô cùng khó khăn. Các vấn đề như ô nhiễm môi trường, cạn kiệt năng lượng từ đó ngày càng trở nên nghiêm trọng. Hệ thống đèn tự động được phát triển với công dụng bật/tắt các bóng đèn dựa trên điều kiện ánh sáng để phục vụ nhu cầu của con người và giải quyết các vấn đề về môi trường.

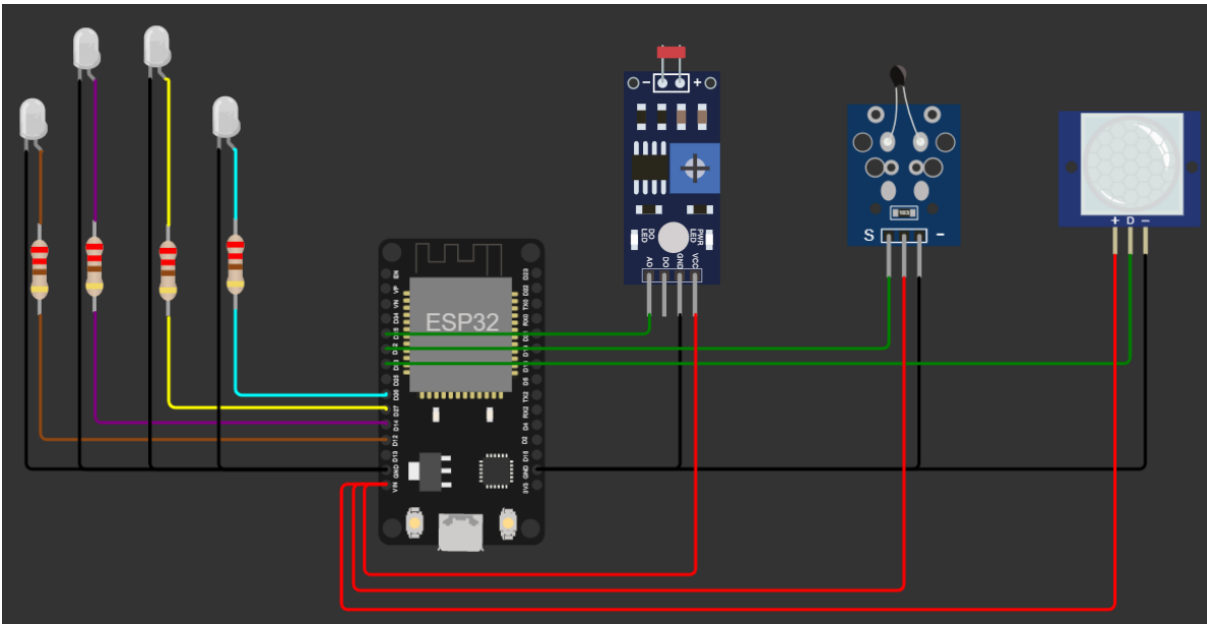


c. Ưu điểm của sản phẩm:

- i. Hệ thống đèn tự động giải quyết các nhu cầu về ánh sáng kịp thời hơn hệ thống đèn thủ công.
- ii. Tiết kiệm chi phí quản lý thủ công, chi phí bảo trì bảo dưỡng.
- iii. Giảm đáng kể lượng khí thải CO₂, năng lượng và vấn đề ô nhiễm ánh sáng.
- iv. Hệ thống được điều khiển thông qua kết nối không dây (internet), tạo nên sự tiện lợi trong việc quản lý.
- v. Người dùng luôn có thể cập nhật tình trạng của hệ thống thông qua gmail và cloud.

4. Chức năng của sản phẩm

- Sản phẩm được xây dựng trên module ESP32 kết hợp với Node-RED để xây dựng app quản lý và MQTT để truyền dữ liệu thông qua internet.
- Sản phẩm sử dụng ThingSpeak để lưu trữ dữ liệu và IFTTT để gửi thông báo đến người dùng trong những tình huống khẩn cấp.
- Hệ thống bao gồm các thiết bị:
 - Input: cảm biến ánh sáng (đo mật độ ánh sáng của môi trường), cảm biến nhiệt độ (đo nhiệt độ của môi trường), cảm biến chuyển động (nhận biết sự hiện diện của vật thể nóng chuyển động).
 - Output: 4 bóng led tượng trưng cho hệ thống đèn.
 - Module ESP32: bo mạch kết nối internet và là thành phần chính của hệ thống.



Mô hình mô phỏng hệ thống (từ trái sang phải: 4 bóng led, module ESP32, cảm biến ánh sáng, cảm biến nhiệt độ, cảm biến chuyển động)

- Hệ thống gồm 3 chế độ hoạt động là ánh sáng, thủ công và chuyển động.

- **Chế độ ánh sáng**

- Quang trở dựa trên mật độ ánh sáng đo được để bật/tắt hệ thống đèn tự động: khi giá trị ánh sáng (đơn vị: lux) mà quang trở đo được ≤ 100 (giá trị ánh sáng được tham khảo từ <https://docs.wokwi.com/parts/wokwi-photoresistor-sensor>) \rightarrow hệ thống đèn tự động bật và ngược lại.

- **Chế độ thủ công**

- Bật/tắt hệ thống đèn thông qua website: hệ thống đèn tiếp nhận các thao tác của người dùng trên website thông qua internet, từ đó bật/tắt tương ứng với tác vụ của người dùng.

- **Chế độ chuyển động**

- Bật/tắt đèn dựa trên cảm biến chuyển động Passive Infrared (PIR) motion sensor: PIR chuyển đổi tia nhiệt phát ra từ cơ thể người dùng thành dạng tín hiệu điện, từ đó phát hiện các vật thể nóng đang chuyển động để bật/tắt hệ thống đèn.


- **Chức năng dành cho tất cả chế độ**

- Nhiệt độ của môi trường luôn được cập nhật bởi cảm biến nhiệt độ được lắp đặt bên cạnh hệ thống đèn.
 - Hoạt động của đèn (thời gian bật/tắt); giá trị nhiệt độ của môi trường được cập nhật và lưu trữ trên cloud (cần kích hoạt chức năng “Thông báo từ

IFTTT” trên giao diện web): mỗi khi hệ thống đèn chuyển trạng thái → tình trạng đèn (bật/tắt) và nhiệt độ môi trường được lưu trữ lên ThingSpeak.

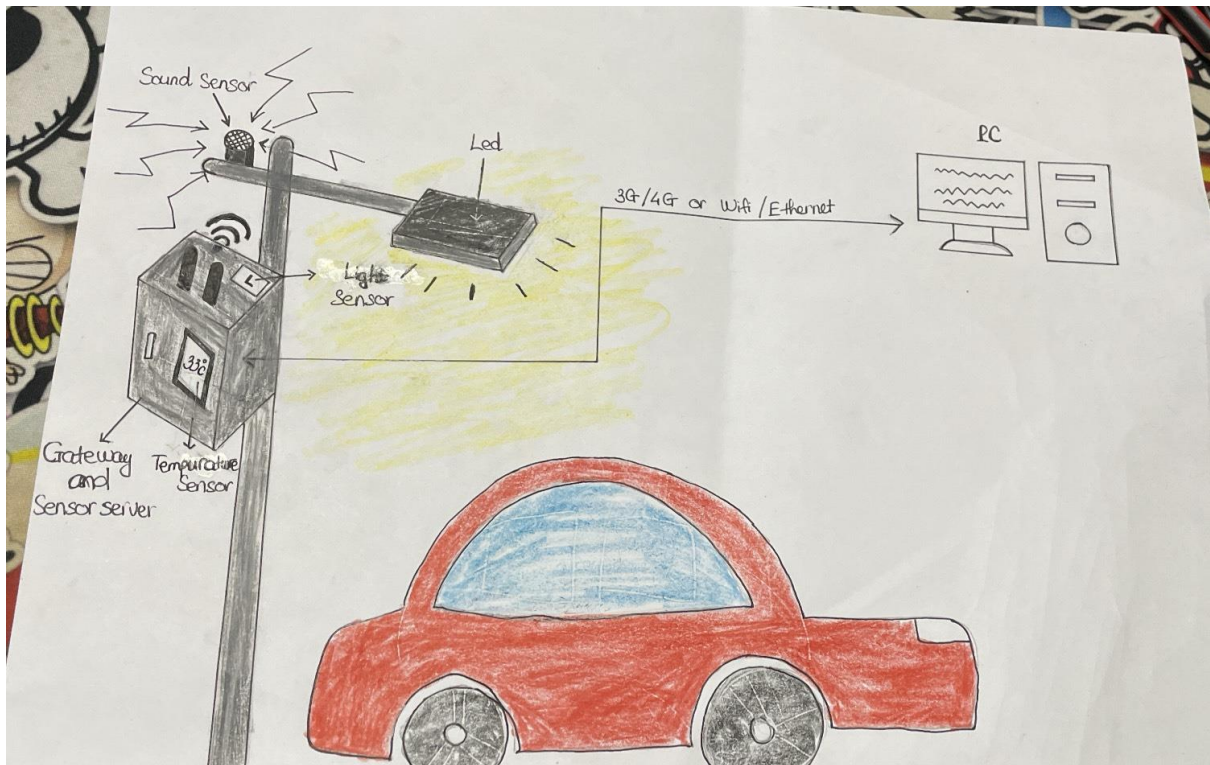
- Thông báo đến điện thoại của người dùng thông qua IFTTT (cần kích hoạt chức năng “Thông báo từ IFTTT” trên giao diện web): mỗi khi hệ thống chuyển trạng thái (bật → tắt/tắt → bật) hoặc khi nhiệt độ môi trường trên 50° C, thông báo sẽ được gửi đến điện thoại của người dùng. Với trường hợp nhiệt độ trên 50° C, cứ mỗi 5 giây người dùng sẽ nhận được cảnh báo (người dùng có thể tắt thông báo từ IFTTT thông qua giao diện web).

5. Các thiết bị

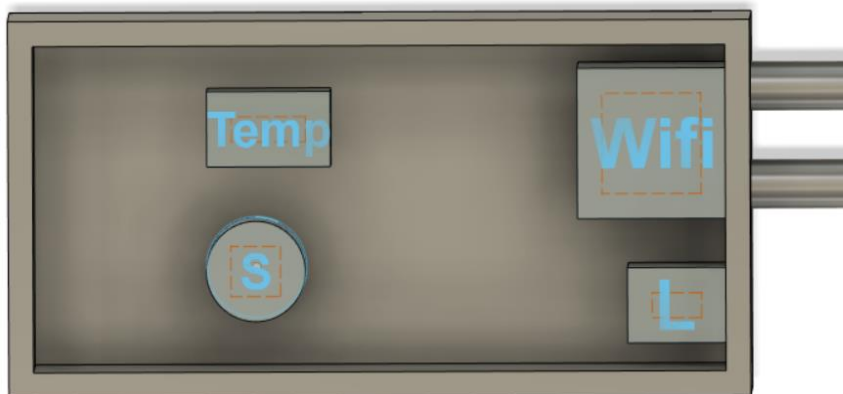
Tên thiết bị	Hình ảnh	Số lượng	Giá thành (1 thiết bị)
Quang trở		1	20.000đ
Cảm biến nhiệt độ		1	50.000đ
Cảm biến chuyển động		1	16.000đ
Relay		1	18.000đ
Module ESP32		1	109.000đ
Đèn led		4	3.000đ
Điện trở 220 Ohm 1/4W		6	15.000đ (gói 100 cái)

6. Bảng vẽ phác thảo sản phẩm

• Bản vẽ 2D

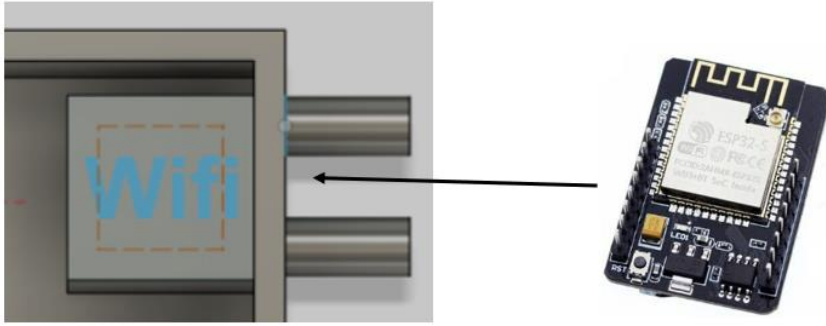


• Bản vẽ 3D



• Đây là hình ảnh 3D mặt cắt ngang của thiết bị để thấy được bên trong thiết bị gồm những linh kiện nhỏ nào.

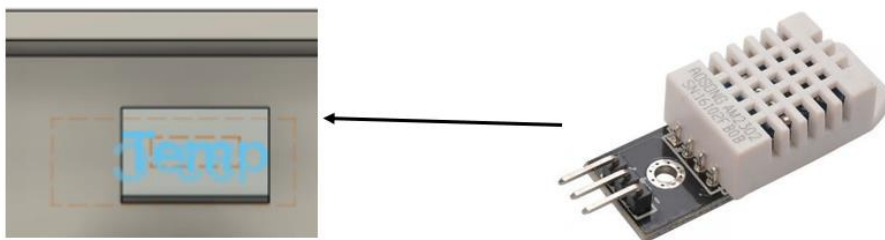
- Các linh kiện bên trong thiết bị bao gồm:
 - Mạch ESP32
 - Cảm biến nhiệt độ
 - Quang trở
 - Cảm biến chuyển động



- Mạch ESP32 nằm bên trong sản phẩm, đặt ở trên cùng dùng để kết nối các linh kiện lại với nhau và thu/phát wifi để nhận và truyền tải tín hiệu.



- Quang trở nằm bên trong sản phẩm, đặt ở trên cùng kế mạch ESP32 dùng để đo cường độ ánh sáng (phần lắp đặt quang trở trong sản phẩm thay vì bao bọc kín hết thì ở phần quang trở sẽ có rãnh khoét và dùng 1 miếng nhựa trong suốt để che kín lại → mục đích là để ánh sáng có thể đi vào để quang trở có thể đo được cường độ ánh sáng).



- Cảm biến nhiệt độ nằm bên trong sản phẩm, đặt ở giữa dùng để đo nhiệt độ môi trường xung quanh nơi sản phẩm được lắp đặt.

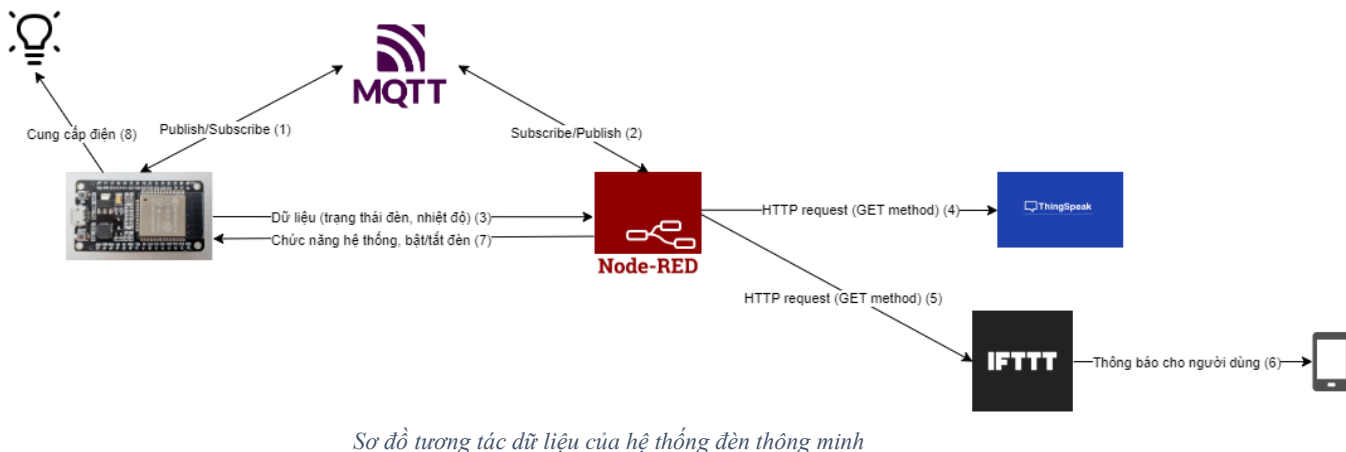


- Cảm biến chuyển động nằm bên trong sản phẩm, đặt ở giữa dùng để bắt chuyển động của vật thể nóng (với thân nhiệt bình thường là 37°C) bằng tia hồng ngoại trong phạm vi quét được.



- Đây là mặt trước của sản phẩm sau khi được đóng nắp để che chắn, bảo vệ các linh kiện bên trong khỏi các yếu tố ngoại lực và thời tiết, cùng với đó là 1 tay cầm để mở hộp sản phẩm và 1 màn hình LCD hiển thị nhiệt độ hiện tại của môi trường.

7. Sơ đồ truyền và nhận dữ liệu của hệ thống



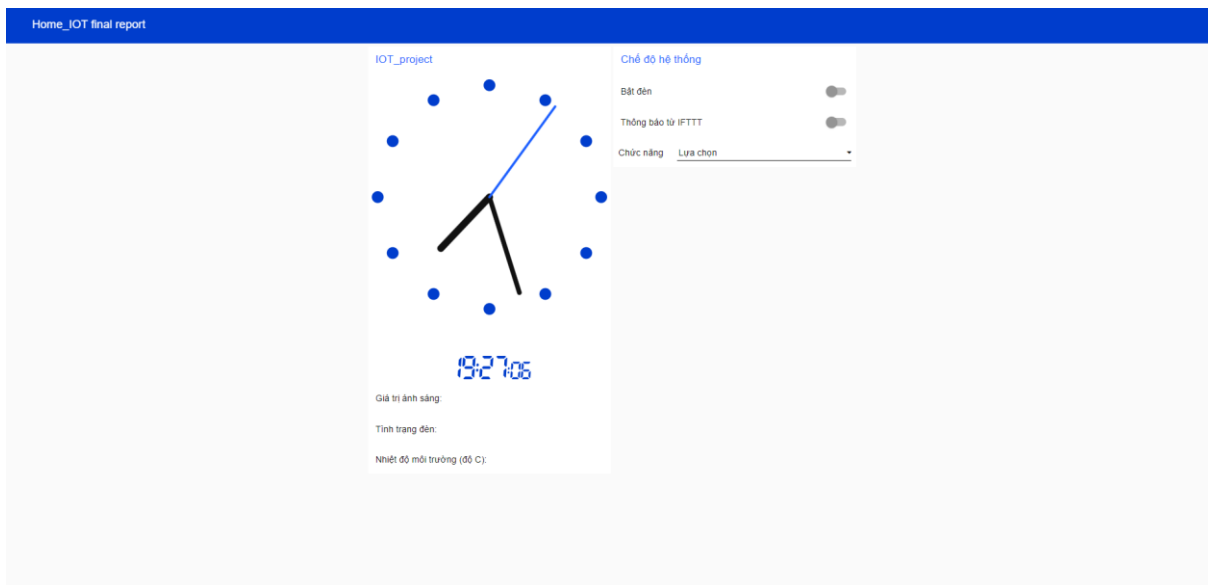
Mô tả sơ đồ:

- (1), (2): module ESP32 và Node-RED cùng publish/subscribe vào MQTT với cùng các topic để gửi nhận dữ liệu:
 - Khi module ESP32 publish (Node-RED subscribe topic tương ứng) dữ liệu về trạng thái đèn (bật/tắt), giá trị ánh sáng của môi trường, nhiệt độ của môi trường lên MQTT → Node-RED nhận được dữ liệu và xử lý dữ liệu (hiển thị lên giao diện người dùng hoặc thông báo đến điện thoại của người dùng và lưu trữ lên đám mây).
 - Khi Node-RED publish (module ESP32 subscribe topic tương ứng) dữ liệu về sự tương tác của người dùng (chức năng hệ thống, thông báo IFTTT...) lên MQTT → module ESP32 nhận được dữ liệu và xử lý dữ liệu (bật/tắt đèn, tìm kiếm chuyển động của vật thể...).
- (3): khi người dùng kích hoạt chức năng thông báo IFTTT, dữ liệu về tình trạng đèn và nhiệt độ môi trường được module ESP32 publish đến các topic để Node-RED gửi đến ThingSpeak và thông báo đến người dùng.
- (4): Node-RED gửi dữ liệu đã nhận được ở liên kết (3) đến ThingSpeak thông qua phương thức HTTP request GET. Sau đó, ThingSpeak sẽ lưu trữ những dữ liệu này lên cloud.
- (5): Node-RED gửi dữ liệu đã nhận được ở liên kết (3) đến IFTTT thông qua phương thức HTTP request GET.
- (6): IFTTT gửi thông báo đến điện thoại đã được setup từ trước của người dùng.

- (7): khi người dùng tương tác với hệ thống thông qua Node-RED UI, dữ liệu tương tác của người dùng được Node-RED publish đến các topic. ESP32 nhận được các dữ liệu và xử lý dữ liệu.

- (8): dựa trên dữ liệu nhận được ở liên kết (7), module ESP32 cung cấp điện để bật/tắt hệ thống đèn.

8. Giao diện trang web



- Đây là toàn bộ giao diện của trang web dùng để thao tác tới thiết bị.

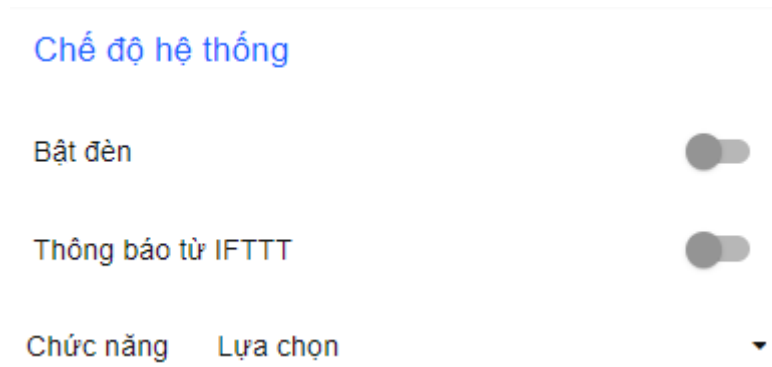


- Chính giữa trang web sẽ có 1 đồng hồ lớn chạy bằng kim và dưới đó là 1 đồng hồ số để hiển thị giá trị thời gian hiện tại.

- Tiếp đến là 1 đoạn text để hiển thị cho người dùng thấy được giá trị ánh sáng tại ngay thời điểm được đo (giá trị ánh sáng được hiển thị trên web sẽ được tính bằng đơn vị Lux).

- Ngay dưới đoạn text giá trị ánh sáng sẽ là Tình trạng đèn, dùng để hiển thị cho người dùng thấy được tình trạng đèn tại thời điểm đó đang được bật hay tắt (giá trị hiển thị là On hoặc Off).

- Và cuối cùng đoạn text nhiệt độ môi trường, dùng để hiển thị cho người dùng thấy được nhiệt độ môi trường xung quanh sản phẩm tại thời điểm đó là bao nhiêu độ C (với trường hợp nhiệt độ trên 50° C, cứ mỗi 5 giây người dùng sẽ nhận được cảnh báo).



- Trên hình là những chế độ cho người dùng lựa chọn để sử dụng sản phẩm theo nhu cầu của họ.



- Đầu tiên là chế độ bật tắt đèn, người dùng thao tác bật tắt đèn bất cứ khi nào thông qua web (lưu ý: chế độ này chỉ sử dụng được khi bạn lựa chọn chức năng thủ công).



- Tiếp theo sẽ là chế độ gửi thông báo đến điện thoại người dùng, hệ thống sẽ báo cho người dùng mỗi khi hệ thống chuyển trạng thái (bật → tắt/tắt → bật) hoặc khi nhiệt độ môi trường trên 50° C, thông báo sẽ được gửi đến điện thoại của người dùng. Với trường hợp nhiệt độ trên 50° C, cứ mỗi 5 giây người dùng sẽ

nhận được cảnh báo (người dùng có thể tắt thông báo từ IFTTT thông qua giao diện web).

Chức năng	Thủ công	▼
Chức năng	Ánh sáng	▼
Chức năng	Chuyển động	▼

- Hệ thống sẽ có 3 loại chức năng cho người dùng lựa chọn:
 - Chức năng thủ công: người dùng sẽ phải thao tác bật/tắt đèn trên giao diện web (lưu ý: giá trị ánh sáng hiển thị trên web sẽ là giá trị lần cuối cùng mà cảm biến đo được khi người dùng bật chế độ Ánh sáng).
 - Chức năng Ánh sáng: hệ thống đèn sẽ tự động bật/tắt đèn dựa trên giá trị ánh sáng mà thiết bị đo được tại thời điểm hiện tại.
 - Chức năng Chuyển động: khi sử dụng chức năng này nếu có 1 vật thể sống với thân nhiệt trung bình 37°C đi ngang qua tầm quét của thiết bị thì đèn sẽ tự động sáng trong 5s và sẽ tắt cảm biến sẽ ngắt đèn trong 1.2s sau đó cảm biến sẽ tiếp tục quét để bắt chuyển động.

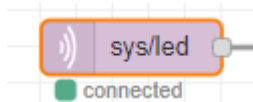
9. Giải thích flow NodeRED

- Flow được chia thành 4 nhóm chính dựa trên các chức năng của sản phẩm:
 - Nhóm 1: nhận dữ liệu từ Wokwi (hệ thống đèn) để hiển thị lên Node-RED UI.
 - Nhóm 2: hiển thị thời gian thực (UTC+07:00) lên Node-RED UI.
 - Nhóm 3: điều khiển hệ thống đèn bằng Node-RED UI.
 - Nhóm 4: thông báo IFTTT và lưu trữ cloud bằng ThingSpeak.

• Nhóm 1: Hiển thị dữ liệu từ Wokwi lên UI



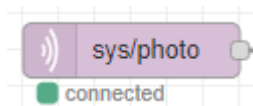
Nhóm 1



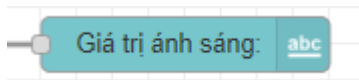
- Tên: sys/led
- Cấu hình:
 - Server: broker.hivemq.com (Để kết nối internet thông qua MQTT)
 - Topic: sys/led (Tín hiệu đèn On/Off)
 - QoS: 2 (Để chắc chắn sẽ có 1 gói tin được gửi và nhận)
- Công dụng: nhận tín hiệu đèn từ thiết bị.



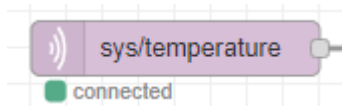
- Tên: Tình trạng đèn
- Công dụng: In ra giá trị tình trạng đèn (On/Off)



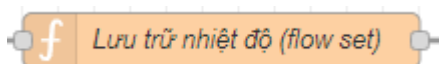
- Tên: sys/photo
- Cấu hình:
 - Server: broker.hivemq.com (Để kết nối internet thông qua MQTT)
 - Topic: sys/photo (Giá trị quang trở)
 - QoS: 2 (Để chắc chắn sẽ có 1 gói tin được gửi và nhận)
- Công dụng: Nhận giá trị ánh sáng mà quang trở đo được



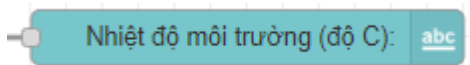
- Tên: Giá trị ánh sáng
- Công dụng: In ra giá trị ánh sáng (Đơn vị Lux)



- Tên: sys/temperature
- Cấu hình:
 - Server: broker.hivemq.com (Để kết nối internet thông qua MQTT)
 - Topic: sys/photo (Giá trị nhiệt độ)
 - QoS: 2 (Để chắc chắn sẽ có 1 gói tin được gửi và nhận)
- Công dụng: Nhận giá trị nhiệt độ mà cảm biến nhiệt đo được

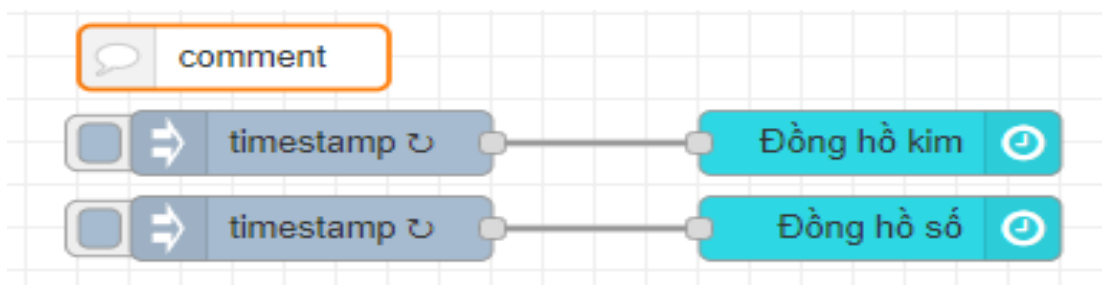


- Tên: Lưu trữ nhiệt độ (flow set)
- Công dụng: Lưu trữ giá trị nhiệt độ



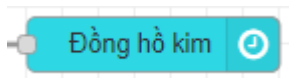
- Tên: Nhiệt độ môi trường (độ C)
- Công dụng: In ra giá trị nhiệt độ (Đơn vị 0C)

• Nhóm 2: Hiển thị thời gian thực lên UI

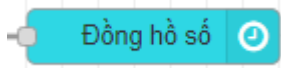




- Tên: timestamp
- Cấu hình:
 - Repeat: interval every 0.01s (set khoảng thời gian lặp lại 0.01 để đồng hồ được chạy liên tục)

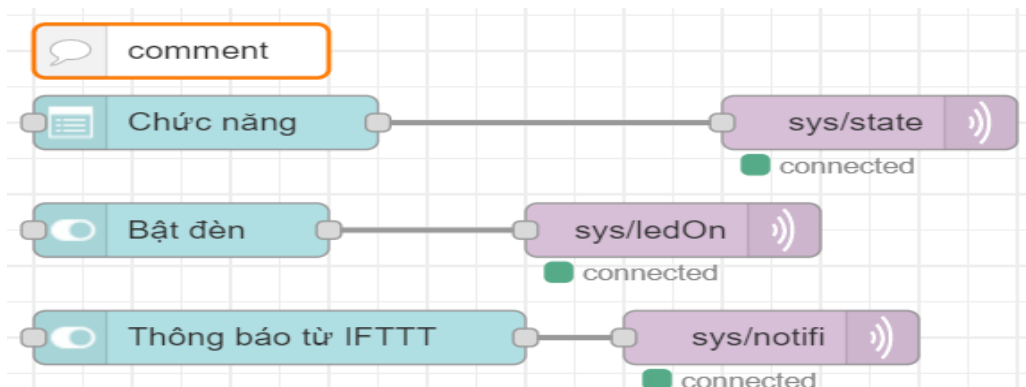


- Tên: Đồng hồ kim
- Công dụng: Hiển thị đồng hồ kim

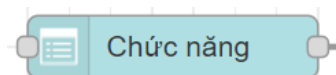


- Tên: Đồng hồ số
- Công dụng: Hiển thị đồng hồ số

• Nhóm 3: Điều khiển hệ thống bằng UI



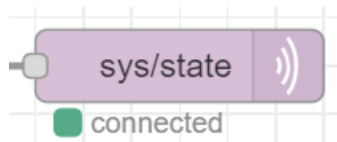
Nhóm 3



- Tên: Chức năng
- Cấu hình:
 - Tab: [Home_IOT final report]
 - Group: Chế độ hệ thống
 - Placeholder: “Lựa chọn”

○ Options: photoresistor (string) – Ánh sáng, PIR (string) – Chuyển động, manual (string) – Thủ công.

• Công dụng: ghi nhận và xử lý lựa chọn chức năng hệ thống của người dùng trên Node-RED UI.



- Tên: sys/state

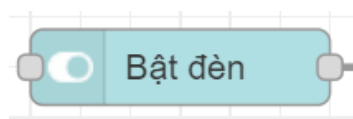
- Cấu hình:

- Server: broker.hivemq.com:1883

- Topic: sys/state

- QoS: 2

• Công dụng: gửi dữ liệu về lựa chọn chức năng hệ thống của người dùng đến module ESP32.



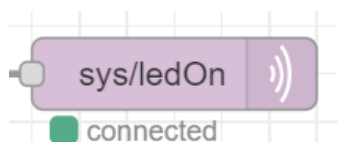
- Tên: Bật đèn

- Cấu hình:

- Tab: [Home_IOT final report]

- Group: Chế độ hệ thống

• Công dụng: ghi nhận và xử lý hành động bật/tắt đèn bằng Node-RED UI của người dùng khi hệ thống ở chế độ thủ công.



- Tên: sys/ledOn

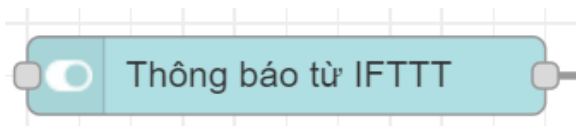
- Cấu hình:

- Server: broker.hivemq.com:1883

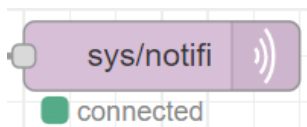
- Topic: sys/ledOn

- QoS: 2

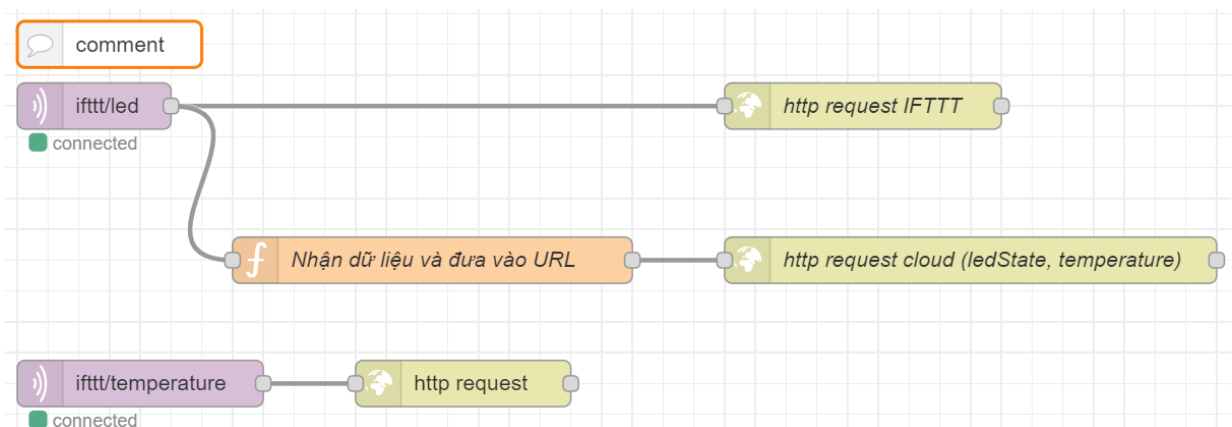
• Công dụng: gửi dữ liệu về hành động bật/tắt của người dùng bằng Node-RED UI đến module ESP32.



- Tên: Thông báo từ IFTTT
- Cấu hình:
 - Tab: [Home_IOT final report]
 - Group: Chế độ hệ thống
- Công dụng: ghi nhận và xử lý nhu cầu sử dụng chức năng thông báo đến điện thoại của người dùng.



- Tên: sys/notifi
- Cấu hình:
 - Server: broker.hivemq.com:1883
 - Topic: sys/state
 - QoS: 2
- Công dụng: gửi dữ liệu về nhu cầu sử dụng chức năng thông báo đến điện thoại của người dùng đến module ESP32.
- Nhóm 4: Thông báo đến người dùng bằng IFTTT và lưu trữ dữ liệu lên cloud.



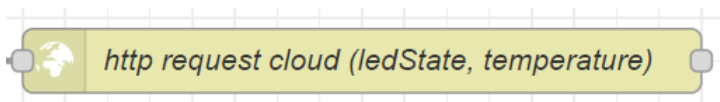
Nhóm 4



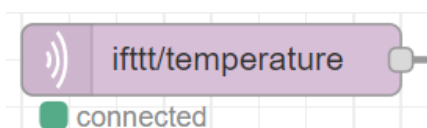
- Tên: ifttt/led
- Cấu hình:
 - Server: broker.hivemq.com:1883
 - Topic: ifttt/led
 - QoS: 2
- Công dụng: nhận dữ liệu về trạng thái của đèn (bật/tắt) từ hệ thống (Wokwi) khi người dùng kích hoạt chế độ thông báo IFTTT.



- Tên: http request IFTTT
- Cấu hình:
 - Method: GET
- Công dụng: gửi một cảnh báo đến điện thoại người dùng (được setup trước) khi hệ thống đèn chuyển trạng thái bằng phương thức http request GET.

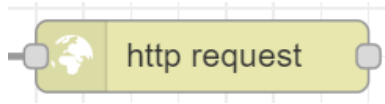


- Tên: http request cloud (ledState, temperature)
- Cấu hình:
 - Method: GET
- Công dụng: gửi dữ liệu về tình trạng đèn và nhiệt độ môi trường đến ThingSpeak để lưu trữ dữ liệu bằng phương thức http request GET.



- Tên: ifttt/temperature
- Cấu hình:
 - Server: broker.hivemq.com:1883

- Topic: ifttt/temperature
- QoS: 2
- Công dụng: nhận dữ liệu về nhiệt độ môi trường từ hệ thống (Wokwi) khi người dùng kích hoạt chế độ thông báo IFTTT.



- Tên: http request
- Cấu hình:
 - Method: GET
- Công dụng: gửi một cảnh báo đến điện thoại người dùng khi nhiệt độ môi trường vượt quá 50°C.