

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

-----o0o-----



**BÁO CÁO: Sản phẩm điều khiển điều
hòa thông minh**

Môn: Vật lý cho công nghệ thông tin

Giảng viên hướng dẫn:
Thầy Cao Xuân Nam
Thầy Đặng Hoài Thương

NHÓM 15:

Nguyễn Văn Tuấn Kiệt

21127331

Cao Nguyễn Khánh

21127627

Huỳnh Sơn Hà

21127035

MỤC LỤC

<i>I. THÔNG TIN NHÓM:</i>	<i>2</i>
<i>1.GIỚI THIỆU CÁC THÀNH VIÊN:</i>	<i>2</i>
<i>2.VAI TRÒ VÀ CÔNG VIỆC:</i>	<i>2</i>
<i>II.THÔNG TIN SẢN PHẨM:</i>	<i>4</i>
<i>1.TÊN SẢN PHẨM:</i>	<i>4</i>
<i>Điều khiển điều hòa thông minh</i>	<i>4</i>
<i>2.MÔ TẢ VỀ SẢN PHẨM VÀ CHỨC NĂNG SẢN PHẨM:</i>	<i>4</i>
<i>III. SƠ ĐỒ TRUYỀN DỮ LIỆU:</i>	<i>5</i>
<i>IV. THIẾT KẾ 3D:</i>	<i>6</i>
<i>V.GIAO DIỆN WEB:</i>	<i>9</i>
<i>VI.MÔ TẢ WEB:</i>	<i>10</i>
<i>VII.FLOW NODE-RED</i>	<i>11</i>

I. THÔNG TIN NHÓM:

1.GIỚI THIỆU CÁC THÀNH VIÊN:

Mã số nhóm: Nhóm 15

STT	Học tên	MSSV
1	Nguyễn văn Tuấn Kiệt	21127035
2	Cao Nguyễn Khánh	21127627
3	Huỳnh Sơn Hà	21127035

2.VAI TRÒ VÀ CÔNG VIỆC:

Thành viên	Công việc
Nguyễn Văn Tuấn Kiệt	Quay video
	Code điều khiển máy lạnh (dùng nút bấm trên sản phẩm)
	Code dùng web điều khiển máy lạnh (node-red)
	Xử lý RTC
Cao Nguyễn Khánh	Vẽ sơ đồ truyền và nhận dữ liệu giữa các đối tượng trong hệ thống IoT
	Lưu trữ nhiệt độ, độ ẩm đã đo được lên cloud và lấy lịch sử (thingspeak, node-red)
	Thiết kế web (node-red)
	Gửi thông báo cho người dùng khi nhiệt độ cao (IFTTT, node-red)

Huỳnh Sơn Hà	Thiết kế bản vẽ 3D
	Đo nhiệt độ, độ ẩm
	Hiển thị nhiệt độ, độ ẩm lên LCD
	Hiển thị nhiệt độ, độ ẩm lên web (node-red)
	Code tự động điều khiển máy lạnh

II.THÔNG TIN SẢN PHẨM:

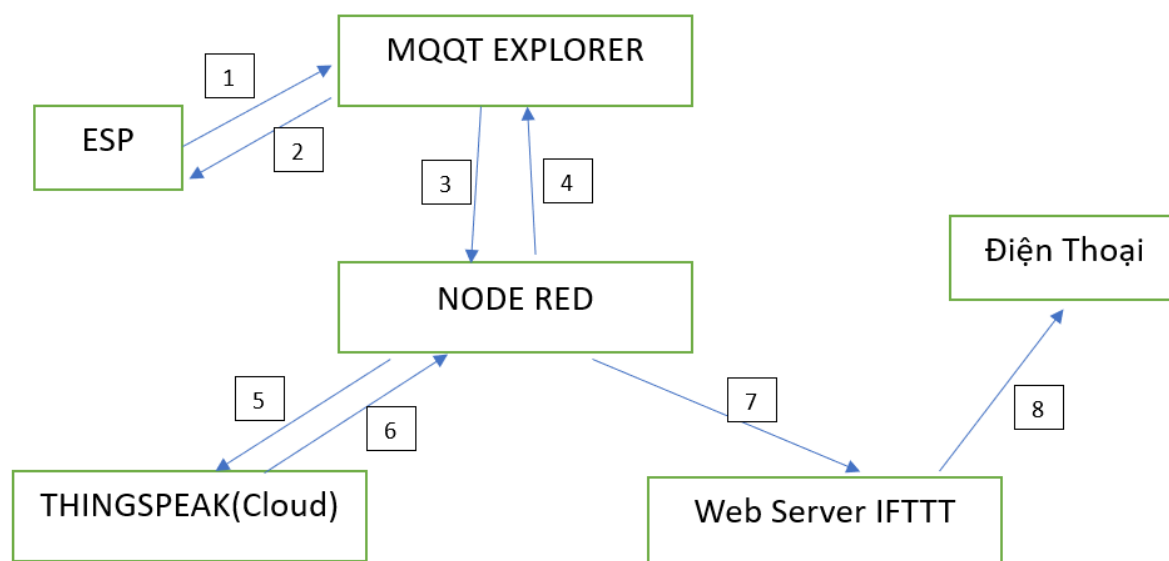
1.TÊN SẢN PHẨM:

Điều khiển điều hòa thông minh

2.MÔ TẢ VỀ SẢN PHẨM VÀ CHỨC NĂNG SẢN PHẨM:

STT	Chức năng
1	Đo nhiệt độ và độ ẩm không khí (thông qua cảm biến nhiệt độ và độ ẩm)
2	Thông báo khẩn cấp: Dựa vào nhiệt độ đo được nếu nhiệt độ quá lớn có thể thông báo cho người sở hữu
3	Hiển thị các thông tin về nhiệt độ và độ ẩm lên web
4	Lưu trữ lên cloud: <ul style="list-style-type: none">● Ghi nhận nhiệt độ không khí, độ ẩm● Lịch sử ghi nhận lại nhiệt độ, độ ẩm
5	Điều khiển điều hòa (thông qua bóng hồng ngoại và thiết bị thu hồng ngoại)
6	Hiển thị các thông số đo được lên màn hình LCD
7	Sử dụng ESP8266: Kết nối internet qua thiết bị ESP8266

III. SƠ ĐỒ TRUYỀN DỮ LIỆU:



Hình: Sơ đồ truyền dữ liệu

Thông tin Gửi và Nhận:

[1] : ESP gửi tới MQTT Explorer thông tin về : nhiệt độ, độ ẩm, trạng thái của điều hòa(Bật/ Tắt)

[2] : ESP nhận từ MQTT Explorer thông tin về : thực hiện lệnh tăng, giảm nhiệt độ và bật, tắt điều hòa.

[3] : MQTT Explorer gửi tới Node Red thông tin về : nhiệt độ, độ ẩm, trạng thái của điều hòa(Bật/ Tắt) để thể hiện trên web.

[4] : MQTT Explorer nhận từ Node Red thông tin về : thực hiện lệnh tăng, giảm nhiệt độ và bật, tắt điều hòa thông qua các nút bấm trên web.

[5] : Node Red gửi tới ThingSpeak (Cloud) thông tin về : nhiệt độ, độ ẩm để lưu trữ dữ liệu.

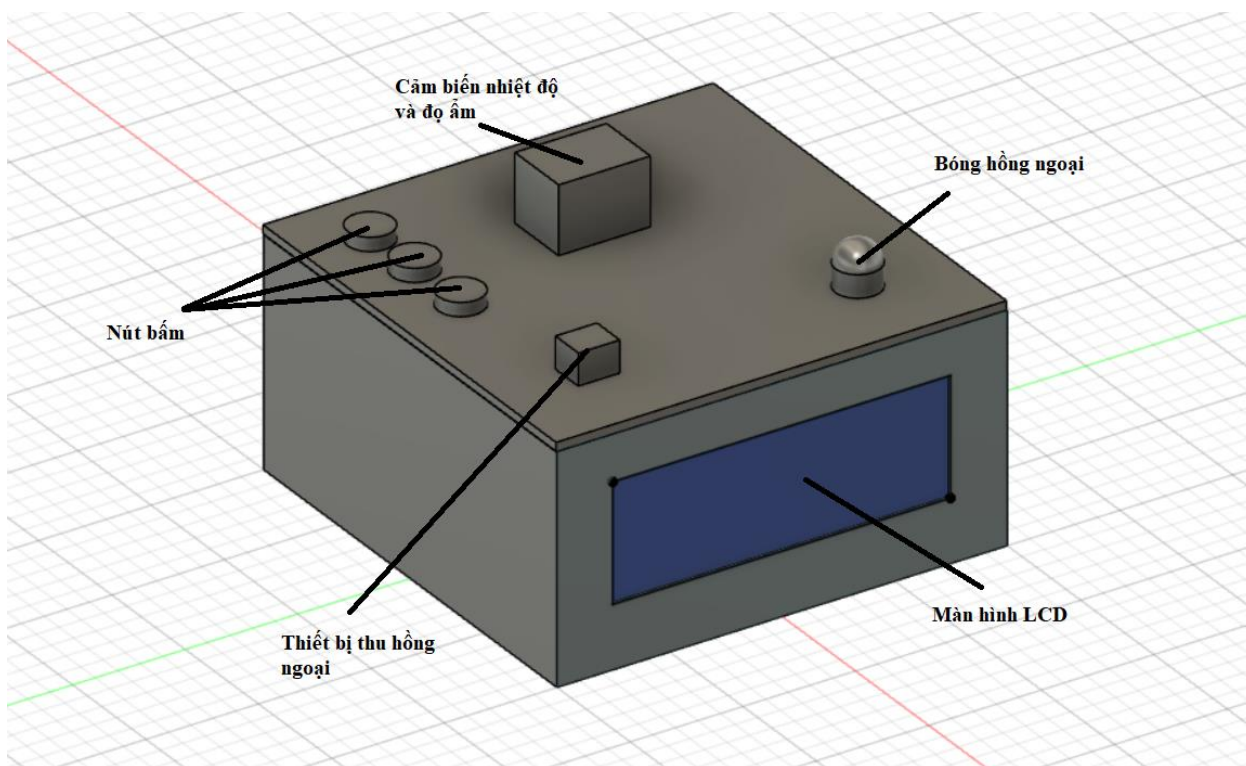
[6] : Node Red nhận từ ThingSpeak (Cloud) thông tin về : nhiệt độ, độ ẩm đã lưu trữ dữ liệu. Lấy 200 dữ liệu mới nhất được lưu trữ để thể hiện trên web.

[7] : Node Red gửi tới Web Server IFTTT thông tin cảnh báo về nhiệt độ. Nếu nhiệt độ lên tới trên 40 độ C.

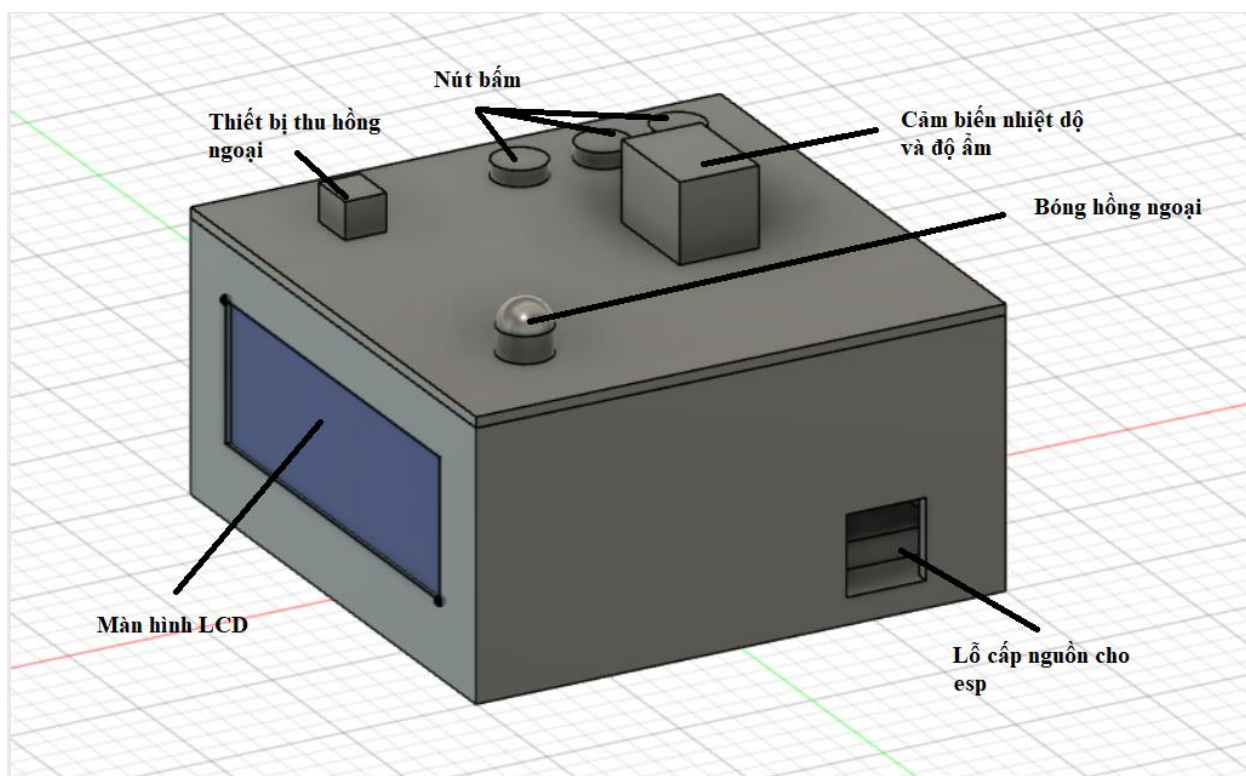
[8] : Web Server IFTTT gửi tới Điện thoại thông tin cảnh báo về nhiệt độ. Nếu nhiệt độ lên tới trên 40 độ C.

IV. THIẾT KẾ 3D:

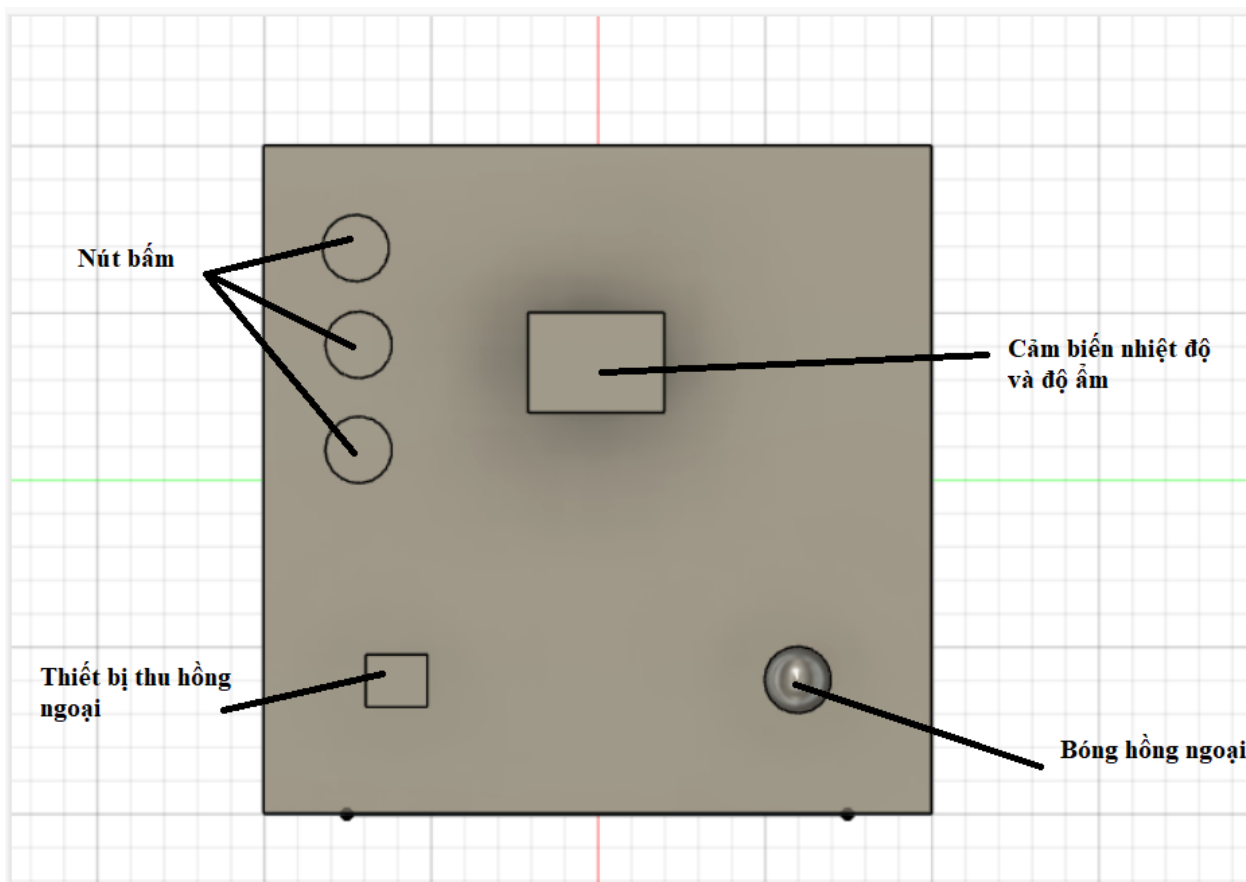
- Hình ảnh các thiết bị có trong hình:



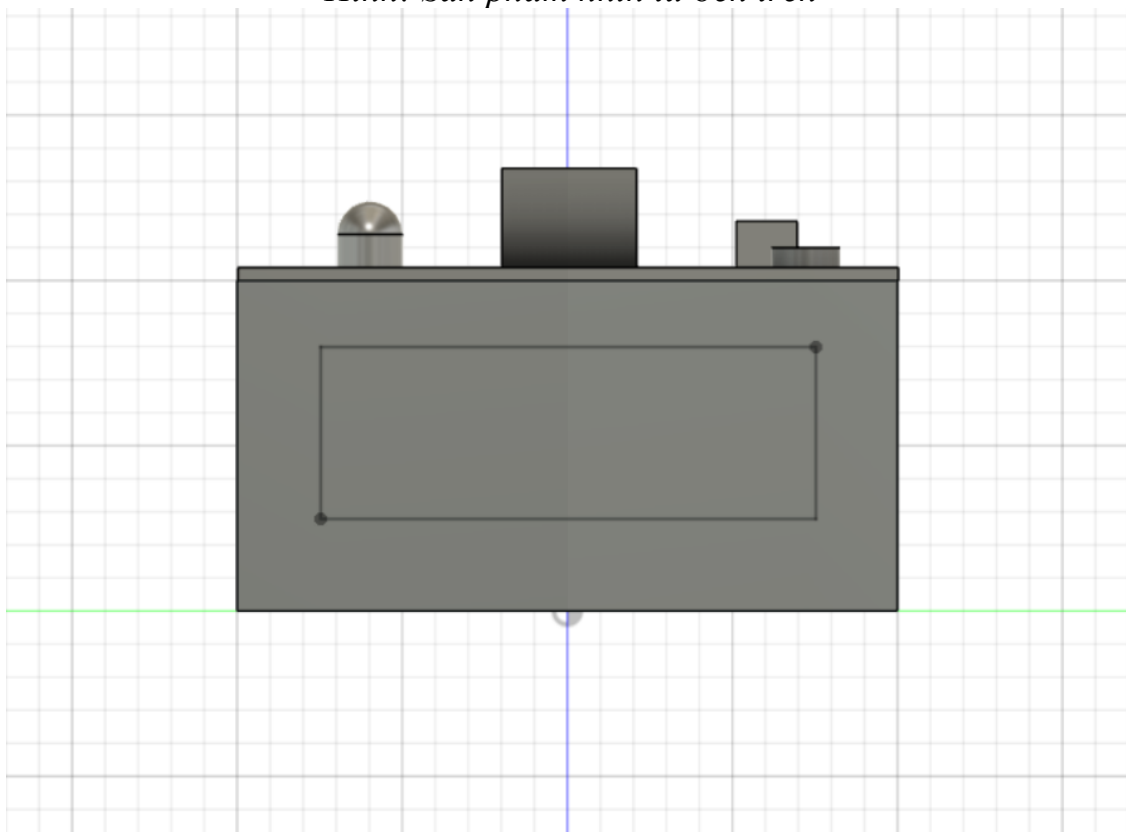
Hình: Sản phẩm nhìn từ bên trái, phía trước



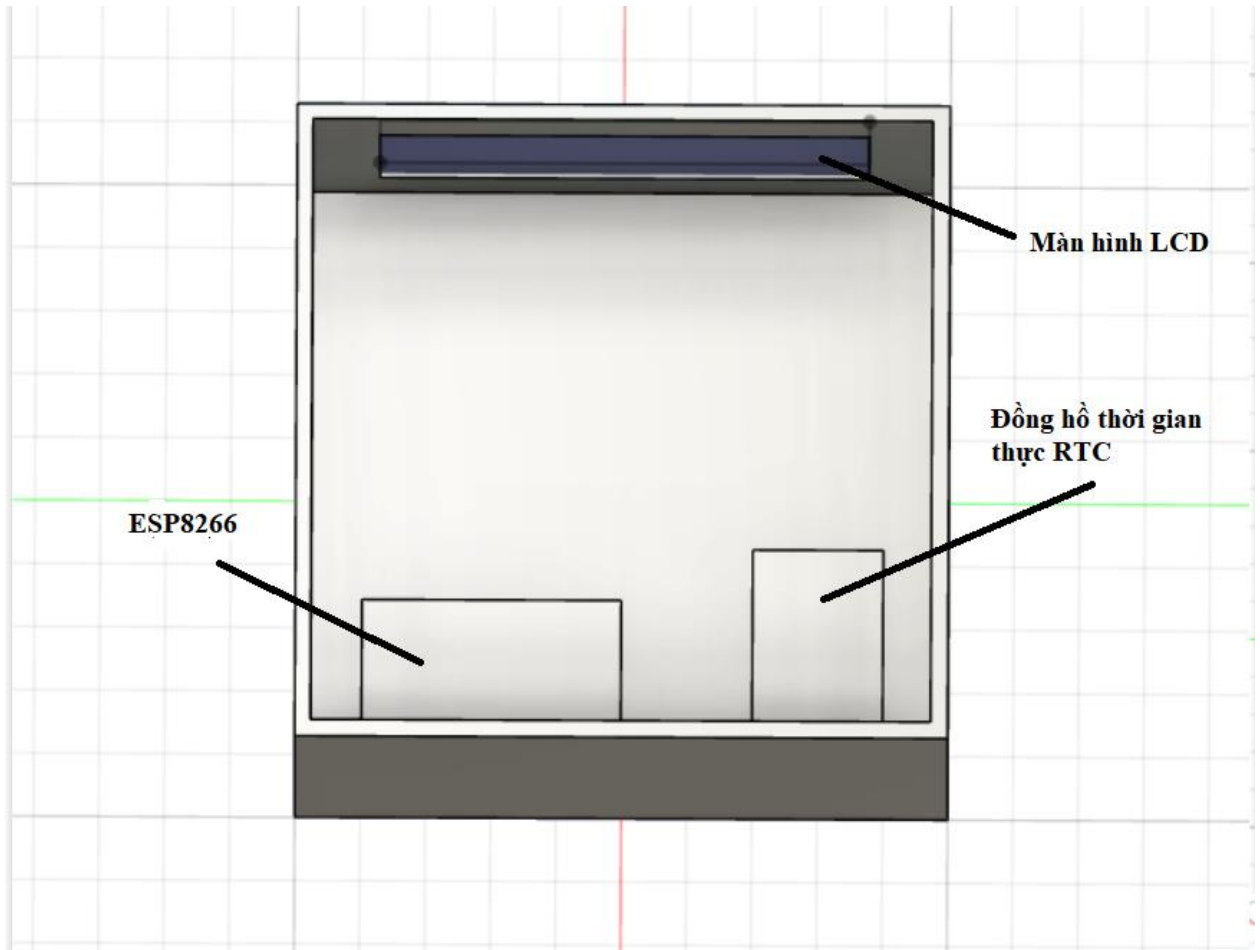
Hình: Sản phẩm nhìn từ bên phải, phía trước



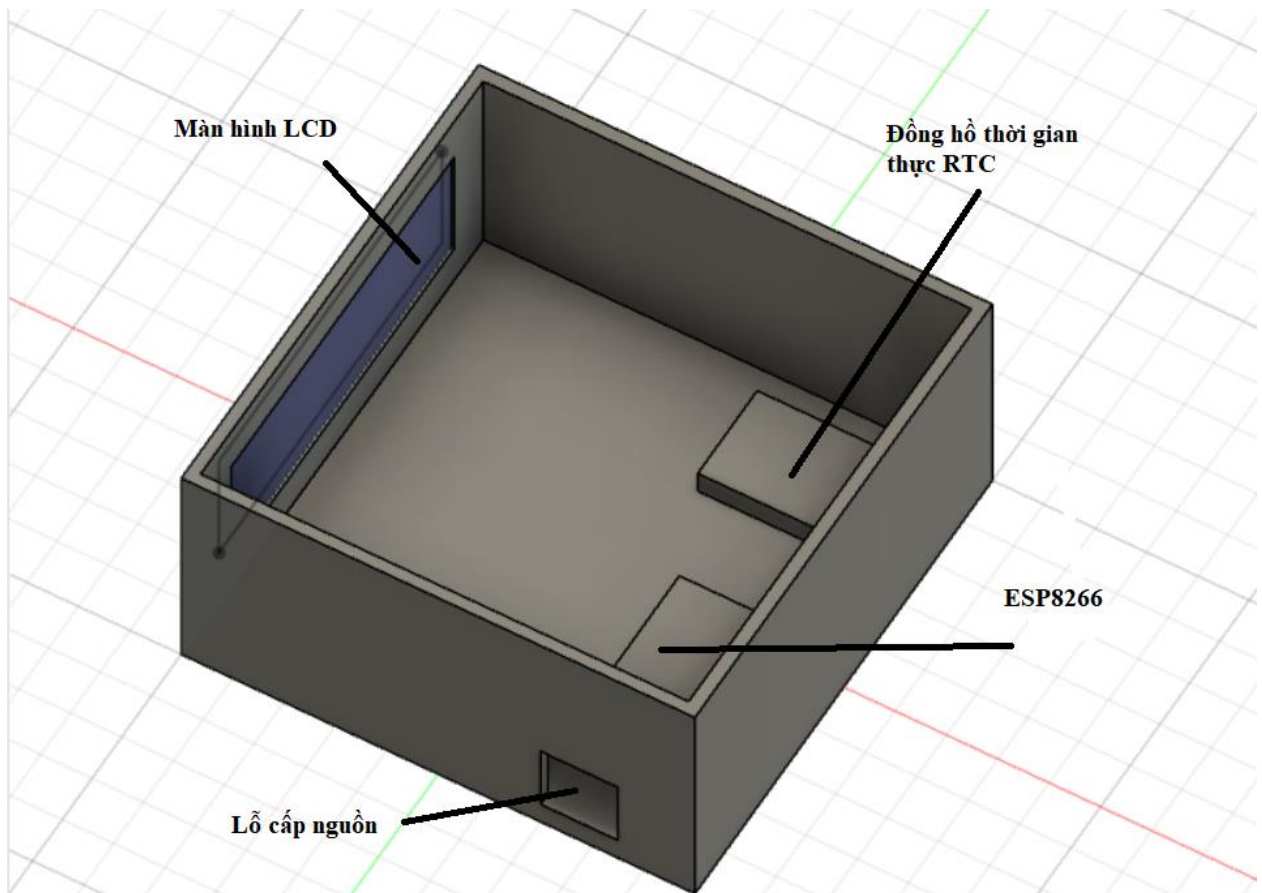
Hình: Sản phẩm nhìn từ bên trên



Hình: Sản phẩm nhìn từ đằng sau

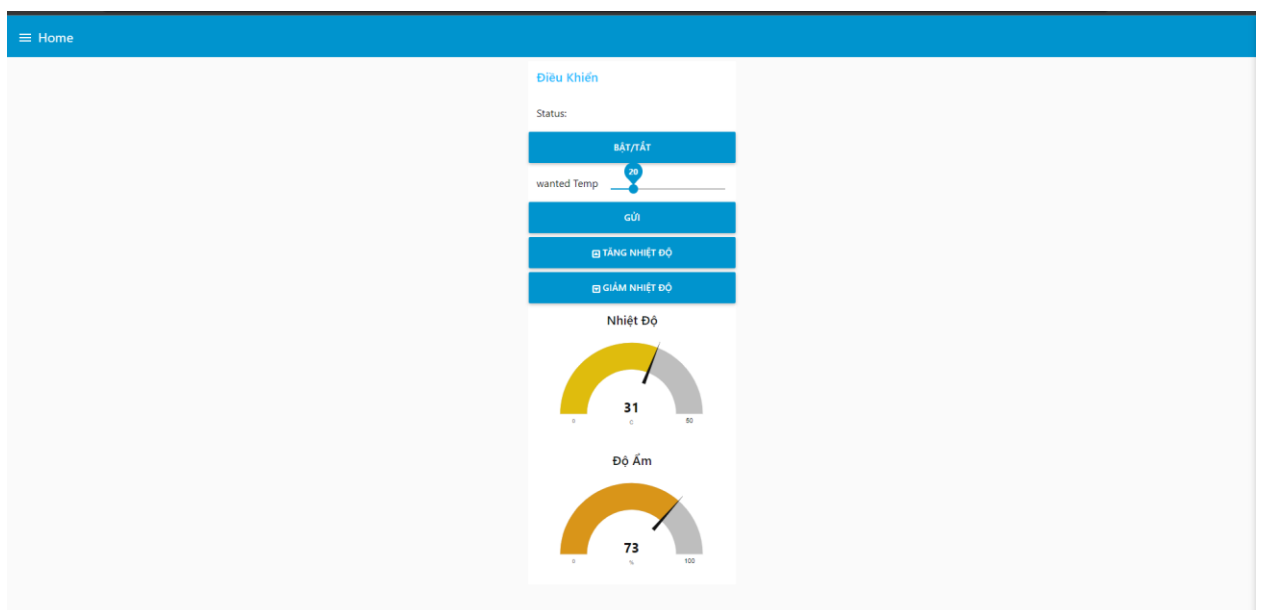


Hình: Bên trong sản phẩm nhìn từ bên trên



Hình: Bên trong sản phẩm nhìn từ phía trên bên trái

V.GIAO DIỆN WEB:

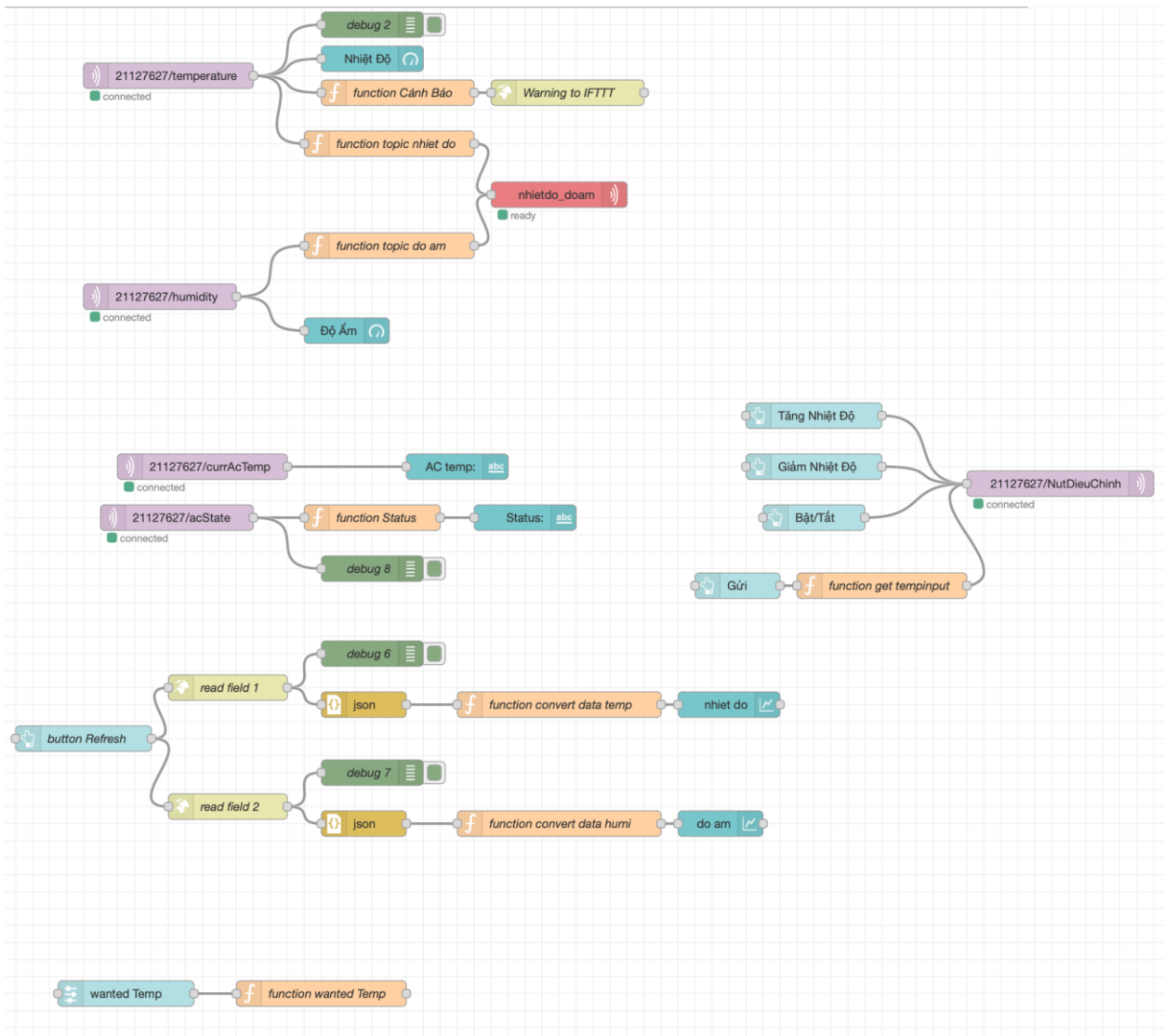




VI.MÔ TẢ WEB:

- Hiển thị thông tin các thành viên
- Có các nút bật/tắt, tăng nhiệt độ, giảm nhiệt độ: Dùng để điều khiển máy lạnh bằng web
- Hiển thị trạng thái hiện tại của điều hòa: On hoặc Off
- Hiển thị nhiệt độ và độ ẩm đo được dưới dạng gauge
- Hiển thị được lịch sử nhiệt độ và độ ẩm đã đo được từ thinkspeak dưới dạng chart
- Thông báo cho người dùng nếu nhiệt độ quá cao

VII.FLOW NODE-RED



Danh sách các node và chức năng:

- [21127627/temperature] (mqtt in): nhận thông tin về nhiệt độ từ mqtt explorer.
- [21127627/humidity] (mqtt in): nhận thông tin về độ ẩm từ mqtt explorer.
- [21127627/acState] (mqtt in): nhận thông tin về trạng thái từ mqtt explorer.
- [21127627/currAcTemp] (mqtt in): nhận thông tin về nhiệt độ đang đặt của điều hòa.
- [21127627/NutDieuChinh] (mqtt out): gửi thông tin thực hiện lệnh bật/ tắt , tăng, giảm nhiệt độ điều hòa tới mqtt explorer.
- [21127627/wantedTemp] (mqtt out): gửi thông tin về nhiệt độ mong muốn tới mqtt explorer.
- [Nhiệt Độ] (gauge): thể hiện nhiệt độ nhận được từ mqtt explorer lên gauge.
- [Độ Ẩm] (gauge): thể hiện nhiệt độ nhận được từ mqtt explorer lên gauge.

- [nhiet do] (chart): thể hiện nhiệt độ nhận được từ thingspeak (cloud) lên biểu đồ.
- [do am] (chart): thể hiện độ ẩm nhận được từ thingspeak (cloud) lên biểu đồ.
- [function Cảnh Báo](function): nếu nhiệt độ nhận được > 40 thì cảnh báo tới ifttt.

Name: function Cảnh Báo

Setup On Start On Message

```

1
2  var nhietdo = msg.payload
3
4  if (nhietdo < 40)
5  {
6      return null;
7  }
8
9  msg.payload = nhietdo;
10
11  return msg;
12

```

- [function topic nhiet do](function): nhận được nhiệt độ từ mqtt explorer thì set topic thành “nhiet do” để đưa lên thingspeak.

Name: function topic nhiet do

Setup On Start On Message

```

1  msg.topic = "nhiet do";
2  return msg;

```

- [function topic do am](function): nhận được độ ẩm từ mqtt explorer thì set topic thành “do am” để đưa lên thingspeak.

Name: function topic do am

Setup On Start On Message

```

1  msg.topic = "do am";
2  return msg;

```

- [function Status](function): nhận được thông tin trạng thái từ mqtt explorer. Nếu là 1 thì trả ra On, ngược lại thì Off. node này nối với 1 node text để hiển thị lên màn hình.

Name: function Status

Setup On Start **On Message**

```

1
2  if(msg.payload == "1")
3  {
4      msg.payload=  "On";
5  }
6  else
7  {
8      msg.payload = "off"
9  }
10
11  return msg

```

- [function wanted Temp](function) : hàm này lấy thông tin từ node Slide là 1 số nguyên từ 18 đến 28, và lưu vào biến tempinput của flow.

Name: function wanted Temp

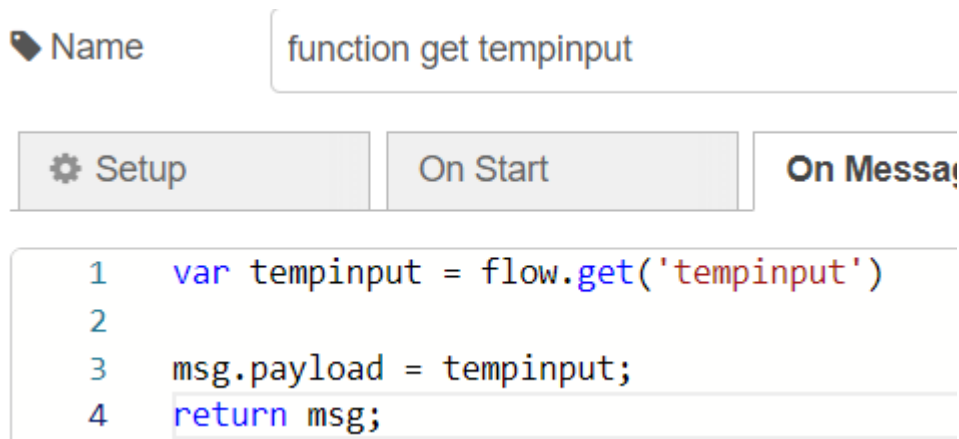
Setup On Start On Message

```

1  var tempinput = msg.payload
2
3  flow.set('tempinput', tempinput)
4
5  return msg;

```

- [function get tempinput](function): được nối với 1 button, khi ấn nút thì sẽ lấy dữ liệu từ tempinput và gửi nhiệt độ mong muốn đến node mqtt out.



- [function convert data temp](function) và [function convert data humi](function) hai hàm này có chức năng tương tự và được code giống nhau, chỉ đổi tên và biến : nhận được dữ liệu nhiệt độ, độ ẩm từ thingspeak và đã được xử lý qua node JSON.

- Kiểm tra Dữ liệu Đầu Vào:

+ Dòng `if (msg.payload && msg.payload.feeds && Array.isArray(msg.payload.feeds))` kiểm tra xem `msg.payload` có tồn tại và có các thuộc tính `feeds` là một mảng hay không.

+ Nếu các điều kiện này đều đúng, có nghĩa là dữ liệu đầu vào từ Thingspeak có cấu trúc chính xác và có thuộc tính `feeds` là một mảng chứa dữ liệu cần xử lý.

-Xử Lý Dữ liệu:

+ Nếu dữ liệu đầu vào có cấu trúc chính xác, mã sẽ tiếp tục xử lý dữ liệu từ mảng `feeds`.

+Dòng `var data = msg.payload.feeds.map(entry => { ... })` sử dụng hàm `.map()` để duyệt qua từng phần tử trong mảng `feeds` và tạo một mảng mới `data` chứa các đối tượng dữ liệu được định dạng lại.

-Định Dạng Lại Dữ liệu:

+ Dòng `return { x: new Date(entry.created_at), y: parseFloat(entry.field1) }`; tạo một đối tượng mới với thuộc tính `x` là thời gian từ `created_at` và thuộc tính `y` là giá trị số từ `field1`. Cả hai giá trị này đều được định dạng lại để phù hợp với việc hiển thị trên biểu đồ.

- Định Dạng Lại Payload:

+ Sau khi đã xử lý và định dạng lại dữ liệu, mã tiếp tục với `msg.payload = [{ series: ["Nhiệt độ"], data: [data] }];`. Điều này định dạng lại payload để phù hợp với yêu cầu của node `"ui_chart"` trong Node-RED Dashboard.

- Trả Về Payload Đã Định Dạng Lại:

+ Dòng `return msg`; trả về payload đã được định dạng lại. Điều này sẽ chuyển dữ liệu đã xử lý qua các node khác trong luồng làm việc.

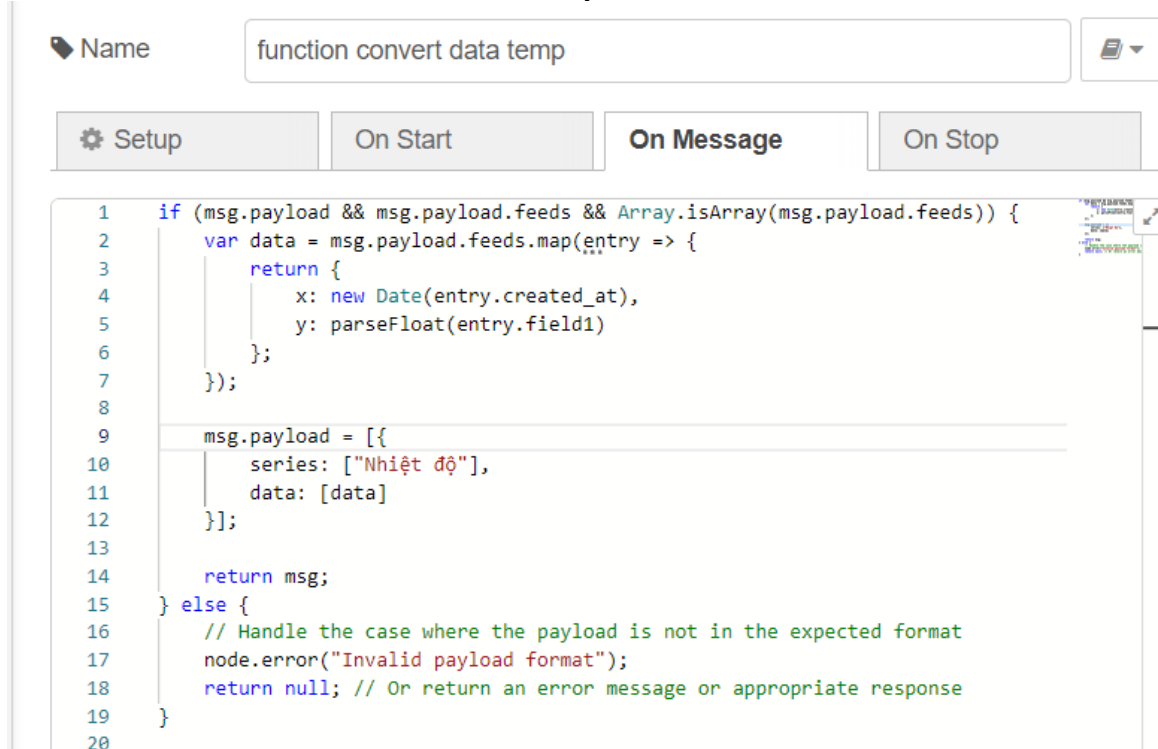
- Xử Lý Lỗi:

+ Trong trường hợp dữ liệu đầu vào không đúng định dạng hoặc thiếu các thuộc tính cần thiết, mã sẽ đi vào phần `else`.

+ Dòng `node.error("Invalid payload format");` tạo một thông báo lỗi để ghi vào bảng điều khiển của Node-RED để theo dõi.

+ Dòng `return null`; hoặc `return null`; trả về giá trị `null` để đặt dừng thực hiện cho các node phía sau trong luồng làm việc.

- Tóm lại, chức năng này thực hiện việc kiểm tra cấu trúc dữ liệu đầu vào, xử lý dữ liệu, và định dạng lại payload để phù hợp với hiển thị trên biểu đồ. Nếu dữ liệu đầu vào không hợp lệ, nó ghi lại một thông báo lỗi và dừng thực hiện các node phía sau.



- [Warning to IFTTT](http request): hàm này sẽ gửi nhiệt độ hiện tại ($t > 40$), yêu cầu tới ifttt để cảnh báo tới điện thoại.
- [read field 1](http request) và [read field 2](http request) 2 hàm này có chức năng giống nhau: hàm này sẽ đọc thông tin 200 dữ liệu gần nhất về nhiệt độ và độ ẩm từ thingspeak để thể hiện lên trên biểu đồ.

- các node Button (Tăng Nhiệt Độ, Giảm Nhiệt Độ, Bật/Tắt): khi ấn vào sẽ gửi 1 chuỗi tới node mqtt out để gửi tới esp để thực hiện các lệnh tăng, giảm, bật tắt điều hòa.
- Node button Refresh: khi ấn vào để thực hiện việc Refresh. Đọc dữ liệu mới nhất từ thingspeak(cloud)
- [nhietdo_doam](thingspeak): node gửi dữ liệu lên thingspeak. cần cài đặt thư viện để sử dụng

