**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG – HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

A picture containing logo

Description automatically generated----- 🙡 🕮 🙣 -----

**BÁO CÁO CÁ NHÂN**

**Tên đề tài:** *Dự án Thùng rác thông minh*

|  |
| --- |
| Cao Xuân Nam |
| Đặng Hoài Thương |

**Môn học**: Vật lý cho Công nghệ thông tin

**Giảng viên:**

**Sinh viên thực hiện:** Dương Trung Nghĩa

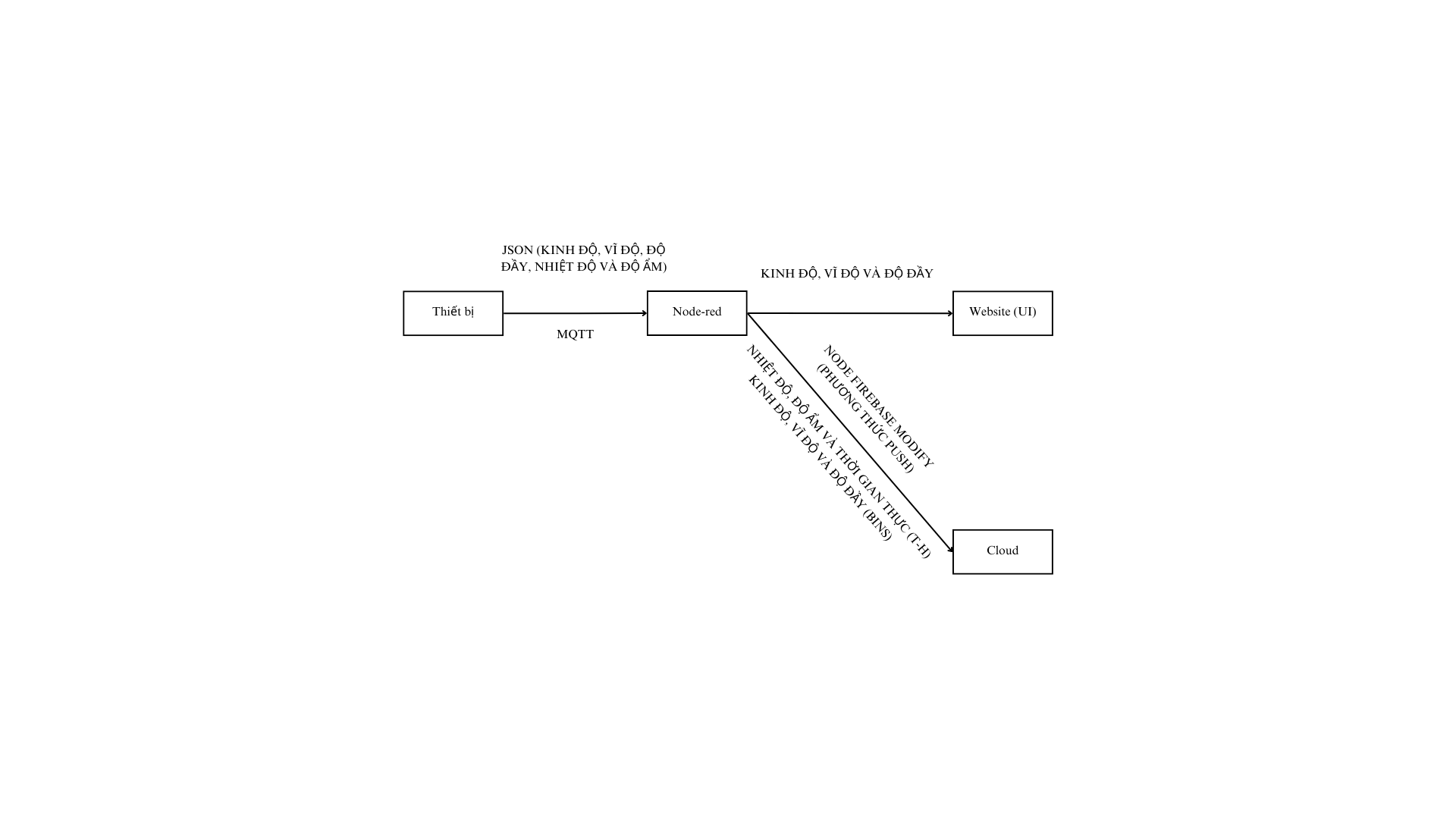
**MSSV:** 22127293

**Lớp:** 22CLC05

**Thành phố Hồ Chí Minh, năm 2024**

**CÁC YÊU CẦU CĂN BẢN**

**-** Luồng xử lý dữ liệu từ thiết bị lên website:



Hình 1: Luồng xử lý thiết bị lên website

- Mô tả chi tiết luồng:

* Sau khi đọc xong các giá trị các thiết bị cảm biến cần thiết, tạo kiểu dữ liệu json trên wokwi chứa 5 giá trị cảm biến.
* Truyền lên Node-red bằng giao thức truyền thông MQTT (hàm wifiConnect và hàm mqttConnect đảm bảo rằng thiết bị ESP32 sẽ tự động kết nối với mạng Wi-Fi và máy chủ MQTT, đồng thời chuẩn bị để nhận các tin nhắn từ các topic cần thiết).
* Trên Node red dùng node “MQTT in” với topic tương ứng nhận dữ liệu json, sau đó tách ra thành 2 json, chứa thông tin thùng rác (kinh độ, vĩ độ, độ đầy) và chứa sự thay đổi (nhiệt độ và độ ẩm) thực hiện các công việc riêng biệt.
* Với thông tin thùng rác, thêm vào list json có sẵn (tạo cho thấy nhiều thùng rác trên bản đồ) sau đó coi có tồn tại trên firebase chưa nếu rồi thì cập nhật lại độ đầy còn không thì push thùng rác đó lên để hiển thị trên bản đồ và trong trang “Schedule” vẽ biểu đồ phần trăm độ đầy.
* Với sự thay đổi nhiệt độ và độ ẩm, thêm khóa thời gian thực chứa dữ liệu dạng Date vào, sau đó đẩy lên cloud firebase để có thể xem lịch sử dữ liệu đã lưu trên cloud trên giao diện web lúc sau (yêu cầu 6).

- Phần thiết bị: đã thực hiện đúng yêu cầu (nối mạch đúng quy tắc, đọc ghi dữ liệu từ thiết bị và bật tắc thiết bị đúng cách)

- Phần Website: Hiển thị được bản đồ hiện các thùng rác, vẽ các biểu đồ để đưa ra quyết định lập lịch diệt khuẩn cho thùng rác.

- Công việc đã làm: truyền vào kinh độ, vĩ độ hiển thị lên bản đồ với các marker đánh dấu phân biệt, một vài nút tương tác chuyển trang cơ bản, truyền vào kinh độ, vĩ độ hiển thị lên bảng có thể tương tác trực tiếp, vẽ biểu đồ phần trăm độ đầy tương ứng từng thùng rác trong trang “Schedule” tùy vào việc tương tác với bảng, hoàn thành các yêu cầu cơ bản và nâng cao.

- Đánh giá nhóm: Thực hiện tốt các yêu cầu thầy giao, tích cực tìm hiểu tài liệu phục vụ đồ án.

- Đánh giá bản thân: Thực hiện tốt các yêu cầu được chia trong nhóm, hiểu rõ luồng xử lý từ thiết bị lên website và có hiểu luồng xử lý từ website về thiết bị nhất định.

**CÁC YÊU CẦU NÂNG CAO**

Yêu cầu 1: Sử dụng thêm thiết bị INPUT trong nội dung giảng dạy.

- Thiết bị: DHT22. Công dụng để đọc nhiệt độ và độ ẩm bên trong thùng rác.

- Thiết bị: PIR. Công dụng để mở không chạm tay vào nắp thùng.

Yêu cầu 2: Sử dụng thêm thiết bị OUTPUT trong nội dung giảng dạy.

- Thiết bị: Servo. Công dụng để bật/ đóng nắp thùng rác theo điều kiện độ đầy.

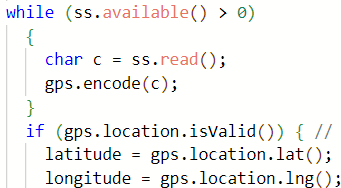
Yêu cầu 3: Sử dụng thêm thiết bị INPUT ngoài nội dung giảng dạy.

- Thiết bị: cảm biến GPS. Công dụng để ghi nhận giá trị kinh độ và vĩ độ vị trí thùng rác.

- Nhóm có tham khảo việc sử dụng thiết kế chip trên wokwi để giả lập một “gps-neo6m Breakout” với các chân kết nối bao gồm:

* “VCC": Chân cung cấp nguồn điện cho module.
* "GND": Chân nối đất, kết nối với GND của hệ thống.
* "RX": Chân nhận dữ liệu (Receive), kết nối với chân TX của vi điều khiển hoặc thiết bị khác.
* "TX": Chân truyền dữ liệu (Transmit), kết nối với chân RX của vi điều khiển hoặc thiết bị khác.

- Trong file giả lập chứa một mảng các câu lệnh NMEA 0183 được định nghĩa sẵn, chứa thông tin giả lập về dữ liệu GPS như tọa độ và trạng thái của vệ tinh.



Hình 2: Đọc và giải mã lấy kinh độ và vĩ độ

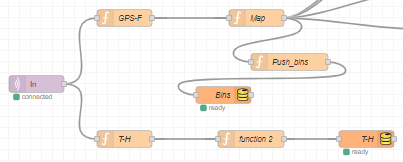
- Giải mã nội dung đọc trong file giả lập, kiểm tra có đọc được hay không và sau đó lưu kinh độ và vĩ độ (các hàm đã được cài đặt khi tham khảo).

Yêu cầu 5: Có lưu trữ dữ liệu cảm biến lên cloud.

- Nhóm chúng em sử dụng Firebase để lưu trữ dữ liệu cảm biến “Nhiệt độ và Độ ẩm” vì cần hiển thị lịch sử thay đổi dữ liệu cho phần sau (yêu cầu 6), dữ liệu cảm biến “Độ đầy” để dễ dàng thao tác cập nhật biểu đồ hiển thị do có nhiều thùng rác khác nhau, dữ liệu cảm biến “Kinh độ và vĩ độ” để phân biệt giữa các thùng rác khác nhau cũng để hiển thị các điểm trên world map của thư viện “node-red-contrib-web-worldmap”.

- Để đẩy dữ liệu lên Firebase sử dụng thư viện “node-red-contrib-firebase” thông qua node “firebase modify” theo phương thức push (đẩy các dữ liệu lên cloud theo dạng json thông qua msg.payload).

- Giao thức MQTT gửi qua json chứa 5 dữ liệu cảm biến kể trên.



Hình 3: Nhận từ MQTT chia thành 2 luồng để lưu trên cloud.

|  |  |
| --- | --- |
|  |  |

Hình 4: Dữ liệu cảm biến lưu trữ trên cloud.

Yêu cầu 8: Có thể thông báo cho người dùng bằng email.

- Khi thực hiện đăng ký, nhóm em có lưu dữ liệu “email” lên cloud “firebase”.

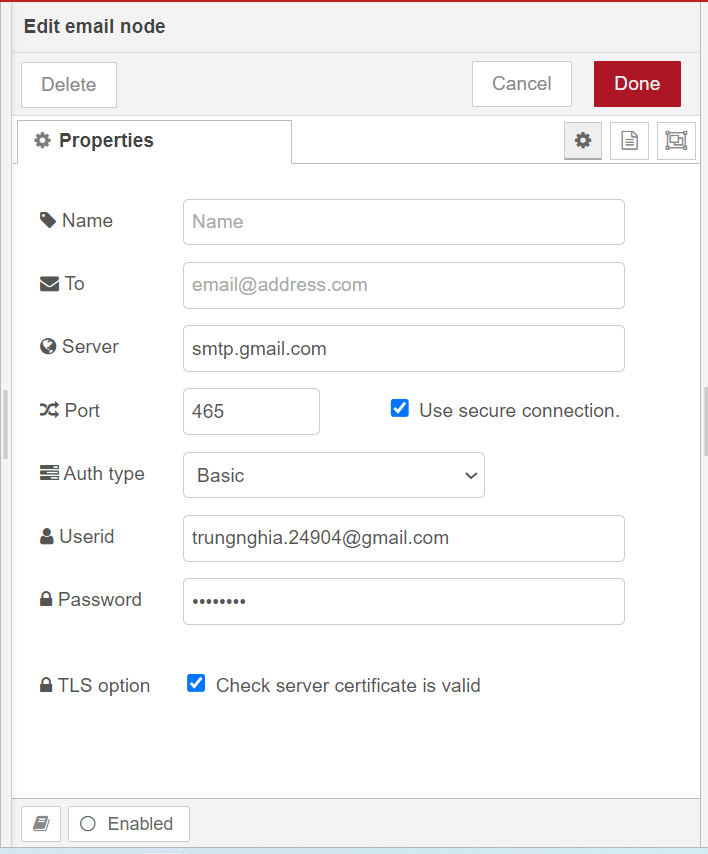
- Nhóm sẽ gửi đồng thời đoạn nội dung là địa chỉ thực dựa trên kinh độ và vĩ độ về điện thoại (yêu cầu 7) và email thông qua thư viện “node-red-node-email”.



Hình 5: Luồng xử lý gửi email về cho người dùng.

- Tương tác trên “table” để lấy dữ liệu “Kinh độ và Vĩ độ” sau đó sử dụng web thứ ba để lấy địa chỉ thực (gán url "https://nominatim.openstreetmap.org/reverse?format=json&lat=" + kinh độ + "&lon=" + vĩ độ) sau đó gửi http request đến url được trả về file json. Trong đó, lấy phần “display\_name” gửi qua node email với format msg truyền vào là “.to” là email tương ứng với tài khoản đăng nhập, “.topic” là header gửi về (“Địa chỉ thùng rác”) và “.payload” là nội dung truyền về.

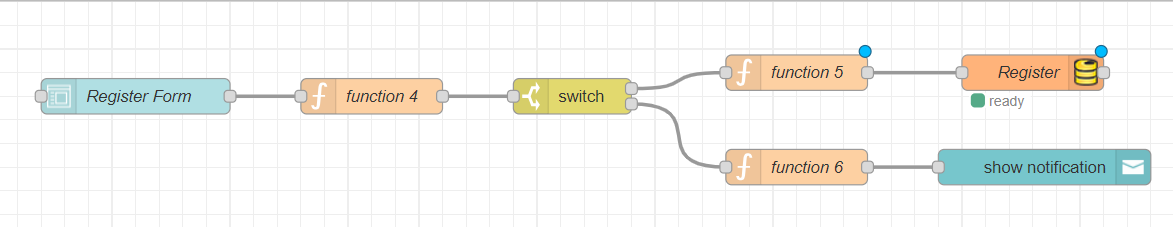
- Dùng email cá nhân làm host gửi tới các tài khoản khác. Phải xác thực 2 lớp và tạo mật khẩu ứng dụng



Hình: Cửa sổ node email.

- Điền Userid host gửi mail và password ứng dụng được tạo ra.

Yêu cầu 9: Có giao diện web cho phép đăng ký tài khoản (ghi thành công thông tin tài khoản vào database/cloud/file).



Hình 6: Luồng trang đăng ký.

- Tạo form ui bằng thư viện “node-red-dashboard” để nhập tên đăng nhập và mật khẩu để kiểm tra khi đăng nhập, email để gửi email “địa chỉ thùng rác” cho người dùng và điện thoại.

- Hàm “function 4” dùng để tạo biến global lưu tạm đối tượng đăng ký, kiểm tra từng đối tượng được lưu trên cloud Firebase xem có từng đăng ký hay chưa? Dùng “switch” để chia luồng theo “status” là đăng ký thành công hay không? Nếu thành công, thì theo luồng “function 5” lấy biến global lưu đối tượng đăng ký lên cloud Firebase bằng node “firebase modify” bằng phương thức push một đối tượng chứa các dữ liệu trên, còn rồi thì theo luồng “function 6” để in thông báo thất bại lên góc phải màn hình trong vòng 3 giây.