

TỔ CHỨC THỰC HIỆN

I. Yêu cầu của dự án Nhóm 11

Yêu cầu	Chức năng, Thông số,...	Mức độ ưu tiên
<ul style="list-style-type: none"> - Dải đo: $25^{\circ}\text{C} \div 125^{\circ}\text{C}$ - Độ chính xác: $\pm 1^{\circ}\text{C}$ - Độ phân giải hiển thị: 0.1°C 	<ul style="list-style-type: none"> • Đo nhiệt độ tại mỗi lò ấp trứng. • Sử dụng màn hình LCD để hiển thị tại mỗi lò 	1
Nguồn pin, thời gian hoạt động của thiết bị là 4h (nâng cao: 8h). Pin có thể sạc trực tiếp trên máy hoặc tháo ra ngoài	<ul style="list-style-type: none"> • Sử dụng Pin Li – Po dung lượng xx • Tính toán công suất tiêu thụ (min – max) • Pin sạc trực tiếp (tính toán suy hao) 	2
<ul style="list-style-type: none"> - Kích thước (dự kiến): $70 \times 50 \times 100 \text{ mm}$ (kiểu trụ để dễ cầm tay) - Trọng lượng (dự kiến): $< 150\text{g}$. 	<ul style="list-style-type: none"> • Chưa tính toán (phụ thuộc kích thước và vị trí đặt cảm biến trong lò) 	3
- Thời gian đo một mẫu : $< 20\text{s}$. (nâng cao $< 5\text{s}$)	<ul style="list-style-type: none"> • (thời gian tổng cộng đọc cảm biến, lưu EEPROM, truyền BLE, hiển thị) • Thời gian này phụ thuộc vào công nghệ truyền tin và module sử dụng, cảm biến • Test truyền nhận • Có/không thay đổi chu kỳ đo qua OTA 	1
- Kết nối máy tính: RF Khoảng cách truyền trong phạm vi 20m từ hệ thống đo đến trạm thu RF có nối nguồn và mạng	<ul style="list-style-type: none"> • Sử dụng công nghệ BLE: thông tin của các node được đưa về gateway và đẩy lên server • Có kết nối máy tính: Wi-fi • Back up: Nếu gateway lỗi 	1
Quản lý tối thiểu cho 10 thiết bị đo. (nâng cao: phương án mở rộng số thiết bị với khoảng cách từ thiết bị đến trạm tiếp nhận có thể lên tới 100m)	<ul style="list-style-type: none"> • Tìm hiểu cách định tuyến và gia nhập mạng (plug and play) • Đưa ra thuật toán truyền dữ liệu • Quản lý 10 thiết bị Phát hiện được thiết bị lỗi và thiết bị thêm vào mạng? • Dự kiến cấu trúc mạng, topology 	1
- Phần mềm máy tính: thu thập giá trị đo từ thiết bị đo, quản lý dữ liệu, xuất báo cáo dạng excel, giao diện theo mẫu thống nhất.	<ul style="list-style-type: none"> • Sử dụng platform Thingsboard: build phần mềm để hiển thị • Chi tiết giao diện hiển thị realtime/lưu trữ rồi hiển thị/lich sử/xuất báo cáo dữ liệu 	3
Có nút bấm bắt đầu đo; Đèn LED báo ngưỡng nhiệt độ (3 LED); Các	<ul style="list-style-type: none"> • $37.5 - 37.7 - 38$ • $36.5 - 37 - 37.5$ • Không có nút bấm bắt đầu đo 	2

Commented [DBN21]: Thêm cột đạt được/không
Tính toán độ phân giải
Cần nhắc dùng led 7 thanh hiển thị

Commented [DBN22R1]: Tính lại công suất tiêu thụ chế độ truyền/nhận

ngưỡng nhiệt độ có thể cập nhật từ máy tính	• Có nút bấm reset	
OTA (nâng cao)		3

II. Giới thiệu thành viên của dự án

	Họ và tên: Phạm Hải Đăng MSSV: 20191729 Phụ trách công việc
	Họ và tên: Đinh Bảo Ngân MSSV: 20191984 Phụ trách công việc
	Họ và tên: Nguyễn Đình Khánh MSSV: 20191906 Phụ trách công việc

III. Kế hoạch thực hiện chung của dự án

STT	Nội dung	Kết quả cần đạt	Thời gian (tuần)	Ghi chú
1	Tìm hiểu về các bài toán liên quan	<ul style="list-style-type: none">- Báo cáo tổng quan, đánh giá- Tìm hiểu chung về hệ thống và dự án: đưa ra yêu cầu cụ thể, công nghệ truyền thông	T1 - T5	Phải có nhật ký và quy tắc báo cáo
2	Lên phương án sơ bộ	<ul style="list-style-type: none">- Sơ đồ khối chức năng- Kiến trúc hệ thống- Tìm hiểu các tài liệu liên quan- Lựa chọn phần cứng	T5 – T10	Theo dõi tiến độ
3	Lập trình các chức năng đề ra	<ul style="list-style-type: none">- Hiểu được các ví dụ lập trình và xây dựng chương trình theo yêu cầu	T11 – T15	
4	Hoàn thiện phần lập trình và phần cứng của thiết bị	<ul style="list-style-type: none">- Kết hợp các khối chức năng thành mô hình hoàn chỉnh- Làm mạch phần cứng	T15 – T18	
5	Hoàn thiện sản phẩm, trình bày báo cáo	<ul style="list-style-type: none">- Kiểm tra các tính năng, đóng vỏ, viết báo cáo	T18 – T20	

IV. Kế hoạch và nội dung thực hiện của từng thành viên

Phạm Hải Đăng

Nội dung	Mô tả (tính năng, thông số,...)	Kết quả cần đạt	Thời gian thực hiện (theo tuần)	Ghi chú
Tìm hiểu về các bài toán liên quan			T1 – T5	
			T5 – T8	
			T9	
			T10	
			T11	
			T12	
			T13	
			T14	
			T15 – T16	
			T17	
			T18	
			T19	
			T20	

Đinh Bảo Ngân

Nội dung	Mô tả (tính năng, thông số,...)	Kết quả cần đạt	Thời gian thực hiện (theo tuần)	Ghi chú
			T1 – T5	
			T5 – T8	
			T9	
			T10	
			T11	
			T12	
			T13	
			T14	
			T15 – T16	
			T17	
			T18	
			T19	
			T20	

Nguyễn Đình Khánh

Nội dung	Mô tả (tính năng, thông số,...)	Kết quả cần đạt	Thời gian thực hiện (theo tuần)	Ghi chú
			T1 – T5	
			T5 – T8	
			T9	
			T10	

			T11	
			T12	
			T13	
			T14	
			T15 – T16	
			T17	
			T18	
			T19	
			T20	

V. Tự đánh giá tỷ lệ đóng góp của từng thành viên trong dự án theo kế hoạch (trước khi thực hiện, thực hiện trong khi lên kế hoạch thực hiện dự án)

Người thực hiện	Tỷ lệ	Giải quyết được những vấn đề gì của dự án (cần ghi rõ để có cơ sở đánh giá tỷ lệ)
Phạm Hải Đăng	Xx%	
Đinh Bảo Ngân	Yy%	
Nguyễn Đình Khánh	Zz%	

NỘI DUNG THỰC HIỆN

BLE bật tắt theo chu kỳ: Bật khi cần trao đổi dữ liệu, tắt khi ngủ

Tốc độ trao đổi dữ liệu lên đến 2Mbps

3 lớp: class 1: 100mW, class 2:

Broadcast: không cần kết nối giữa các thiết bị, broadcaster phát bản tin và nhận dữ liệu. 37 kênh dữ liệu và 3 kênh quảng bá.

Chọn công nghệ gì: tìm hiểu phần truyền tin đó.

VI. Nhật ký

(ghi ngắn gọn, mỗi tuần ghi không quá 1 trang A4 đánh máy)

Tuần 4

Tổ chức

- Địa điểm: OFFLINE
- Thời gian: 04/11/2022
- Tham dự: 3/3
- Không tham dự: Lý do:.....

Nội dung cuộc họp:

- Nội dung 1: Làm rõ yêu cầu của đề tài

Kết luận của buổi họp: Mỗi người tự về tìm hiểu các mô hình đã có ở trên thị trường để đưa ra được mục tiêu cho đề tài

Tuần 5

Tổ chức

- Địa điểm: ONLINE
- Thời gian: 11/11/2022
- Tham dự: 3/3
- Không tham dự: Lý do:.....

Nội dung cuộc họp:

- Nội dung 1: Mỗi thành viên trình bày nội dung đã tìm hiểu
- Nội dung 2: Lựa chọn mô hình

Kết luận của buổi họp:

Tuần 6

Tổ chức

- Địa điểm: OFFLINE
- Thời gian: 16/11/2022
- Tham dự: 3/3
- Không tham dự: Lý do:.....

Nội dung cuộc họp:

- Nội dung 1: Phân công tìm hiểu về các nội dung sau: phần cứng thiết bị, mô hình mạng, công nghệ BLE

Kết luận của buổi họp: Mỗi thành viên tìm hiểu các nội dung được phân công và trình bày vào buổi họp sau

Tuần 7

Tổ chức

- Địa điểm: OFFLINE
- Thời gian: 25/11
- Tham dự: 3/3
- Không tham dự: Lý do:.....

Nội dung cuộc họp:

- Nội dung 1: Báo cáo các nội dung đã tìm hiểu

Kết luận của buổi họp:....

Tuần 8

Tổ chức

- Địa điểm: OFFLINE
- Thời gian: 25/11
- Tham dự: 3/3
- Không tham dự: Lý do:.....

Nội dung cuộc họp:

- Nội dung 1: Báo cáo các nội dung đã tìm hiểu

Tuần 9

Tổ chức

- Địa điểm: OFFLINE
- Thời gian: 25/11
- Tham dự: 3/3
- Không tham dự: Lý do:.....

Nội dung cuộc họp:

- Nội dung 1: Báo cáo các nội dung đã tìm hiểu

VII. Phân tích các yêu cầu của dự án

1. Tổng quan đề tài
2. Đặt vấn đề, giới thiệu về ứng dụng (tại sao cần lồng ấp trứng). Các chức năng của lồng
3. Mục tiêu cụ thể:
 - Trung bình, chu kỳ ấp trứng gà là 21 ngày. Trong 17 ngày đầu, nhiệt độ cần được duy trì trong khoảng 37,5 - 38; 4 ngày cuối là 36,5 - 37,5
 - Xây dựng những node cảm biến đặt tại lò ấp nhằm giám sát nhiệt độ của lò:
 - Duy trì trong khoảng 37,5 - 38°C hoặc 36,5 – 37,5°C và hiển thị nhiệt độ hiện tại trong lò ở nút cảm biến, độ phân giải là 0.1°C
 - Sử dụng pin Li – po có thể sạc lại
 - Kích thước dự kiến...
 - Thời gian đo mẫu: 10 giây
 - Khoảng cách truyền trong phạm vi 20m, từ node tới trạm thu RF có nguồn và mạng: sử dụng công nghệ Bluetooth
 - Quản lý tối thiểu 10 thiết bị đo
 - Xây dựng phần mềm/web giao tiếp với người dùng: thu thập giá trị đo từ thiết bị đo, quản lý dữ liệu, xuất báo cáo dạng excel
 - Có nút bấm bắt đầu đo (công tắc on/off), đèn led báo 3 ngưỡng nhiệt độ và 3 ngưỡng này có thể cài đặt lại từ máy tính

VIII. Tìm hiểu các nghiên cứu, dự án liên quan

- Review máy ấp trứng:
Trên thị trường hiện tại, có 1 số máy ấp trứng tự động. Các máy ấp trứng nhận thông tin từ cảm biến nhiệt, sau đó phát tín hiệu điều khiển module cấp nhiệt, ngoài ra máy ấp trứng còn có thêm module đặt nhiệt độ ấp trứng, thời gian ấp trứng.
Trên thị trường có 1 số máy ấp trứng sau:
 - ❖ Máy ấp trứng GTL-1000
 - Tự động hoàn toàn 100%, công suất tối đa 1000 trứng, đảo trứng tự động (chế độ hẹn giờ), phun ẩm và nhiệt độ tùy chỉnh tự động đóng khi quá nhiệt độ.
 - Điện áp : 220VAC
 - Công suất tiêu thụ: 10kW/1 kỳ
 - Phun ẩm: Tùy chỉnh, tự động điều chỉnh đóng cắt khi quá % quy định
 - Hệ thống cung cấp nhiệt: Bóng nhiệt Halozen chuyên dùng cho máy ấp trứng
 - Đảo trứng: Tự động hẹn giờ từ 30p – 120h
 - Nhiệt độ được điều khiển tự động, ổn định bằng vi xử lý
 - Tạo độ ẩm tự động
 - Đảo trứng tự động
 - Có thể ấp theo chế độ đa kỳ
 - ❖ Máy ấp trứng OXY GTL 240K
 - Là loại máy ấp trứng đảo trứng bằng khí
 - Điện áp: 220VAC
 - Công suất tiêu thụ: 2kW/ 1 Kỳ
 - Hệ thống cấp nhiệt: điện trở
 - Nhiệt độ được điều khiển tự động bằng vi xử lý
 - Tự động trộn khí oxy
 - Có thể ấp theo chế độ đa kỳ, hoặc đơn kỳ
- Review 1 bài báo:

Design and implementation of chicken egg incubator for hatching using IoT
Article in International Journal of Computational Science and Engineering · June 2021

- Các tiêu chí cần quan tâm: nhiệt độ, độ ẩm, thông gió và hệ thống đảo trứng
- 3 nhiệt độ cho mục đích thử nghiệm là $T1 = 36,5^{\circ}\text{C}$, $T2 = 37,5^{\circ}\text{C}$ và $T3 = 38^{\circ}\text{C}$
- Sử dụng arduino nano và cảm biến LM35, DHT22, thời gian phản hồi từ 5 - 10s
- Trong 17 ngày đầu, nhiệt độ cần được duy trì trong khoảng 37,5 - 38; 4 ngày cuối là 36,5 - 37,5
- DS18B20 được dùng để duy trì nhiệt độ cần thiết

IX. Lựa chọn giải pháp và lên phương án thiết kế

- Mô hình kiến trúc
- Sơ đồ khối chức năng
- Nguyên lý hoạt động
- Tìm hiểu về HTTP -> giao tiếp gateway với Web server

Khái niệm

HTTP (HyperText Transfer Protocol) là một giao thức lớp ứng dụng cho các hệ thống thông tin siêu phương tiện phân tán, cộng tác. Là nền tảng của truyền thông dữ liệu cho world wide web, nơi siêu văn bản tài liệu bao gồm các siêu liên kết đến các tài nguyên khác mà người dùng có thể dễ dàng truy cập.

Thành phần

- HTTP Session (phiên làm việc của HTTP) là một khái niệm phổ biến được dùng trong lập trình web có kết nối với database. Đặc biệt các chức năng như đăng nhập, đăng xuất người dùng sẽ khó có thể thực hiện nếu không sử dụng session. Ví dụ: email, tài khoản ngân hàng, ...
- HTTP Request methods: chỉ phương thức để được thực hiện trên nguồn được nhận diện bởi Request-URI đã cung cấp:
- GET: được sử dụng để lấy lại thông tin từ server đã cung cấp bởi sử dụng một URI đã cung cấp. Các yêu cầu sử dụng GET chỉ nhận dữ liệu và không có ảnh hưởng gì đến dữ liệu.
- HEAD: tương tự như GET, nhưng nó truyền tải dòng trạng thái và khu vực Header.
- POST: một yêu cầu POST được sử dụng để gửi dữ liệu tới Server, ví dụ: thông tin khách hàng, file tải lên, ... bởi sử dụng các mẫu HTML.
- PUT: thay đổi tất cả các đại diện hiện tại của nguồn mục tiêu với nội dung được tải lên.
- DELETE: Gỡ bỏ tất cả các đại diện hiện tại của nguồn mục tiêu bởi URI.
- CONNECT: Thiết lập một tunnel tới Server được xác định bởi URI đã cung cấp.
- OPTIONS: Miêu tả các chức năng giao tiếp cho nguồn mục tiêu.
- TRACE: Trình bày một vòng lặp kiểm tra thông báo song song với path tới nguồn mục tiêu.
- HTTP Response: khi nhận và phiên dịch một HTTP Request, Server sẽ gửi tín hiệu phản hồi là một HTTP Response bao gồm các thành phần sau:
- Một dòng trạng thái (Status-Line).

- Không hoặc nhiều hơn các trường Header (General|Response|Entity) được theo sau CRLF.
- Một dòng trống chỉ dòng kết thúc của các trường Header.
- Một phần thân thông báo tùy ý.
- Ví dụ Server response:

```
HTTP/1.1 200 OK
Date: Mon, ngày 23 tháng 5 năm 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close

<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

- HTTP Status Code: Mã trạng thái HTTP được server phản hồi lại mỗi khi nhận được http request. Yếu tố Status-code là một số nguyên 3 kí tự, trong đó ký tự đầu tiên của mã hóa trạng thái định nghĩa hạng (loại) phản hồi và 2 ký tự cuối không có bất cứ vai trò phân loại nào.
- HTTP Response Header Fields: Các trường Header phản hồi cho phép Server truyền thông tin thêm về phản hồi mà không thể được đặt trong dòng Status-Line. Những trường Header này cung cấp thông tin về Server và về truy cập từ xa tới nguồn được xác định bởi Request-URI:

```
response-header = Accept-Ranges;
                  | Age      ;
                  | ETag     ;
                  | Location;
                  | Proxy-Authenticate;
                  | Retry-After;
                  | Server;
                  | Vary     ;
                  | WWW-Authenticate;
```

1. Lựa chọn công nghệ truyền thông

1.1. Tìm hiểu chung

- Bluetooth là 1 công nghệ truyền thông sử dụng băng tần ISM từ 2.4GHz đến 4.4835GHz hoặc từ 2.402Ghz đến 2.48GHz để trao đổi dữ liệu. Bluetooth đạt tốc độ truyền dữ liệu tối đa khoảng 721kps và chỉ kết nối được ở khoảng cách

gần (nhỏ hơn 10m). Từ phiên bản 4.x trở lên, Bluetooth được chia làm 2 hướng phát triển và BLE là 1 trong các hướng đó.

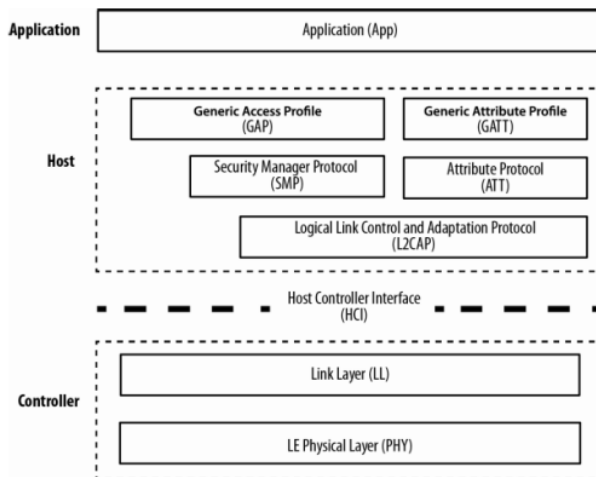
- BLE được tích hợp vào Bluetooth 4.x bởi Bluetooth Special Interest Group (Bluetooth SIG) vào năm 2010 với trọng tâm là thiết kế một chuẩn truyền tiêu thụ năng lượng thấp nhất có thể, đồng thời tối ưu về mặt chi phí cũng như độ phức tạp khi chế tạo, kết nối.

1.2. Mô hình mạng BLE

- Một thiết bị BLE có thể giao tiếp theo 2 cơ chế: quảng cáo và thiết lập liên kết. Mỗi cơ cấu đều có ưu nhược điểm riêng nhưng chúng đều được khai báo ở phần thiết lập truy cập (GAP) của lớp Host. Khi sử dụng ở cơ chế quảng bá, người dùng có thể gửi dữ liệu đến thiết bị phát hiện được trong dải tần quét và trong phạm vi phát hiện. Đây là cơ chế cực kì hữu dụng nếu muốn truyền lượng nhỏ thông tin đến nhiều thiết bị cùng 1 lúc, tuy nhiên không có cơ chế bảo mật nên không phù hợp với những dữ liệu nhạy cảm.
- Nếu người dùng muốn bảo mật thông tin hay gửi nhiều thông tin hơn đến một thiết bị thì cấu hình mạng theo kiểu thiết lập kết nối lại hữu dụng hơn. Thiết bị trong kết nối được chia làm 2 vai trò riêng biệt:
 - Thiết bị trung tâm (chủ)
Liên tục quét để tìm thiết bị muốn kết nối (thiết bị nào muốn kết nối sẽ gửi gói tin "muốn kết nối" ở dải tần định sẵn. Nếu tìm thấy dải tin phù hợp, nó sẽ khởi tạo kết nối. Khi kết nối được khởi tạo, thiết bị trung tâm sẽ quản lý xung nhịp và bắt đầu trao đổi dữ liệu theo chu kì.
 - Thiết bị ngoại vi (tớ)
Khi kết nối với thiết bị trung tâm sẽ quảng bá gói tin theo chu kì đồng thời đợi gói tin trả về để biết có kết nối hay không. Khi kết nối với thiết bị trung tâm, thiết bị ngoại vi sẽ trao đổi thông tin theo chu kì và xung nhịp nhưng lại không bị hạn chế về số lượng bản tin và bị áp đặt mức độ ưu tiên. Một thiết bị có thể vừa là chủ, vừa là tớ. Một chủ có thể kết nối với nhiều tớ và tớ cũng có thể kết nối với nhiều chủ.

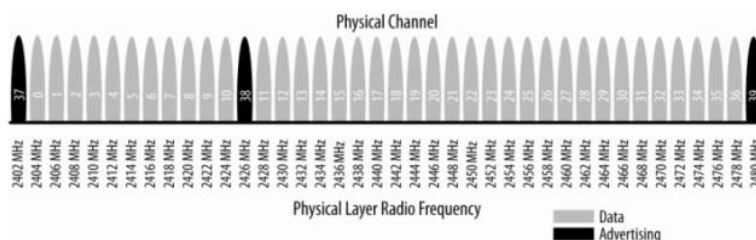
1.3. Sơ đồ khối của một thiết bị sử dụng BLE

- Thiết bị BLE được chia làm 3 khối cơ bản và trong mỗi khối đều có một hoặc nhiều lớp sử dụng giao thức khác nhau để thực hiện từng chức năng cụ thể
 - Khối ứng dụng cũng giống như hầu hết các hệ thống khác, khối này chứa các hàm logic, giao diện người dùng và xử lý tất cả dữ liệu liên quan đến use-case mà ứng dụng triển khai. Kiến trúc của nó tùy thuộc vào từng ứng dụng cụ thể.
 - Khối chủ chịu trách nhiệm về những giao thức cao hơn để lập trình viên có thể: thiết lập vai trò, quản lý cơ chế kết nối, bảo mật, định nghĩa dữ liệu và cách tổ chức dữ liệu... Phần cốt lõi của BLE là phải nắm được các cách định nghĩa và tổ chức dữ liệu, cái mà được xây dựng dựa trên thuộc tính.
 - Khối điều khiển sẽ chịu trách nhiệm về giao thức cấp thấp liên quan đến phần vật lý như: ăng ten, bộ điều chế/ giải mã điều chế tín hiệu tương tự và tín hiệu tương tự thành số và ngược lại, hoàn thiện khung bản tin,...



1.4. Tần số hoạt động của BLE

- BLE sử dụng dải tần 2.40 GHz – 2.4835 GHz để trao đổi dữ liệu và chia dải tần này thành 40 kênh truyền. Trong đó có 3 kênh 37, 38, 39, được gọi là kênh quảng cáo (Advertising) và dùng để thiết lập kết nối cũng như gửi thông tin quảng bá. 37 kênh còn lại (từ 0 đến 36) dùng để truyền dữ liệu sau khi kết nối được thiết lập. Điều này là mấu chốt khiến cho 1 thiết bị trung tâm có thể giao tiếp với nhiều thiết bị ngoại vi cùng lúc và ngược lại.



1.5. Quá trình thiết lập kết nối giữa các thiết bị có BLE

- Như đã biết ở trên, một thiết bị ngoại vi muốn kết nối với thiết bị trung tâm thì sẽ phát ra gói tin quảng cáo (Advertising) ra ngoài không gian.
- Vậy thiết bị ngoại vi sẽ liên tục phát ra các gói tin quảng cáo và thiết bị trung tâm thì liên tục quét theo các tần số để tìm kiếm thiết bị muốn kết nối?

Tất nhiên là không vì như thế sẽ rất tốn năng lượng và sẽ không bao giờ được gọi là Bluetooth năng lượng thấp. Thực tế, thiết bị ngoại vi sẽ được thiết lập một chu kỳ phát quảng cáo (Advertising Interval). Trong khoảng thời gian này, BLE ngoại vi sẽ phát các gói tin quảng cáo trên cả 3 kênh 37, 38, 39 để tăng khả năng nhận được gói tin của thiết bị trung tâm. Tương tự như vậy, thiết bị trung tâm cũng có một chu kỳ quét (Scanning Interval). Khi đó, BLE trung tâm sẽ thức dậy và quét thông tin quảng cáo trên cả 3 kênh 37, 38, 39. Để tránh lãng phí năng lượng, BLE trung tâm có 2 kiểu quét

gói tin quảng cáo: thụ động và chủ động. Với kiểu quét gói tin thụ động, BLE trung tâm đơn giản chỉ nhận các gói tin. Do đó BLE ngoại vi sẽ không biết BLE trung tâm đã nhận được tin quảng cáo và vẫn liên tục phát thông tin quảng cáo đến BLE trung tâm đến khi thiết lập được kết nối. Điều này gây tổn hao năng lượng không cần thiết trên BLE ngoại vi nhưng vẫn thường được sử dụng ở một số ứng dụng đơn giản. Còn với kiểu quét chủ động, BLE trung tâm khi quét được gói tin quảng cáo thì sẽ trả 1 gói tin thông báo (gói tin này hoàn toàn không chứa bất kì dữ liệu/yêu cầu đặc biệt nào, chỉ nhằm mục đích thông báo đã quét được tin quảng cáo) về BLE ngoại vi. Khi nhận được gói tin quảng cáo từ BLE ngoại vi “phù hợp” nào, BLE trung tâm sẽ gửi gói yêu cầu thiết lập kết nối đến BLE ngoại vi đó. Gói yêu cầu kết nối này sẽ chứa thông tin của chu kỳ kết nối mà 2 BLE phải tuân theo.

1.6. Cách thức BLE trung tâm đọc/ghi dữ liệu từ BLE ngoại vi

- Tóm tắt sơ lược lý thuyết về lớp “Cấu hình các thuộc tính chung” (Generic Attribute Profile- GATT) do lớp này quyết định chính đến cách lưu trữ cũng như cách truy xuất dữ liệu trong BLE. Cấu trúc dữ liệu ứng dụng trong lớp GATT được tổ chức theo mô hình:

Dịch vụ - Đặc điểm - Thuộc tính

Hay

Service – Characteristic - Attribute

- Thuộc tính là thực thể dữ liệu nhỏ nhất được xác định bởi GATT. Chúng là các thông tin được định địa chỉ rõ ràng và có thể chứa dữ liệu hoặc các siêu dữ liệu (°C, độ F, đơn vị đo áp suất...). Do GATT chỉ làm việc với các thuộc tính nên tất cả mọi thông tin đều được tổ chức dưới dạng này. 1 thuộc tính bao gồm: địa chỉ thuộc tính (được người lập trình quy định trong từng ứng dụng cụ thể); kiểu thuộc tính; quyền trong thuộc tính (xác định quyền có thể đọc/ghi/không được làm gì/thông báo/...) và cuối cùng là giá trị của thuộc tính (nó có thể là bất cứ thứ gì: giá trị nhíp tim, trạng thái công tắc, một chuỗi hoặc có thể là một địa chỉ của thuộc tính khác...)
- Characteristic như một vùng chứa dữ liệu người dùng. Chúng bao gồm ít nhất 2 thuộc tính: khai báo characteristic và giá trị characteristic.
- Service như một vùng chứa các đặc điểm liên quan đến một phần nào đó. Thuộc tính mô tả service thường chỉ là khai báo service đó.
- Hầu hết các thuộc tính khai báo đều có quyền “cho đọc” và không yêu cầu bảo mật bởi vì nó không chứa thông tin nhạy cảm nào mà chỉ phục vụ quá trình BLE trung tâm “khám phá” BLE ngoại vi.
- GATT tổ chức dữ liệu dựa trên 2 vai trò: máy khách (Client) – tương ứng với BLE trung tâm và Máy chủ (Server) - tương ứng với BLE ngoại vi.

Cụ thể, mỗi máy chủ sẽ chứa dữ liệu được tổ chức dưới dạng các thuộc tính - ATT (gồm địa chỉ thuộc tính, kiểu thuộc tính, quyền thuộc tính, giá trị thuộc tính). Khi một máy khách muốn đọc hoặc viết các giá trị thuộc tính từ/đến một máy chủ, nó sẽ đưa ra yêu cầu đọc hoặc ghi vào máy chủ. Máy chủ sẽ trả lời với giá trị thuộc tính hoặc xác nhận. Trong trường hợp hoạt động đọc, tùy thuộc vào kiểu thuộc tính mà máy khách muốn đọc, máy chủ sẽ tìm trong dữ liệu của mình giá trị phù hợp. Mặt khác, trong một thao tác ghi, máy khách sẽ phải cung cấp dữ liệu phù hợp với kiểu thuộc tính muốn ghi trong máy chủ và máy chủ có thể từ chối lệnh ghi nếu không phù hợp. Ngoài ra, máy chủ cũng có thể chủ động tự gửi giá trị đến máy khách bằng cách đặt quyền trong thuộc tính là thông báo (Notification) hoặc chỉ định (Indication). Như vậy có thể thấy, máy chủ sẽ tổng hợp tất cả các dữ liệu theo dạng thuộc tính và máy khách muốn truy cập thì cần phải tuân theo quy tắc “thuộc tính” này.

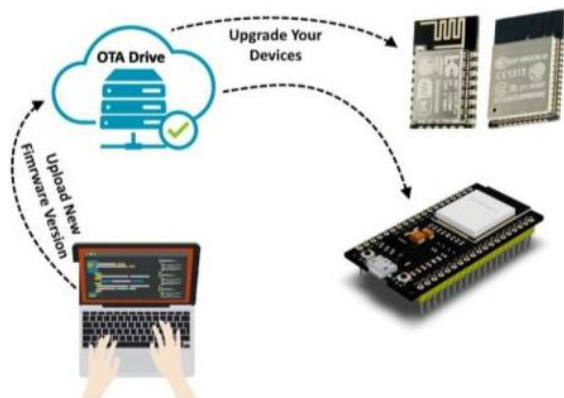
2. Tìm hiểu về cách thức hoạt động OTA

- ESP32 có thể nạp firmware thông qua wifi mà không cần kết nối với mạch nạp. Ta có thể tự tạo 1 web site để truy cập vào ESP32 và upload firmware lên. Vì vậy, cần có một máy chủ lưu trữ thông tin và quản lý version của sản phẩm.



- Nguyên lý FOTA: Để có thể nạp firmware từ xa, cần tuân theo các bước:
 - Update firmware lên máy chủ.
 - ESP32 sẽ gọi 1 API lên server đó, truy vấn xem có bản cập nhật mới không. Thời gian đó có thể lặp lại tùy thuộc vào người phát triển lập trình.
 - Nếu có thông tin thay đổi version, ESP32 sẽ download firmware về thông qua internet.
 - Cuối cùng, thiết bị được khởi động lại với firmware mới.

(Ta có thể sử dụng server của OTA Drive. Một website quản lý firmware miễn phí.)



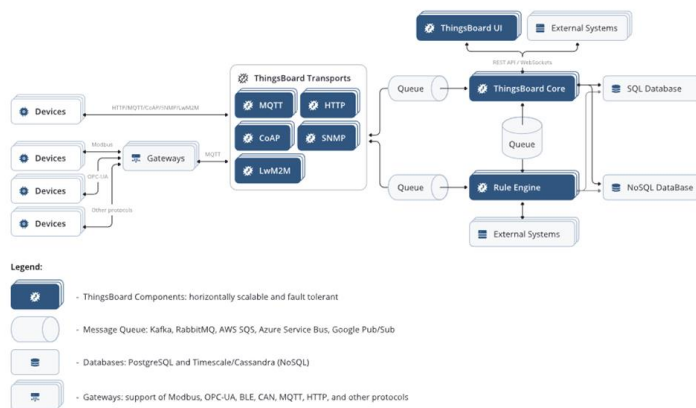
3. Quản lý bằng phần mềm (lấy thingsboard từ IoT)

- Đối với Web, đề xuất sử dụng nền tảng ThingsBoard

Thingsboard là một nền tảng IoT mã nguồn mở cho phép phát triển, quản lý và mở rộng quy mô nhanh chóng cho các dự án IoT. Nền tảng cung cấp các cổng kết nối cho thiết bị lên với các giao thức truyền thông lên mạng internet như MQTT, HTTP, CoAP. Với ThingsBoard, chúng ta có thể:

- Cung cấp cơ sở dữ liệu lưu thông tin gửi từ thiết bị lên và thông tin người dùng cấu hình vào
- Thu thập và trực quan hóa dữ liệu từ các thiết bị và tài sản.
- Phân tích phép đo từ xa đến và kích hoạt cảnh báo với xử lý sự kiện phức tạp.
- Điều khiển thiết bị của bạn bằng các cuộc gọi thủ tục từ xa (RPC).

- Xây dựng luồng công việc dựa trên sự kiện vòng đời thiết bị, sự kiện API REST, yêu cầu RPC,...
- Thiết kế bảng thông tin động và đáp ứng, đồng thời trình bày phép đo từ xa và thông tin chi tiết về thiết bị hoặc tài sản cho khách hàng của bạn.
- Bật các tính năng cụ thể trong trường hợp sử dụng bằng cách sử dụng chuỗi quy tắc có thể tùy chỉnh.
- Đẩy dữ liệu thiết bị sang các hệ thống khác.
- Ngoài ra có cơ chế hàng chờ xử lý các bản tin để tránh tắc nghẽn lưu thông bản tin trên server và làm mất gói tin. Cung cấp các khối giao diện nhỏ widget để người dùng tự xây dựng giao diện từng khối theo ngôn ngữ javascript hoặc có sẵn các widget cho người dùng sử dụng. Một điểm đặc biệt nữa là nó cung cấp phân quyền quản lý thiết bị và giao diện cho những nhóm quản lý thiết bị hay giao diện đó.



❖ ThingsBoard Transports

ThingsBoard cung cấp các API dựa trên MQTT, HTTP, CoAP và LwM2M có sẵn cho các ứng dụng/chương trình cơ sở của thiết bị của bạn. Mỗi API giao thức được cung cấp bởi một thành phần máy chủ riêng biệt và là một phần của “Lớp vận chuyển” của ThingsBoard. MQTT Transport cũng cung cấp API cổng được sử dụng bởi các cổng đại diện cho nhiều thiết bị và/hoặc cảm biến được kết nối.

Khi Transport nhận được tin nhắn từ thiết bị, nó sẽ được phân tích cú pháp và đẩy vào Hàng đợi tin nhắn lâu bền. Việc gửi tin nhắn chỉ được xác nhận tới thiết bị sau khi tin nhắn tương ứng được hàng đợi tin nhắn xác nhận.

❖ ThingsBoard Core

ThingsBoard Core chịu trách nhiệm xử lý lệnh gọi API REST và đăng ký WebSocket. Nó cũng chịu trách nhiệm lưu trữ thông tin cập nhật về các phiên hoạt động của thiết bị và theo dõi trạng thái kết nối của thiết bị. Các nút nền tảng có thể tham gia cụm, trong đó mỗi nút chịu trách nhiệm cho các phân vùng nhất định của các thông báo đến

❖ ThingsBoard Rule Engine

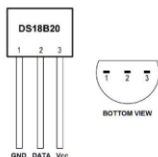
Công cụ quy tắc ThingsBoard là trung tâm của hệ thống và chịu trách nhiệm xử lý các thư đến. Công cụ quy tắc sử dụng Hệ thống diễn viên dưới mui xe để triển khai các diễn viên cho các thực thể chính: chuỗi quy tắc và nút quy tắc. Các nút của Công cụ quy tắc có thể tham gia cụm, trong đó mỗi nút chịu trách nhiệm cho các phân vùng nhất định của thư đến.

Công cụ quy tắc đăng ký nguồn cấp dữ liệu đến từ (các) hàng đợi và chỉ xác nhận thông báo sau khi nó được xử lý. Có nhiều chiến lược có sẵn để kiểm soát thứ tự hoặc quá trình xử lý thông báo và tiêu chí xác nhận thông báo. Xem chiến lược gửi và chiến lược xử lý để biết thêm chi tiết.

- Đối với App, đề xuất sử dụng ThingsBoard

4. Tìm hiểu các thiết bị phần cứng

a. Cảm biến DS18B20



- Kích thước: 28mm x 12mm x 10mm (L x W x H)
- Chip chính: 18B20 cảm biến nhiệt độ
- Điện áp làm việc: DC 3- 5. 5V
- Phạm vi đo nhiệt độ (độ C): -55 ~ 125°C
- Sai số: 0.5 độ C khi đo ở dải -10 – 85 độ C
- Độ phân giải: người dùng có thể chọn từ 9 – 12 bits
- Chuẩn giao tiếp: 1-Wire (1 dây).
- Có cảnh báo nhiệt khi vượt ngưỡng cho phép và cấp nguồn từ chân data.

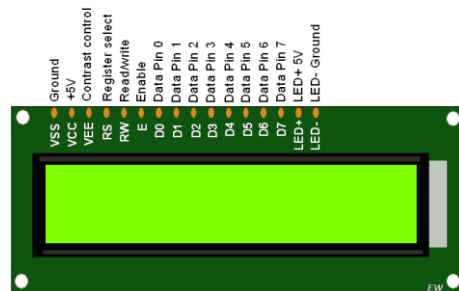
b. Vi điều khiển ESP32: tích hợp bộ nhớ, khối xử lý, khối truyền thông



Vi điều khiển	Dual-Core 32-bit (ESP-WROOM-32)
---------------	---------------------------------

Điện áp đầu vào	2.2 – 3.6V
GPIO	34
Flash Memory	4MB
SRAM	520KB
Clock Speed	240MHz
Wi-Fi	802.11b/g/n Wi-Fi transceiver
Bluetooth	Bluetooth 4.2/BLE

- c. **Sử dụng màn hình hiển thị LCD** tại mỗi node cảm biến với độ phân giải là 0.1°C



- Điện áp MAX: 7V
- Điện áp MIN: - 0,3V
- Hoạt động ổn định : 2.7-5.5V
- Điện áp ra mức cao : > 2.4
- Điện áp ra mức thấp : <0.4V
- Dòng điện cấp nguồn : 350uA - 600uA
- Nhiệt độ hoạt động : - 30 - 75 °C

Thông thường, LCD 16x2 cần cấp 5V đầu vào để hoạt động. Tuy nhiên, điện áp đầu ra của ESP32 cho 3.3V, nên ta cần hàn thêm một số linh kiện vào mặt sau của LCD
 Nguồn: <http://www.smythe-consulting.com/2018/04/converting-5v-lcd-displays-to-3v.html>

Bài viết cho thấy chúng ta cần sử dụng ICL7660 và 2 tụ 10uF

d. IC ổn áp 3.3V AMS1117

- Mỗi node cảm biến cần sử dụng nguồn từ pin Li – po 3.7V. Năng lượng từ pin sẽ cung cấp cho vi điều khiển ESP32, chân 3.3V của ESP32 sẽ cấp cho màn hình LCD và cảm biến DS18B20. Tuy nhiên, do điện áp đầu vào tối đa của ESP32 là 3.6V, không thể vượt quá, nên chúng em sẽ sử dụng một IC ổn áp để có điện áp 3.3V cấp cho ESP32



- Thông tin về AMS1117

Điện áp ngõ ra	1.5V, 1.8V, 2.5V, 2.85V, 3.3V, 5V
Dòng ra tối đa	1A
Áp ngõ vào	$1.5 < V_{in}-V_{out} < 12V$ (4.8-8.7VDC)

- Do có sự sụt áp ở trên IC, nên để được đầu ra 3.3V cung cấp cho ESP32, ta cần có đầu vào trong khoảng 4.8 – 15.3V. Do đó, chúng em đề xuất sử dụng 2 hoặc 3 viên pin Li – po 3.7V để có mức điện áp này

e. **Tính toán công suất tiêu thụ**

Link kiện	Điện áp sử dụng (V)	Dòng điện cực đại (mA)	Công suất cực đại (mW)
ESP32	3.3	260	858
LCD 16x2	3.3	20	66
DS18B20	3.3	1.5	4.95
Tổng		281.5	928.95

Nhận xét: Năng lượng được lấy ra từ IC ổn áp AMS1117 là 3.3V với dòng cực đại là 1A. Dựa vào bảng trên, ta thấy dòng điện tiêu thụ của thiết bị là khoảng 300mA.

X. Thử nghiệm và Đánh giá

XI. Hoàn thiện sản phẩm

XII. Kết luận

(Đối chiếu lại kế hoạch dự kiến ở mục Tổ chức thực hiện, Nếu có thay đổi thì cần ghi rõ những phần thay đổi so với Kế hoạch, nêu lý do)

Nguyễn Văn A

Nội dung	Mô tả (tính năng, thông số,...)	Kết quả cần đạt	Thời gian thực hiện thực tế (theo tuần)	Trạng thái (hoàn thành, chưa hoàn thành, không thực hiện) và kết quả đã đạt được so với dự kiến

Nguyễn Văn B

Nội dung	Mô tả (tính năng, thông số,...)	Kết quả cần đạt	Thời gian thực hiện thực tế (theo tuần)	Trạng thái (hoàn thành, chưa hoàn thành, không thực hiện) và kết quả đã đạt được so với dự kiến

Nguyễn Văn C

Nội dung	Mô tả (tính năng, thông số,...)	Kết quả cần đạt	Thời gian thực hiện thực tế (theo tuần)	Trạng thái (hoàn thành, chưa hoàn thành, không thực hiện) và kết quả đã đạt được so với dự kiến