

This image shows a full page of primary-ruled paper. It features multiple horizontal rows of small black dots, evenly spaced vertically across the entire page. There are no margins, text, or other markings present.

[illegible]

[illegible]

[illegible]

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ và tên	Mã số SV	Lớp	Khoa
1	Trần Thế Đầu	106150160	15DT3	Điện tử - Viễn thông
2	Lê Hữu Phú	106150194	15DT3	Điện tử - Viễn thông
3	Nguyễn Ngọc Quang	106150128	15DT2	Điện tử - Viễn thông

- Tên đề tài đồ án:* **Hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT.**
- Đề tài thuộc diện:* ☐ Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện
- Các số liệu và dữ liệu ban đầu:*
 - Thông tin về các ngôn ngữ lập trình.
 - Thông tin về các giao thức.
 - Thông tin về các linh kiện điện tử, các module chức năng.
- Nội dung các phần thuyết minh và tính toán:*
 - Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Trần Thế Đầu	<ul style="list-style-type: none">- Tìm hiểu và lên ý tưởng cho đề tài.- Thiết kế mô hình.
2	Lê Hữu Phú	
3	Nguyễn Ngọc Quang	

- Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Trần Thế Đầu	<ul style="list-style-type: none">- Tìm hiểu và thiết kế server và web.
2	Lê Hữu Phú	<ul style="list-style-type: none">- Tìm hiểu và thiết kế khối vi xử lý trung tâm, các kết nối ngoại vi (phần cứng và phần mềm).- Thi công mô hình.
3	Nguyễn Ngọc Quang	<ul style="list-style-type: none">- Tìm hiểu và thiết kế ứng dụng Android.

5. *Họ tên người hướng dẫn:* ThS. Hồ Viết Việt.

6. *Ngày giao nhiệm vụ đồ án:* 06/01/2020.

7. *Ngày hoàn thành đồ án:* 06/06/2020.

Đà Nẵng, ngày ... tháng ... năm 2020.

Trưởng Bộ môn

Người hướng dẫn

ThS. Hồ Viết Việt

LỜI NÓI ĐẦU

Chúng em xin chân thành cảm ơn thầy Hồ Viết Việt và các thầy cô trong Khoa Điện tử - Viễn thông, trường Đại học Bách Khoa - Đại học Đà Nẵng đã truyền đạt những kiến thức quý báu trong những năm học vừa qua và nhất là tạo điều kiện thuận lợi cho chúng em học tập thực hiện đồ án tốt nghiệp này.

Để có được kết quả như ngày hôm nay, chúng em rất biết ơn gia đình bạn bè đã động viên, khích lệ, tạo mọi điều kiện thuận lợi nhất về vật chất lẫn tinh thần trong suốt quá trình học tập cũng như quá trình thực hiện đồ án tốt nghiệp.

Mặc dù đã cố gắng hoàn thành đồ án trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm, góp ý và tận tình chỉ bảo của quý thầy cô và các bạn.

CAM ĐOAN

Kính gửi: Hội đồng bảo vệ Đồ án tốt nghiệp khoa Điện tử - Viễn thông, trường Đại học Bách Khoa – Đại học Đà Nẵng.

Chúng em: TRẦN THẾ ĐẦU

 LÊ HỮU PHÚ

 NGUYỄN NGỌC QUANG

Sinh viên chuyên ngành Kỹ thuật máy tính, khoa Điện tử - Viễn thông, trường Đại học Bách Khoa – Đại học Đà Nẵng.

Chúng em xin cam đoan nội dung của đồ án này không phải là bản sao chép nào của bất cứ Đồ án hoặc công trình đã có từ trước. Nếu vi phạm, chúng em xin chịu mọi hình thức kỷ luật của Khoa hoặc Nhà trường.

Đà Nẵng, ngày tháng năm 2020

Trần Thế Đầu

Lê Hữu Phú

Nguyễn Ngọc Quang

MỤC LỤC

NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN

NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

LỜI NÓI ĐẦU.....i

CAM ĐOAN..... ii

MỤC LỤC iii

DANH SÁCH CÁC BẢNG, HÌNH VẼ.....v

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT vii

MỞ ĐẦU.....1

CHƯƠNG 1: TỔNG QUAN HỆ THỐNG NHẮC NHỞ UỐNG THUỐC DỰA TRÊN NỀN TẢNG IOT.....2

1.1 Giới thiệu chương2

1.2 Giải pháp2

1.2.1 Sơ đồ hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT.....2

1.2.2 Kiến trúc chức năng và vận hành hệ thống.....2

1.2.3 Chức năng và lựa chọn linh kiện chính cho từng khối4

1.3 Kết luận chương5

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ TÌM HIỂU CÁC THÀNH PHẦN.....6

2.1 Giới thiệu chương6

2.2 Giới thiệu về Internet of Things6

2.3 Giao thức7

2.3.1 Giao thức HTTP.....7

2.3.2 Giao thức MQTT10

2.4 Các ngôn ngữ sử dụng.....13

2.4.1 Ngôn ngữ Java13

2.4.2 Ngôn ngữ Javascript15

2.4.3 Ngôn ngữ Python16

2.4.4 Ngôn ngữ PHP18

2.4.5 Ngôn ngữ HTML19

2.5 Giới thiệu về Server20

2.6 Kết luận chương20

CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ PHẦN CỨNG, PHẦN MỀM	21
3.1 Giới thiệu chương	21
3.2 Thiết kế phần cứng.....	21
3.2.1 Khối vi điều khiển trung tâm.....	21
3.2.2 Khối điều khiển	22
3.2.3 Khối còi báo hiệu.....	23
3.2.4 Khối hiển thị	23
3.3 Thiết kế phần mềm.....	24
3.3.1 Phần mềm cho vi điều khiển trung tâm	24
3.3.2 Phần mềm cho server.....	31
3.3.3 Phần mềm ứng dụng Android.....	32
3.4 Thiết kế Server và Website	34
3.4.1 Server.....	34
3.4.2 Website	35
3.5 Kết luận chương.....	35
CHƯƠNG 4: THI CÔNG, KIỂM TRA KẾT QUẢ VÀ ĐÁNH GIÁ HỆ THỐNG	36
4.1 Giới thiệu chương	36
4.2 Thi công mạch.....	36
4.2.1 Về mạch trên phần mềm.....	36
4.2.2 Mạch thực tế	38
4.3 Thi công website và tạo cơ sở dữ liệu.....	38
4.3.1 XAMPP	38
4.3.2 Cơ sở dữ liệu.....	39
4.3.3 Trang chủ web	40
4.4 Xây dựng ứng dụng Android	41
4.5 Kết quả thực nghiệm và đánh giá hệ thống.....	43
4.5.1 Kết quả thực nghiệm.....	43
4.5.2 Đánh giá hệ thống.....	46
4.6 Kết luận chương.....	46
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	47
TÀI LIỆU THAM KHẢO	
PHỤ LỤC	

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Hình 1.1 Mô hình tổng quan hệ thống nhắc nhở uống thuốc	2
Hình 1.2 Giải đồ usecase của hệ thống.....	3
Hình 1.3 Giải đồ activity của hệ thống.....	4
Hình 2.1 Mô hình tổng quan về hệ thống IoT	6
Hình 2.2 Ví dụ một phiên làm việc HTTP	7
Hình 2.3 Ví dụ về gói tin HTTP request	8
Hình 2.4 Ví dụ gói tin HTTP response.....	9
Hình 2.5 Mô hình giao thức MQTT	10
Hình 2.6 Giao thức MQTT nằm ở lớp ứng dụng trong bộ giao thức TCP/IP.....	11
Hình 2.7 Mô hình publish - subscribe	12
Hình 2.8 Biểu tượng của Mosquitto trên trang chủ https://mosquitto.org/	12
Hình 2.9 Màn hình khi khởi chạy Mosquitto trên máy chủ	12
Hình 2.10 Biểu tượng của Java	13
Hình 2.11 Biểu tượng của Javascript.....	15
Hình 2.12 Biểu tượng của Python	16
Hình 2.13 Python đứng thứ nhất trong ngành khoa học dữ liệu (số liệu năm 2018)	17
Hình 2.14 Biểu tượng của PHP	18
Hình 2.15 Biểu tượng chính thức của phiên bản HTML mới nhất, HTML5	19
Hình 2.16 Cấu trúc cơ bản của một trang web sử dụng HTML	19
Hình 3.1 Sơ đồ nguyên lý khối VDK trung tâm.....	21
Hình 3.2 Sơ đồ nguyên lý khối điều khiển.....	22
Hình 3.3 Sơ đồ nguyên lý khối còi báo hiệu	23
Hình 3.4 Sơ đồ nguyên lý khối hiển thị.....	23
Hình 3.5 Sơ đồ thuật toán chương trình chính	24
Hình 3.6 Sơ đồ thuật toán chương trình khởi tạo hệ thống	25
Hình 3.7 Sơ đồ thuật toán chương trình hiển thị độ ẩm, nhiệt độ	26
Hình 3.8 Sơ đồ thuật toán chương trình cài đặt giờ	27
Hình 3.9 Sơ đồ thuật toán chương trình cài đặt phút.....	28
Hình 3.10 Sơ đồ thuật toán chương trình nhắc nhở	29
Hình 3.11 Sơ đồ thuật toán chương trình tìm hộp thuốc	30
Hình 3.12 Giải đồ activity của server	31

Hình 3.13 Sơ đồ thuật toán ứng dụng trên điện thoại (1).....	32
Hình 3.14 Sơ đồ thuật toán ứng dụng trên điện thoại (2).....	33
Bảng 3.1 Các topic đăng ký và nhiệm vụ tương ứng.....	34
Bảng 3.2 Bảng danh sách nhắc nhở	35
Bảng 3.3 Bảng độ ẩm	35
Bảng 3.4 Bảng nhiệt độ	35
Hình 4.1 Sơ đồ nguyên lý của mạch	36
Hình 4.2 Sơ đồ đi dây trên mạch in.....	37
Hình 4.3 Hình ảnh 3D của mạch.....	37
Hình 4.4 Board mạch xử lý trung tâm sau khi hoàn thiện	38
Hình 4.5 Giao diện phần mềm XAMPP.....	38
Hình 4.6 Giao diện hệ quản trị cơ sở dữ liệu	39
Hình 4.7 Danh sách các bảng trong cơ sở dữ liệu.....	39
Hình 4.8 Cấu trúc bảng độ ẩm.....	39
Hình 4.9 Cấu trúc bảng nhiệt độ	40
Hình 4.10 Cấu trúc bảng thời gian	40
Hình 4.11 Giao diện chính	40
Hình 4.12 Hiển thị thông tin về độ ẩm và nhiệt độ	41
Hình 4.13 Giao diện quản lý chung.....	41
Hình 4.14 Giao diện khi muốn thêm một báo thức mới	42
Hình 4.15 Giao diện khi đến giờ uống thuốc	43
Hình 4.16 Mô hình sản phẩm mặt trước và sau	43
Hình 4.17 Giao diện mặc định	44
Hình 4.18 Giao diện cài đặt giờ	44
Hình 4.19 Giao diện cài đặt phút.....	45
Hình 4.20 Giao diện nhắc nhở.....	45
Hình 4.21 Giao diện tìm hộp thuốc	46
Hình PL.1 Kit RF WiFi + Bluetooth ESP32	1
Hình PL.2 Màn hình LCD 1602	1
Hình PL.3 Module I2C	2
Hình PL.4 Module DHT11	2
Hình PL.5 Còi chip 5V	3
Hình PL.6 Pin Lithium Battery 3.7V, 2200mAh	3

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Ký hiệu/ chữ viết tắt	Tên đầy đủ	Ghi chú
GPIO	General Purpose Input/Output	Cổng vào/ra vạn năng
HTTP	HyperText Transfer Protocol	Giao thức truyền siêu văn bản
HTML	HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản
XHTML	Extensible HyperText Markup Language	Ngôn ngữ đánh dấu siêu văn bản mở rộng
TCP/IP	Transmission Control Protocol/ Internet Protocol	Giao thức điều khiển truyền nhận/Giao thức liên mạng
SQL	Structured Query Language	Ngôn ngữ truy vấn dữ liệu
LCD	Liquid-Crystal Display	Màn hình tinh thể lỏng
WiFi	Wireless Fidelity	Mạng không dây
App	Application	Ứng dụng
PCB	Printed Circuit Board	Bảng mạch in
LED	Light Emitting Diode	Diode phát quang
DC	Direct Current	Dòng điện một chiều
I/O	Input/Output	Vào/ra
BJT	Bipolar Junction Transistor	Transistor lưỡng cực
IC	Integrated Circuit	Mạch tích hợp
OOP	Object Oriented Programming	Lập trình hướng đối tượng
GUI	Graphical User Interface	Giao diện đồ họa người dùng
API	Application Programming Interface	Giao diện lập trình ứng dụng
CSS	Cascading Style Sheets	Các tập tin định kiểu theo tầng
AI	Artificial Intelligence	Trí tuệ nhân tạo
AWT	Abstract Window Toolkit	Bộ công cụ cửa sổ trừu tượng
MQTT	Message Queuing Telemetry Transport	
VĐK	Vi điều khiển	

MỞ ĐẦU

Mục đích thực hiện đề tài là xây dựng một hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT bao gồm các sản phẩm: mô hình phần cứng, website và ứng dụng điện thoại.

Mục tiêu đề tài là phục vụ cho người sử dụng một cách thuận tiện nhất, nơi mà người dùng có thể thiết lập các nhắc nhở trên bất cứ thiết bị nào có trong tay mà không cần quan tâm đến công việc đồng bộ dữ liệu.

Phạm vi của đề tài là khu vực nhà ở hoặc bệnh viện có thể kết nối được WiFi. Đối tượng hướng tới là người sử dụng cá nhân có nhu cầu được nhắc nhở uống thuốc.

Cấu trúc của đề án này gồm có bốn chương:

Chương 1: Tổng quan hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT.

Chương 2: Cơ sở lý thuyết và tìm hiểu các thành phần.

Chương 3: Phân tích thiết kế phần cứng, phần mềm.

Chương 4: Thi công, kiểm tra kết quả và đánh giá hệ thống.

Nhóm thực hiện đề án xin chân thành cảm ơn ThS. Hồ Viết Việt đã tận tình hướng dẫn, hỗ trợ về kiến thức và hướng hoàn thiện đề án trong suốt quá trình nghiên cứu thực hiện đề án.

Nhóm xin chân thành cảm ơn!

CHƯƠNG 1: TỔNG QUAN HỆ THỐNG NHẮC NHỞ UỐNG THUỐC DỰA TRÊN NỀN TẢNG IOT

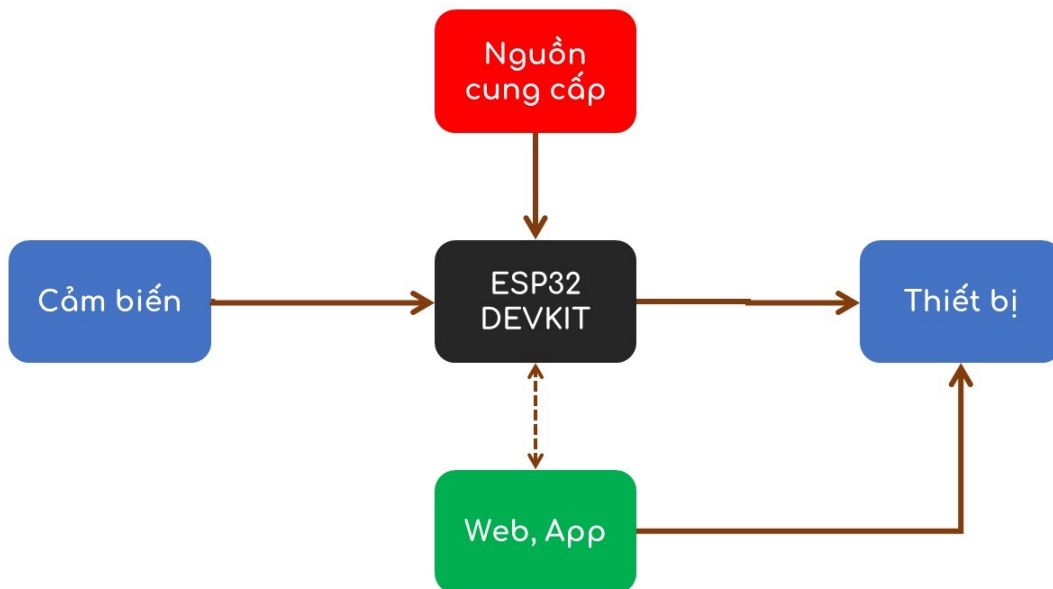
1.1 Giới thiệu chương

Xuất phát từ thực tế hiện nay trên thị trường chỉ là những sản phẩm hộp thuốc đơn giản, không có thông báo tự động hay nhắc nhở người dùng uống thuốc. Do đó, chúng em đã nảy sinh ra ý tưởng tích hợp nền tảng IoT vào hộp thuốc, nhằm đưa ra một sản phẩm mới có khả năng tự động nhắc nhở, thông báo cho người dùng uống thuốc đúng giờ. Sản phẩm này mang lại tính tiện dụng và thân thiện với người dùng.

Chương này chúng em sẽ trình bày tổng quan về hệ thống **“Nhắc nhở uống thuốc dựa trên nền tảng IoT”**, sơ đồ tổng quát của hệ thống, các giản đồ usecase và activity, cũng như chức năng và linh kiện cho từng khối của hệ thống nêu trên.

1.2 Giải pháp

1.2.1 Sơ đồ hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT



Hình 1.1 Mô hình tổng quan hệ thống nhắc nhở uống thuốc

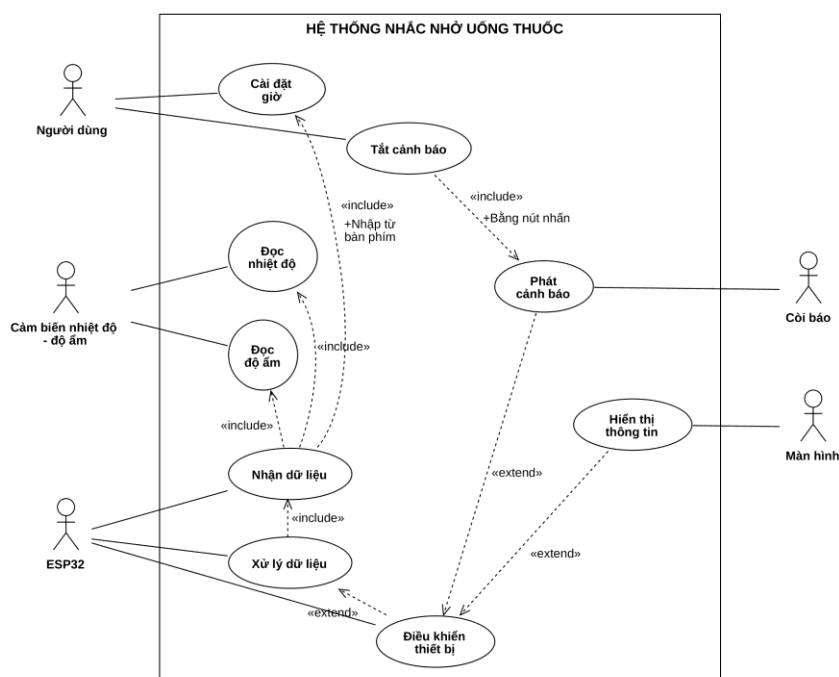
1.2.2 Kiến trúc chức năng và vận hành hệ thống

❖ Giản đồ Use Case

Hình 1.2 giới thiệu về kiến trúc chức năng và cách vận hành của **“Hệ thống nhắc nhở uống thuốc”**. Nhằm bắt được các yêu cầu từ hệ thống, mô tả được sự liên quan của

người dùng từ bên ngoài tác động như thế nào vào hệ thống, chỉ ra các đối tượng của hệ thống và các sự kiện diễn ra trong quá trình hoạt động.

Use Case::Medication Reminder System



Hình 1.2 Giảm đồ usecase của hệ thống

Khi hệ thống bắt đầu hoạt động:

Cảm biến nhiệt độ - độ ẩm: Đọc thông tin nhiệt độ, độ ẩm của môi trường bằng cảm biến chuyên dụng. Nhằm mục đích giúp người dùng theo dõi nhiệt độ, độ ẩm để có cách thức bảo quản thuốc phù hợp.

Người dùng: Thực hiện thao tác cài đặt giờ nhắc nhở uống thuốc thông qua các phím nhân chức năng. Cũng như tắt cảnh báo khi cần.

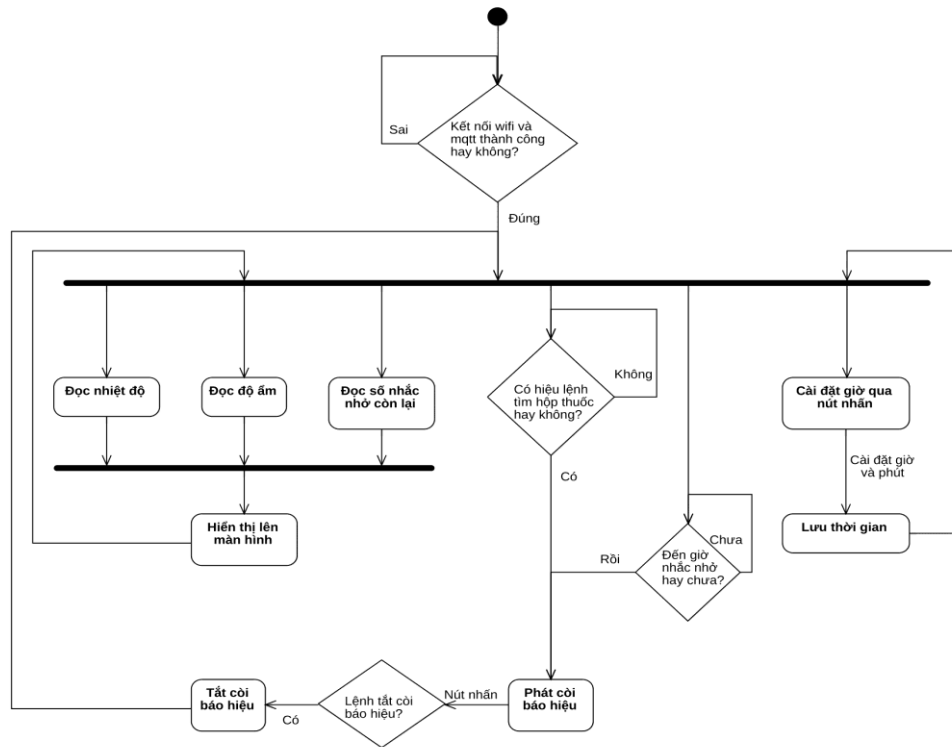
ESP32: Có nhiệm vụ nhận dữ liệu từ cảm biến và người dùng để tiến hành xử lý dữ liệu nhận được, từ đó điều khiển các thiết bị.

Màn hình: Hiển thị thông tin về nhiệt độ, độ ẩm, số lần nhắc nhở còn lại.

Còi báo: Phát hoặc tắt cảnh báo nếu có hiệu lệnh điều khiển.

❖ Giải đồ Activity

Activity1::Medication Reminder System



Hình 1.3 Giải đồ activity của hệ thống

Giải thích giải đồ: Khi hệ thống khởi động, VDK sẽ kiểm tra đã kết nối mạng WiFi và MQTT hay chưa. Nếu đã kết nối thành công thì sẽ tiến đến các tiến trình đọc thông số về nhiệt độ, độ ẩm và số nhắc nhở còn lại sau đó hiển thị lên màn hình. Bên cạnh đó, người dùng có thể cài đặt giờ để nhắc nhở uống thuốc theo giờ được cài, hệ thống sau khi nhận được thông tin về thời gian sẽ lưu lại nội dung. Ngoài ra, nếu có hiệu lệnh tìm hộp thuốc từ app hoặc đã đến giờ nhắc nhở (được cài đặt trước đó) thì hộp thuốc sẽ phát ra còi cảnh báo đồng thời hiển thị nội dung trên màn hình để người dùng nhận biết, người dùng có thể tắt còi báo thông qua nút nhấn.

1.2.3 Chức năng và lựa chọn linh kiện chính cho từng khối

❖ Khối vi điều khiển trung tâm:

Chức năng: VDK trung tâm để kết nối các thiết bị và xử lý dữ liệu. Đồng thời truyền nhận dữ liệu giữa VDK trung tâm - Server thông qua WiFi và giao thức MQTT.

Linh kiện: Kit RF WiFi + Bluetooth ESP32.

❖ Khối Server database:

Chức năng: Đóng vai trò là máy chủ server xử lý các dữ liệu từ khối VDK trung tâm gửi lên.

Linh kiện: Laptop Dell.

❖ **Khối còi báo hiệu:**

Chức năng: Báo hiệu khi có lệnh tìm hộp thuốc hoặc đã đến giờ nhắc nhở người dùng uống thuốc.

Linh kiện: Còi chip 5V.

❖ **Khối nút nhấn:**

Chức năng: Thiết lập thời gian, tắt cảnh báo. Bao gồm có 4 nút nhấn.

Linh kiện: Nút nhấn 4 chân cao.

❖ **Khối cảm biến:**

Chức năng: Thu thập thông tin của môi trường để gửi về khối VĐK trung tâm.

Linh kiện: Module cảm biến nhiệt độ - độ ẩm DHT11.

❖ **Khối hiển thị:**

Chức năng: Hiển thị các thông tin cần thiết lên màn hình.

Linh kiện: LCD 1602 và module giao tiếp I2C.

❖ **Khối nguồn:**

Chức năng: Nguồn 7.4VDC cung cấp cho VĐK trung tâm, khối hiển thị.

Linh kiện: Pin sạc 18650 2200mAh.

1.3 Kết luận chương

Trên đây chúng em đã trình bày một các tổng quan về hệ thống, giải pháp cho vấn đề đã nêu ra. Ở chương sau chúng em sẽ đi vào chi tiết của hệ thống, các thiết bị được sử dụng.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ TÌM HIỂU CÁC THÀNH PHẦN

2.1 Giới thiệu chương

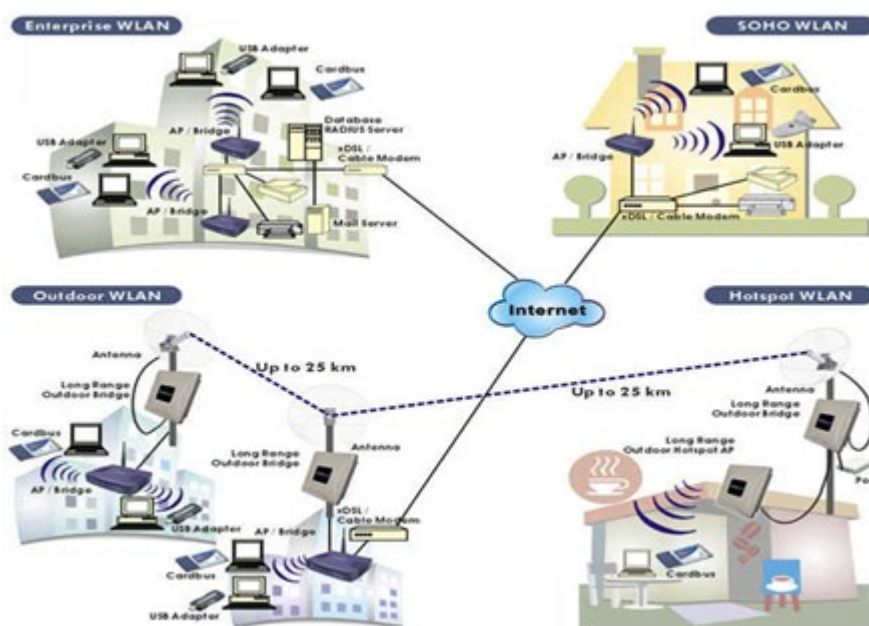
Sau khi trình bày tổng quan hệ thống, chương này chúng em sẽ giới thiệu về nền tảng IoT, về server cũng như những thành phần về phần mềm mà hệ thống sử dụng: các giao thức, ngôn ngữ. Các thành phần về phần cứng chính là các linh kiện (xem Phụ lục giới thiệu về linh kiện).

2.2 Giới thiệu về Internet of Things

❖ Giới thiệu về Internet of Things:

Internet of Things (IoT) hay “Internet của vạn vật” là khi tất cả mọi thứ đều được kết nối với nhau qua mạng Internet, người dùng (chủ) có thể kiểm soát mọi đồ vật của mình qua mạng mà chỉ bằng một thiết bị thông minh, chẳng hạn như điện thoại thông minh, máy tính bảng, máy tính cá nhân.

Điểm quan trọng của IoT đó là các đối tượng phải có thể được nhận biết và định dạng (identifiable) cũng như chỉ sự tồn tại của chúng trong một kiến trúc mạng tính kết nối. Nếu mọi đối tượng, kể cả con người, được “đánh dấu” để phân biệt bản thân đối tượng đó với những thứ xung quanh thì chúng ta có thể hoàn toàn quản lý được nó thông qua máy tính. Việc đánh dấu (tagging) có thể được thực hiện thông qua nhiều công nghệ, chẳng hạn như RFID (Radio Frequency Identification - Nhận dạng qua tần số vô tuyến), NFC (Near-Field Communications - Kết nối trường gần), mã vạch, mã QR (Quick Response), watermark kỹ thuật số,... Việc kết nối thì có thể thực hiện qua Wi-Fi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại, RF, ...



Hình 2.1 Mô hình tổng quan về hệ thống IoT

❖ **Kiến trúc và tính chất của Internet of Things:**

- Kiến trúc dựa trên sự kiện: Các thực thể, máy móc trong IoT sẽ phản hồi dựa theo các sự kiện diễn ra trong lúc chúng hoạt động theo thời gian thực. Một số nhà nghiên cứu từng nói rằng một mạng lưới các cảm biến (sensor) chính là một thành phần đơn giản của IoT.
- Là một hệ thống phức tạp: Trong một thế giới mở, IoT sẽ mang tính chất phức tạp bởi nó bao gồm một lượng lớn các đường liên kết giữa những thiết bị, máy móc, dịch vụ với nhau, ngoài ra còn bởi khả năng thêm vào các nhân tố mới.

2.3 Giao thức

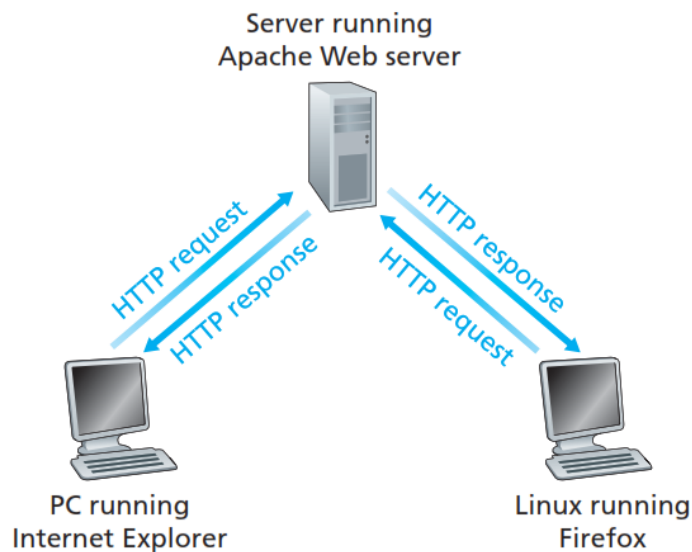
2.3.1 Giao thức HTTP

❖ **Giới thiệu về HTTP:**

HTTP (HyperText Transfer Protocol) là một giao thức nằm ở lớp ứng dụng (Application layer) của tập giao thức TCP/IP, sử dụng để truyền nhận dữ liệu giữa các hệ thống phân tán thông qua Internet, cụ thể giao thức hoạt động theo mô hình Khách hàng – Máy chủ (Client - Server) bằng cách thực hiện các quá trình Yêu cầu – Phản hồi (Request – Response) giữa các hệ thống máy tính khác nhau. Giao thức HTTP quy định cấu trúc của các gói tin và cách thức truyền nhận dữ liệu giữa client và server thông qua môi trường Internet. Với khả năng truyền dẫn siêu văn bản (text, hình ảnh, âm thanh, video,...), HTTP hiện là nền tảng truyền dẫn dữ liệu của ứng dụng duyệt web hiện nay và được ứng dụng rất nhiều trong các hệ thống Internet of Things (IoT).

❖ **Mô hình giao thức:**

Mỗi khi người dùng sử dụng trình duyệt và truy cập vào một website, một phiên làm việc HTTP (gọi là Session) sẽ được diễn ra với client là máy tính của người dùng và server là máy chủ của website. Mặc định HTTP sẽ được thực hiện thông qua cổng (port) 80, đây là port chuẩn của giao thức được định nghĩa bởi tổ chức IANA (Internet Assigned Numbers Authority - Tổ chức cấp phát số hiệu Internet) quy định.



Hình 2.2 Ví dụ một phiên làm việc HTTP

Ví dụ:

Khi người dùng truy cập vào địa chỉ:

“http://localhost/Server/templates/index.html”.

Quá trình của một phiên làm việc HTTP diễn ra như sau:

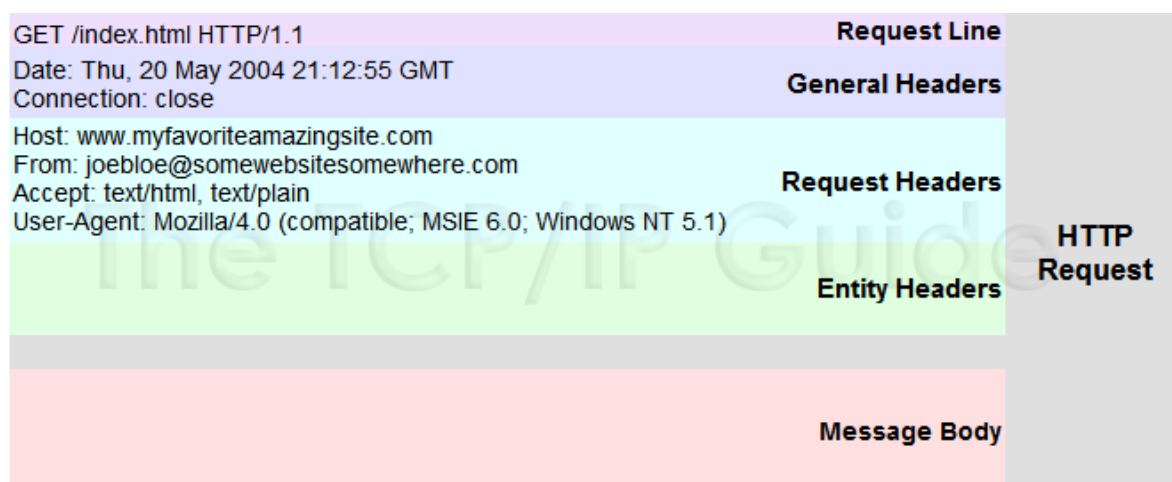
- HTTP client thiết lập một kết nối TCP (Transmission Control Protocol - Giao thức điều khiển truyền vận) đến server. Nếu thiết lập thành công, client và server sẽ truyền nhận dữ liệu với nhau thông qua kết nối này, kết nối được thiết lập còn gọi là socket interface bao gồm các thông tin: địa chỉ IP, loại giao thức giao vận (chính là TCP), và port (mặc định là 80).
- Sau khi kết nối thành công, client gửi một HTTP request đến server thông qua socket interface vừa được thiết lập. Trong gói tin request sẽ chứa đường dẫn yêu cầu (path name) là **“/Server/templates/”** và tệp tin **“index.html”** nằm trong path name.
- Server sẽ nhận và xử lý request từ client thông qua socket, sau đó đóng gói dữ liệu tương ứng và gửi một HTTP response về cho client. Dữ liệu trả về sẽ là một file HTML chứa các loại dữ liệu khác nhau như văn bản, hình ảnh,...
- Server đóng kết nối TCP.
- Client nhận được dữ liệu phản hồi từ server và đóng kết nối TCP.

❖ **Định dạng của một gói tin:**

Trong suốt phiên làm việc HTTP như ở trên, có hai loại gói tin chính được trao đổi là gói tin yêu cầu gửi từ client đến server gọi là **“Request message”** và gói tin phản hồi gửi từ server đến client gọi là **“Response message”**. Hai loại gói tin này được giao thức HTTP định nghĩa rất cụ thể:

HTTP request message:

Gồm 3 phần chính là: Request line, Header và Body. Dữ liệu được chia thành các dòng và định dạng kết thúc dòng là <CR><LF> (tương ứng với các ký tự 0x0A, 0x0D trong bảng mã ASCII). Riêng dòng cuối cùng của gói tin sẽ được báo hiệu bằng 2 lần định dạng kết thúc: <CR><LF><CR><LF>.



Hình 2.3 Ví dụ về gói tin HTTP request

- **Request line** là dòng đầu tiên của gói HTTP Request, bao gồm 3 trường: phương thức (method), đường dẫn (path – có nhiều bài viết gọi là URL hoặc URI cho trường này) và phiên bản giao thức (HTTP version).

Phương thức (method) có thể là: GET, POST, HEAD, PUT và DELETE. Hai phương thức phổ biến nhất là GET và POST, trong ví dụ trên là phương thức GET thường dùng để yêu cầu tài nguyên cung cấp trong trường URL.

Đường dẫn (path) dùng để định danh nguồn tài nguyên mà client yêu cầu, bắt buộc phải có ít nhất là dấu “/”.

Phiên bản giao thức (HTTP version): là phiên bản HTTP client đang sử dụng (thường là HTTP/1.0 hoặc HTTP/1.1)

- Tiếp theo là các dòng **Header**, các dòng này là không bắt buộc, viết ở định dạng “Name:Value” cho phép client gửi thêm các thông tin bổ sung về thông điệp HTTP request và thông tin về chính client. Một số header thông dụng như:

Accept: loại nội dung có thể nhận được từ thông điệp response. Ví dụ: text/plain, text/html

Accept-Encoding: các kiểu nén được chấp nhận. Ví dụ: gzip, deflate, xz, exi...

Connection: tùy chọn điều khiển cho kết nối hiện thời. Ví dụ: Keep-Alive, Close...

Cookie: thông tin HTTP Cookie từ server

- Cuối cùng là phần **Body**, là dữ liệu gửi từ client đến server trong gói tin HTTP request. Đa số các gói tin gửi theo phương thức GET sẽ có Body trống, các phương thức như POST hay PUT thường dùng để gửi dữ liệu nên sẽ có bao gồm dữ liệu trong trường Body.

HTTP response message:

HTTP/1.1 200 OK	Status Line	HTTP Response
Date: Thu, 20 May 2004 21:12:58 GMT	General Headers	
Connection: close		
Server: Apache/1.3.27	Response Headers	
Accept-Ranges: bytes		
Content-Type: text/html	Entity Headers	
Content-Length: 170		
Last-Modified: Tue, 18 May 2004 10:14:49 GMT		
<html>	Message Body	
<head>		
<title>Welcome to the Amazing Site!</title>		
</head>		
<body>		
<p>This site is under construction. Please come back later. Sorry!</p>		
</body>		
</html>		

Hình 2.4 Ví dụ gói tin HTTP response

- Dòng **Status line** gồm 3 trường là phiên bản giao thức (HTTP version), mã trạng thái (Status code) và mô tả trạng thái (Status text):

Phiên bản giao thức (HTTP version): phiên bản của giao thức HTTP mà server hỗ trợ, thường là HTTP/1.0 hoặc HTTP/1.1.

Mã trạng thái (Status code): mô tả trạng thái kết nối dưới dạng số, mỗi trạng thái sẽ được biểu thị bởi một số nguyên. Ví dụ: 200, 404, 302,...

Mô tả trạng thái (Status text): mô tả trạng thái kết nối dưới dạng văn bản một cách ngắn gọn, giúp người dùng dễ hiểu hơn so với mã trạng thái. Ví dụ: 200 OK, 404 Not Found,...

- Các dòng **Header line** của gói tin response có chức năng tương tự như gói tin request, giúp server có thể truyền thêm các thông tin bổ sung đến client dưới dạng các cặp “Name:Value”.
- Phần **Body** là nơi đóng gói dữ liệu để trả về cho client, thông thường trong duyệt web thì dữ liệu trả về sẽ ở dưới dạng một trang HTML để trình duyệt có thể thông dịch được và hiển thị ra cho người dùng.

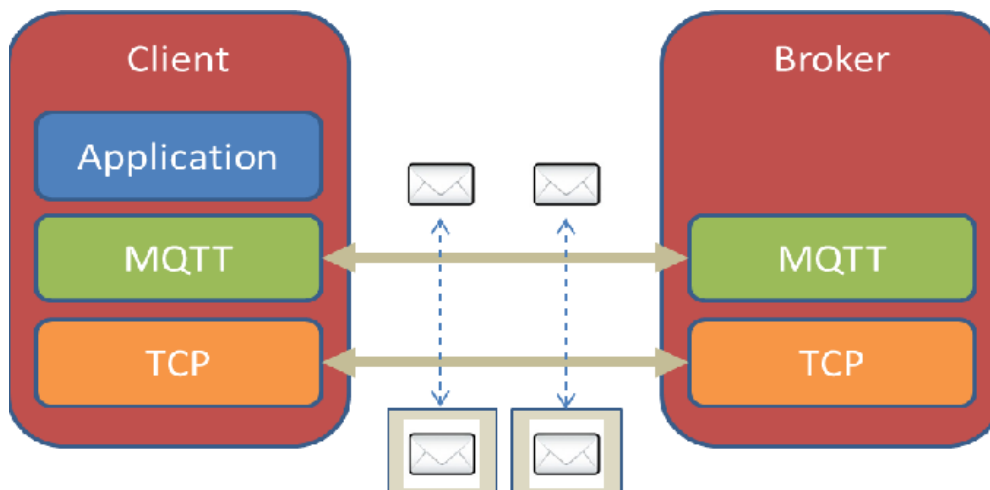
❖ **HTTP trong hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT:**

Trong hệ thống “*Nhắc nhở uống thuốc dựa trên nền tảng IoT*”, HTTP là giao thức được server sử dụng, trong đó server sử dụng phương thức GET để gửi dữ liệu khi người dùng chọn thời gian nhắc nhở và nhấn lưu (save); hoặc khi lấy vị trí của hàng trong bảng danh sách nhắc nhở để xóa (delete).

2.3.2 Giao thức MQTT

❖ **Giới thiệu về MQTT:**

MQTT là từ viết tắt của Message Queue Telemetry Transport. Đây là một giao thức truyền thông điệp (message) theo mô hình publish/subscribe (xuất bản – theo dõi), sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định.

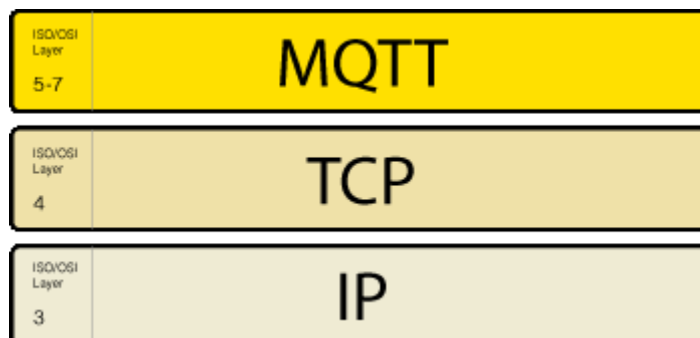


Hình 2.5 Mô hình giao thức MQTT

❖ **Đặc điểm:**

- Giao thức vận chuyển các message đơn giản.
- Các message trong MQTT không đồng bộ.
- Gửi nhận các message theo publish/Subscribe.

- Có 3 cấp độ QoS (Qualities of service – Chất lượng dịch vụ) là 0, 1, 2.
- Giao thức MQTT chạy trên nền tảng TCP/IP.



Hình 2.6 Giao thức MQTT nằm ở lớp ứng dụng trong bộ giao thức TCP/IP

Giao thức MQTT có các thuật ngữ chính cơ bản là: publisher, subscriber, broker topic, qos, retain, session, last will and testament (lwt).

❖ **Ưu điểm:**

Giao thức MQTT là một trong những giao thức phổ biến nhất hiện nay trong các ứng dụng và nền tảng IoT qua môi trường Internet. Mô hình của giao thức này giúp đáp ứng những yêu cầu quan trọng và đặc biệt phù hợp trong việc truyền nhận các gói tin IoT giữa các node (thiết bị, cảm biến,...) mà giao thức HTTP không thể có được, điển hình như:

- MQTT là giao thức 2 chiều, các node không chỉ truyền mà còn có thể dễ dàng đăng ký nhận tin một cách thụ động theo cơ chế publish-subscribe (với HTTP thì node không thể nhận tin thụ động được mà luôn luôn phải khởi tạo kết nối trước với server).
- MQTT hỗ trợ cơ chế 1-n, nghĩa là một node có khả năng gửi tin đến rất nhiều node khác chỉ trong một lần gửi hoặc ngược lại (HTTP chỉ hỗ trợ 1-1 giữa client và server theo cơ chế request-response).
- MQTT là một “lightweight protocol”, với ưu điểm gói tin nhẹ, không chứa nhiều dữ liệu dư thừa nên có thể truyền nhận dữ liệu một cách mượt mà trong điều kiện bị giới hạn về băng thông đường truyền.
- MQTT hỗ trợ cấu hình QoS còn HTTP thì không.

❖ Mô hình publish - subscribe:

Mô hình publish-subscribe là một mô hình được sử dụng thay thế cho mô hình khách – chủ truyền thống. Trong mô hình giao thức MQTT, các client có thể trao đổi các gói tin với nhau bằng cách xuất bản gói tin (gọi là publisher) hoặc đăng ký nhận gói tin (gọi là subscriber), tuy nhiên các client không hề biết đến sự tồn tại của nhau. Có một thành phần thứ ba, đóng vai trò như một server trung tâm (được gọi là broker) là nơi mà các client có thể biết đến và đóng vai trò trung gian có khả năng lọc các gói tin nhận từ các publisher để gửi đến các subscriber phù hợp.



Hình 2.7 Mô hình publish - subscribe

❖ MQTT broker:

Đối tác của các MQTT client là MQTT broker, đây chính là trái tim của bất kỳ một giao thức tổ chức theo mô hình publish-subscribe. Tùy thuộc vào các công việc cụ thể, một broker có thể xử lý hàng ngàn kết nối đồng thời từ các client. MQTT broker chịu trách nhiệm chính trong việc nhận tất cả các gói tin, lọc gói tin và quyết định gửi gói tin đến các client đã đăng ký nhận các gói tin đó.

❖ MQTT broker trong hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT:

Trong hệ thống “*Nhắc nhở uống thuốc dựa trên nền tảng IoT*”, chúng em sử dụng phần mềm Mosquitto làm MQTT broker và là trung gian truyền nhận các gói tin giữa các MQTT client (ở đây là giữa VDK, website và app).



Hình 2.8 Biểu tượng của Mosquitto trên trang chủ <https://mosquitto.org/>

Mosquitto là một MQTT Broker mã nguồn mở được viết bởi Roger Light trong tổ chức Eclipse giúp triển khai các dự án sử dụng giao thức MQTT. Mosquitto là phần mềm đa nền tảng, có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux, Mac OS X, Unix,...

Trong hệ thống, Mosquitto được cài đặt trên máy chủ, sử dụng port 1883 và luôn hoạt động 24/24.

```
C:\WINDOWS\system32\cmd.exe - "C:\Program Files\mosquitto\mosquitto" -v
Microsoft Windows [Version 10.0.18363.836]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin>"C:\Program Files\mosquitto\mosquitto" -v
1591901045: mosquitto version 1.6.9 starting
1591901045: Using default config.
1591901045: Opening ipv6 listen socket on port 1883.
1591901045: Opening ipv4 listen socket on port 1883.
```

Hình 2.9 Màn hình khi khởi chạy Mosquitto trên máy chủ

2.4 Các ngôn ngữ sử dụng

2.4.1 Ngôn ngữ Java

❖ Giới thiệu về Java:



Hình 2.10 Biểu tượng của Java

Java là ngôn ngữ lập trình bậc cao, được phát triển bởi Sun Microsystems, do James Gosling khởi xướng và phát hành vào năm 1995 như là một thành phần cốt lõi của nền tảng Java của Sun Microsystems (Java 1.0 [J2SE]). Java chạy trên rất nhiều nền tảng khác nhau, như Windows, Mac và các phiên bản khác nhau của UNIX.

Java được hiểu là một loại ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp. Không giống với những ngôn ngữ lập trình thông thường, thay vì việc biên dịch mã nguồn trở thành mã nguồn máy hoặc trực tiếp thông dịch mã nguồn khi chạy thì Java được thiết kế để biên dịch mã nguồn thành bytecode (là cách thức lưu trữ dạng mã các chỉ thị trong lập trình máy tính, được thiết kế để phần mềm thông dịch thực hiện hiệu quả trên nền tảng máy ảo). Sau đó, bytecode sẽ được môi trường thực thi (runtime environment).

❖ Đặc điểm:

- **Hướng đối tượng:** Trong Java, mọi thứ đều là Object. Java có thể mở rộng vì nó dựa trên mô hình Object.
- **Nền tảng độc lập:** Không giống như nhiều ngôn ngữ lập trình khác (C, C++), khi Java được biên dịch, nó không biên dịch sang một máy tính cụ thể trên nền tảng nào, thay vào đó là những bytecode độc lập với nền tảng. Byte code này được phân phối trên web và được thông dịch bằng Virtual Machine (JVM) trên bất cứ nền tảng nào mà nó đang chạy.
- **Đơn giản:** Java được thiết kế để dễ học. Nếu bạn hiểu cơ bản về khái niệm lập trình hướng đối tượng Java, thì có thể nắm bắt ngôn ngữ này rất nhanh.
- **Bảo mật:** Với tính năng an toàn của Java, nó cho phép phát triển những hệ thống không có virus, giả mạo. Các kỹ thuật xác thực dựa trên mã hóa công khai.
- **Kiến trúc trung lập:** Trình biên dịch của Java tạo ra một định dạng file object có kiến trúc trung lập, làm cho code sau khi biên dịch có thể chạy trên nhiều bộ vi xử lý, với sự hiện diện của Java runtime system.

- **Portable:** Là kiến trúc trung lập và không phụ thuộc vào việc thực hiện là những đặc điểm chính nhất khi nói về khía cạnh Portable của Java. Trình biên dịch trong Java được viết bằng ANSI C với một ranh giới portable gọn gàng, đó là một tập hợp con của POSIX (Portable Operating System Interface - giao diện hệ điều hành di động). Bạn có thể mang bytecode của Java lên bất cứ nền tảng nào.
- **Mạnh mẽ:** Java nỗ lực loại trừ những tình huống dễ bị lỗi bằng cách nhấn mạnh chủ yếu là kiểm tra lỗi thời gian biên dịch và kiểm tra runtime.
- **Đa luồng:** Với tính năng đa luồng của Java, bạn có thể viết các chương trình có thể thực hiện nhiều tác vụ đồng thời. Tính năng này cho phép các nhà phát triển xây dựng các ứng dụng tương tác có thể chạy trơn tru.
- **Thông dịch:** Bytecode của Java được dịch trực tiếp tới các nền tảng gốc và nó không được lưu trữ ở bất cứ đâu.
- **Hiệu suất cao:** Với việc sử dụng trình biên dịch Just-In-Time, Java cho phép thực thi với hiệu suất cao, nhanh chóng phát hiện, gỡ lỗi.
- **Phân tán:** Java được thiết kế cho môi trường phân tán của Internet.
- **Linh động:** Java được coi là năng động hơn C hay C++ vì nó được thiết kế để thích nghi với môi trường đang phát triển. Các chương trình Java có thể mang theo một lượng lớn thông tin run-time, được sử dụng để xác minh và giải quyết các truy cập đến đối tượng trong thời gian chạy.

❖ **Ứng dụng:**

Ngôn ngữ Java được lựa chọn để tạo ra các website, ứng dụng di động, phần mềm tùy chỉnh, cổng thông tin điện tử,... và được coi như một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới hiện nay. Nhiều nhà phát triển phần mềm khởi đầu với Java và đi theo nó qua rất nhiều dự án cho đến tận bây giờ. Java là một chương trình mặc định trong các hệ điều hành và vai trò của nó đối với chúng ta là vô cùng to lớn. Dưới đây là một số ứng dụng chính mà có thể được tạo bởi sử dụng ngôn ngữ lập trình Java:

- **Ứng dụng độc lập (Standalone App):** Thường được biết đến với tên gọi khác là Desktop App hoặc Windows-based App. Một ứng dụng mà chúng ta cần cài đặt trên mỗi thiết bị như media player, antivirus, ... AWT (bộ công cụ cửa sổ trừu tượng) và Swing (bộ công cụ tiện ích GUI cho Java) được sử dụng trong Java để tạo các Standalone App.
- **Ứng dụng Web (Web App):** Một ứng dụng mà chạy trên máy chủ (server-side) và tạo trang động (dynamic page), được gọi là Web App. Hiện tại, các công nghệ Servlet, JSP (JavaServer Pages), Struts, JSF (JavaServer Faces),... được sử dụng để tạo Web App trong Java.
- **Ứng dụng doanh nghiệp (Enterprise App):** Có lợi thế là tính bảo mật cao, cân bằng tải (load balancing) và phân nhóm dữ liệu (clustering). Trong java, EJB (Enterprise JavaBeans) được sử dụng để tạo các Enterprise App.

- **Ứng dụng di động (Mobile App):** Đây là loại ứng dụng được tạo cho thiết bị mobile. Hiện tại thì Android và Java ME (Java platform, Micro Edition) được sử dụng để tạo loại ứng dụng này.
- ❖ **Java trong hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT:**

Trong hệ thống “*Nhắc nhở uống thuốc dựa trên nền tảng IoT*”, Java là ngôn ngữ được sử dụng để viết ứng dụng trên điện thoại.

2.4.2 Ngôn ngữ Javascript

- ❖ **Giới thiệu về Javascript:**



Hình 2.11 Biểu tượng của Javascript

Javascript (viết tắt là “js”) là một ngôn ngữ lập trình mang đầy đủ tính năng của một ngôn ngữ lập trình động mà khi nó được áp dụng vào một tài liệu HTML, nó có thể đem lại khả năng tương tác động trên các trang web. Cha đẻ của ngôn ngữ này là Brendan Eich, đồng sáng lập dự án Mozilla, quỹ Mozilla, và tập đoàn Mozilla.

Javascript thật sự rất linh hoạt. Bạn có thể bắt đầu với các bước nhỏ, với ? viện ảnh, bố cục có tính thay đổi và phản hồi đến các nút nhấn. Khi có nhiều kinh nghiệm hơn, bạn có thể tạo ra các trò chơi, hoạt họa 2 chiều hoặc 3 chiều, ứng dụng cơ sở dữ liệu toàn diện và nhiều thứ khác.

Bản thân Javascript là một ngôn ngữ linh động. Các nhà phát triển đã viết ra một số lượng lớn các công cụ thuộc top của core Javascript, mở ra một lượng lớn tính năng bổ sung với ít nỗ lực nhất. Nó bao gồm:

- Giao diện lập trình ứng dụng trên trình duyệt (API) - Các API được xây dựng bên trong các trình duyệt web, cung cấp tính năng như tạo HTML động, thu thập và điều khiển video trực tiếp từ webcam của người dùng hoặc sinh ra đồ họa 3D và các mẫu âm thanh.
 - Các API bên thứ ba cho phép nhà phát triển kết hợp tính năng trong website của họ từ người cung cấp nội dung khác chẳng hạn như Twitter hay Facebook.
 - Từ các khung phần mềm (framework) và thư viện bên thứ ba bạn có thể áp dụng tới tài liệu HTML của bạn, cho phép bạn nhanh chóng xây dựng được các trang web và các ứng dụng.
- ❖ **Ứng dụng:**
 - Ứng dụng trong lập trình website: Khi nhắc đến lập trình web người ta không thể không nhắc đến bộ 3 HTML, CSS và Javascript. Có thể nói không phải là

tất cả, song hầu như các website đang chạy hiện nay đều sử dụng Javascript và các Front-end framework của nó như: Bootstrap, jQuery Foundation, UIKit,... Ở đó Javascript giúp tạo các hiệu ứng hiển thị trên website, các tương tác với người dùng.

- Xây dựng các ứng dụng web cho máy chủ: Đây là một xu hướng công nghệ có thể nói là rất hot hiện nay (từ 2016 đến giờ). Các lập trình viên khá hào hứng với các framework từ Javascript như: Node.js, AngularJS,... Cụ thể những cái này sẽ hỗ trợ tạo ra các ứng dụng web thiên về tương tác thời gian thực của người dùng. Nếu cùng cấu hình máy chủ tương tự thì điều đó là không thể đối với PHP, Java, Python, .Net khi số lượng người dùng (user) tương tác cùng lúc quá nhiều. Máy chủ sẽ không thể nào gánh nổi, nhưng với các framework của Javascript thì mọi chuyện sẽ hoàn toàn khác.
- Xây dựng các ứng dụng di động, trò chơi và ứng dụng trên desktop.

❖ Javascript trong hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT:

Trong hệ thống “**Nhắc nhở uống thuốc dựa trên nền tảng IoT**”, Javascript là ngôn ngữ được sử dụng với mục đích tải lại một phần trang web mà không phải tải lại toàn bộ trang (dùng cho bảng nhắc nhở).

2.4.3 Ngôn ngữ Python

❖ Giới thiệu về Python:



Hình 2.12 Biểu tượng của Python

Python được sáng tạo bởi Guido Van Rossum vào những năm cuối thập niên 80, đầu thập niên 90 tại Viện nghiên cứu Quốc gia về Toán học và Khoa học máy tính ở Hà Lan.

Python là một ngôn ngữ bậc cao, thông dịch, ngôn ngữ kịch bản tương tác và hướng đối tượng. Python được thiết kế để lập trình viên có thể đọc hiểu dễ dàng nhất. Python thường sử dụng các từ khóa tiếng anh trong khi các ngôn ngữ khác thường sử dụng các dấu câu. Cấu trúc cú pháp của nó cũng dễ dàng hơn so với các ngôn ngữ khác. Python rất dễ học bởi vì tài liệu liên quan có thể tìm thấy ở bất cứ đâu.

❖ Ứng dụng:

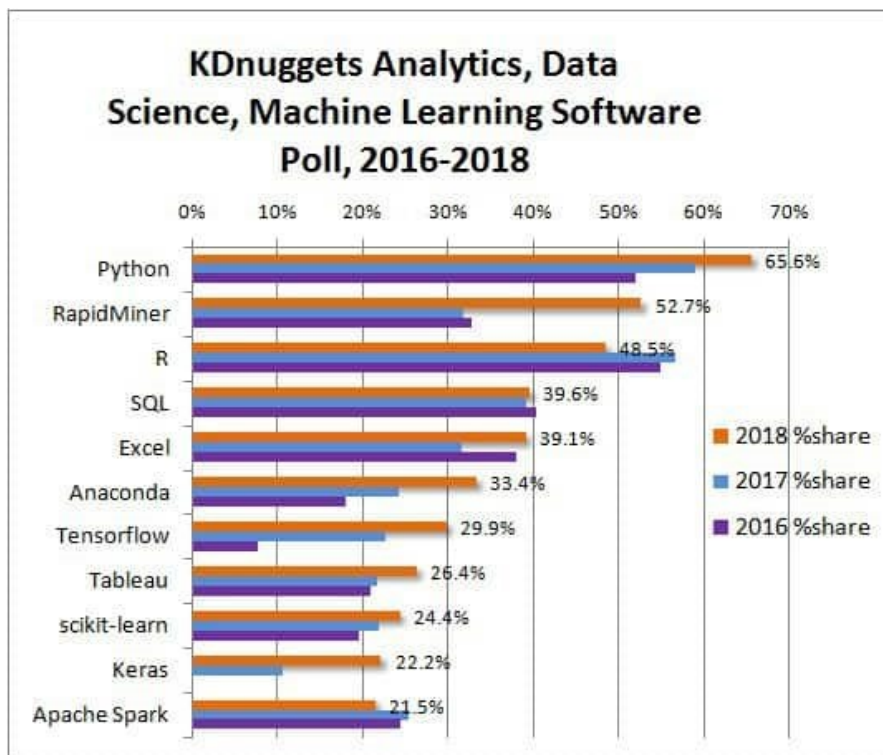
- Python để viết ngôn ngữ lập trình kịch bản (scripting language).

Nhiều người đã ứng dụng Python để viết ra một đoạn kịch bản (script) mà bạn có thể đưa vào bất cứ một video nào và chương trình sẽ cho ra hàng loạt các bản ghi (transcript) đã được chuyển ngữ ngẫu nhiên. Những bản này cũng không mấy hoàn hảo nguyên nhân là do cách dùng từ và ngữ điệu của người Châu Á chúng ta, tuy nhiên ít nhất thì nó cũng được miễn phí.

Các ứng dụng web. Một số ứng dụng được xây dựng dựa vào ngôn ngữ lập trình Python như Dropbox, Instagram, Netflix, Youtube, Spotify.

- Ngành khoa học dữ liệu (data science).

Python đang là ứng dụng đang đóng vai trò một loại ngôn ngữ lập trình phổ biến nhất trong ngành khoa học dữ liệu và nó đang dần nuốt chửng thị phần của các ngôn ngữ khác.



Hình 2.13 Python đứng thứ nhất trong ngành khoa học dữ liệu (số liệu năm 2018)

Có rất nhiều thư viện Python mà bạn có thể dùng cho các bộ dữ liệu lớn. Một số thư viện đáng chú ý như NumPy (cho những thứ thuộc về toán học), SciPy (thư viện tin học kỹ thuật cao), Pandas (dành cho phân tích dữ liệu) và Matplotlib (dành cho mô hình hóa dữ liệu – data visualization).

- Machine Learning và Trí thông minh nhân tạo (AI).

Rất nhiều thư viện Python có thể và đang được sử dụng cho lĩnh vực Machine Learning, Deep Learning và AI. Có thể kể tới: Tensorflow, Theano và PyTorch. Càng nhiều lập trình viên làm việc trong lĩnh vực này thì số lượng nguồn (resources) và thư viện (libraries) lại càng tăng lên.

- Lĩnh vực IoT – Internet of Things.

Bạn không cần phải trả một khoản tiền quá lớn hay mua từ cửa hàng nào cho việc ứng dụng Python vào Internet of Things. Ngày nay, người ta chỉ cần đầu tư một cái máy tính Raspberry Pi để khởi động những dự án DIY IoT của riêng mình.

- Lập trình game.

Nhờ vào Python ứng dụng vào nhiều ngành nghề nên Python được nhiều công ty, trường học sử dụng để dạy lập trình cho trẻ em và những người mới lần đầu học lập trình. Bên cạnh những tính năng và khả năng tuyệt vời thì cú pháp đơn giản và dễ sử dụng của nó là lý do chính cho việc này.

❖ **Python trong hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT:**

Trong hệ thống “**Nhắc nhở uống thuốc dựa trên nền tảng IoT**”, Python là ngôn ngữ được sử dụng để viết chương trình khi mô hình phần cứng nhận được báo thức, khi mô hình phần cứng (một MQTT client) gửi dữ liệu thời gian, nhiệt độ, độ ẩm lên server (một MQTT client khác) thông qua broker (Mosquitto).

2.4.4 Ngôn ngữ PHP

❖ **Giới thiệu về PHP:**



Hình 2.14 Biểu tượng của PHP

PHP – viết tắt hồi quy của cụm từ “Hypertext Preprocessor”, là ngôn ngữ lập trình kịch bản (scripting language) mã nguồn mở miễn phí. PHP được dùng để tạo ra các ứng dụng web chạy ở phía máy chủ (server). Hiện nay, PHP là một trong những ngôn ngữ lập trình được sử dụng phổ biến trên thế giới.

Facebook, Yahoo, Wikipedia,... là những trang web nổi tiếng có sử dụng ngôn ngữ lập trình PHP.

❖ **Ứng dụng:**

- Tạo trang web động.
- Tạo, mở, đọc, viết, xóa và đóng tệp trên máy chủ.
- Thu thập mẫu dữ liệu.
- Gửi và nhận cookies.
- Thao tác (thêm, xóa, sửa) với dữ liệu trong database của bạn.
- Quản lý và điều khiển truy cập của người dùng.
- Mã hóa dữ liệu.

❖ **PHP trong hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT:**

Trong hệ thống “**Nhắc nhở uống thuốc dựa trên nền tảng IoT**”, PHP là ngôn ngữ được sử dụng để xử lý sự kiện từ người dùng trên website.

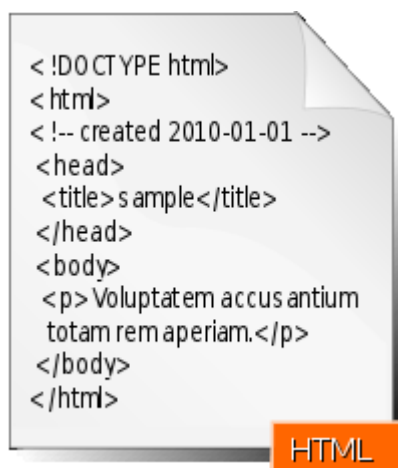
2.4.5 Ngôn ngữ HTML

❖ Giới thiệu về HTML:



Hình 2.15 Biểu tượng chính thức của phiên bản HTML mới nhất, HTML5

HTML (viết tắt của Hypertext Markup Language hay ngôn ngữ đánh dấu siêu văn bản) là ngôn ngữ đánh dấu được dùng để tạo các trang siêu văn bản được sử dụng trên mạng Internet (hay trang web). Gọi là ngôn ngữ đánh dấu là vì HTML sử dụng các thẻ để định nghĩa (hay đánh dấu) các thành phần khác nhau trên trang web.



Hình 2.16 Cấu trúc cơ bản của một trang web sử dụng HTML

Nội dung của một trang web được tạo ra sử dụng các thẻ HTML khác nhau như thẻ `<html>`, thẻ `<body>`...

Siêu văn bản (hay hypertext) là loại văn bản được sử dụng để hiển thị trên các máy vi tính (hoặc các thiết bị điện tử số). Siêu văn bản khác với văn bản thông thường ở chỗ chúng được liên kết lại với nhau và có thể được tham chiếu tới nhờ sử dụng các địa chỉ liên kết.

Để thấy được nội dung của văn bản HTML, chúng ta cần một trình duyệt web (Cốc Cốc, Chim Lạc, Google Chrome, Microsoft Edge, Safari, Opera, Brave, ...). Mục đích của trình duyệt là đọc văn bản HTML và hiển thị chúng. Trình duyệt không hiển thị thẻ HTML nhưng dùng chúng để xác định phải hiển thị văn bản ra sao.

❖ **Các phiên bản của HTML:**

Phiên bản	Năm
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

❖ **HTML trong hệ thống nhắc nhở uống thuốc dựa trên nền tảng IoT:**

Trong hệ thống “*Nhắc nhở uống thuốc dựa trên nền tảng IoT*”, HTML là ngôn ngữ được sử dụng viết website của hệ thống.

2.5 Giới thiệu về Server

Server hay còn gọi là máy chủ là một hệ thống (phần mềm và phần cứng máy tính phù hợp) đáp ứng yêu cầu trên một mạng máy tính để cung cấp, hoặc hỗ trợ cung cấp một dịch vụ mạng. Các máy chủ thường cung cấp các dịch vụ thiết yếu qua mạng, hoặc là để người dùng cá nhân trong một tổ chức lớn hoặc cho người dùng nào thông qua Internet. Các máy chủ máy tính điển hình là máy chủ cơ sở dữ liệu (database server), máy chủ tập tin (file server), máy chủ mail (mail server), máy chủ in (print server), máy chủ web (web server), máy chủ game (game server), máy chủ ứng dụng (application server), hoặc một số loại khác của máy chủ.

Máy chủ (server) trong đề tài có nhiệm vụ xử lý các gói tin hay truy cập được gửi từ máy khách (client) thông qua giao thức MQTT hoặc HTTP. Các truy cập HTTP này được gửi từ các chương trình duyệt web trên máy tính hoặc ứng dụng điện thoại hệ điều hành android.

Ngoài ra, máy chủ còn thực thi chương trình xử lý tính toán thời gian do người dùng cài đặt được lưu trong cơ sở dữ liệu, sử dụng ngôn ngữ Python. Python là một ngôn ngữ lập trình bậc cao, đa mục đích, cú pháp được đơn giản hóa, dễ đọc hiểu. Hiện nay, Python là một trong những ngôn ngữ lập trình mới và phổ biến.

2.6 Kết luận chương

Chương này chúng em đã trình bày chi tiết về các giao thức và ngôn ngữ sử dụng trong hệ thống. Chương tiếp theo chúng em sẽ trình bày về các thiết kế phần cứng và phần mềm, thiết kế server và website.

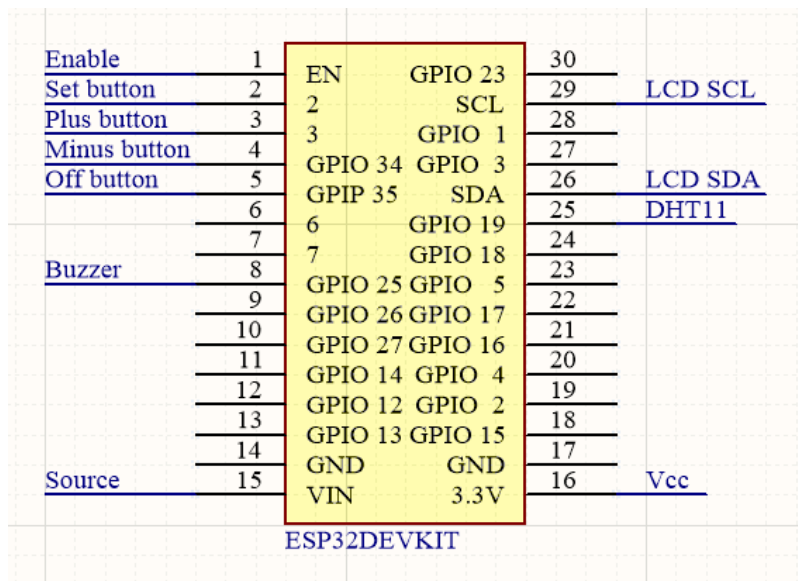
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ PHẦN CỨNG, PHẦN MỀM

3.1 Giới thiệu chương

Chương này chúng em sẽ giới thiệu về phần cứng, phần mềm của hệ thống, server và website của hệ thống.

3.2 Thiết kế phần cứng

3.2.1 Khối vi điều khiển trung tâm



Hình 3.1 Sơ đồ nguyên lý khối VDK trung tâm

Nguồn cấp vào ở chân Vin cho Kit ESP32 là 7 - 12V, ở đây chúng em dùng nguồn 7.4VDC lấy từ 2 viên pin sạc loại 18650.

Chân EN của VDK sử dụng để nạp chương trình tự động vào VDK.

Chân số 2 (GPIO36), 3 (GPIO39), 4 (GPIO34), 5 (GPIO35) lần lượt kết nối với các nút nhấn Set, Plus, Minus, Off để sử dụng cho khối điều khiển.

Chân số 8 (GPIO25) sử dụng kết nối tới khối còi báo hiệu.

Chân số 16 có mức điện áp đầu ra 3.3V, được sử dụng làm nguồn chung, cung cấp điện áp cho các khối điều khiển, khối còi báo hiệu, khối cảm biến hoạt động.

Chân số 26 (SDA) và 29 (SCL) kết nối tới khối hiển thị để truyền tải thông tin cần hiển thị lên màn hình.

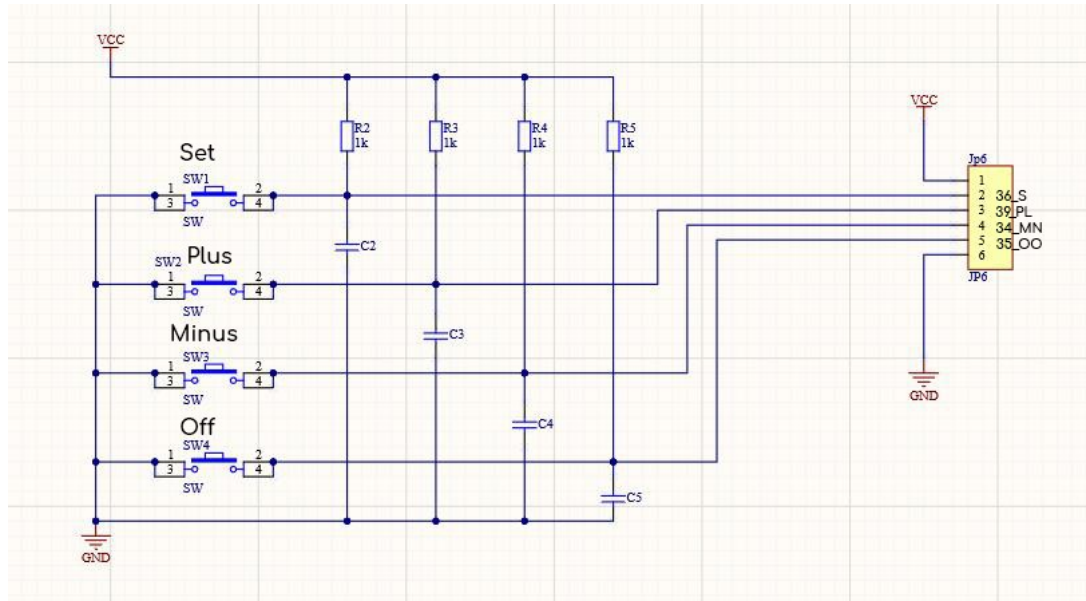
❖ **Những lưu ý:**

GPIO34, 35, 36, 39 chỉ dùng được chức năng INPUT, không dùng được chức năng OUTPUT.

GPIO1, 3 được kết nối đến bộ UART0 và ESP32 cũng nạp chương trình thông qua bộ UART này, nên tránh sử dụng 2 chân này cho chức năng I/O.

GPIO0, 2, 4, 5, 12, 15 đã được định sẵn mức logic bên trong module ESP32 để phục vụ quá trình nạp chương trình, nên tránh sử dụng các chân GPIO này.

3.2.2 Khối điều khiển



Hình 3.2 Sơ đồ nguyên lý khối điều khiển

Có 4 nút nhấn, mỗi nút nhấn kết nối tới 4 chân của VĐK. Đồng thời, mỗi nút nhấn đều có trở kéo lên nguồn và tụ để chống nhiễu cho nút nhấn.

Nút nhấn 1: Nút Set, có chức năng đi vào giao diện cài đặt thời gian, xác nhận thời gian, lưu thời gian được cài đặt.

Nút nhấn 2: Nút Plus, có chức năng tăng thời gian lên 1 đơn vị. Nếu là giờ thì tăng lên 1 giờ, nếu là phút thì tăng lên 1 phút. Trường hợp 23 giờ thì khi nhấn nút Plus sẽ về lại 0 giờ hoặc 59 phút thì sẽ về 0 phút.

Nút nhấn 3: Nút Minus, tương tự như nút Plus nhưng là trừ đi 1 đơn vị. Trường hợp 0 giờ thì khi nhấn nút Minus sẽ thành 23 giờ hoặc 0 phút thì sẽ thành 59 phút.

Nút nhấn 4: Nút Off, có chức năng tắt các cảnh báo khi đang có cảnh báo.

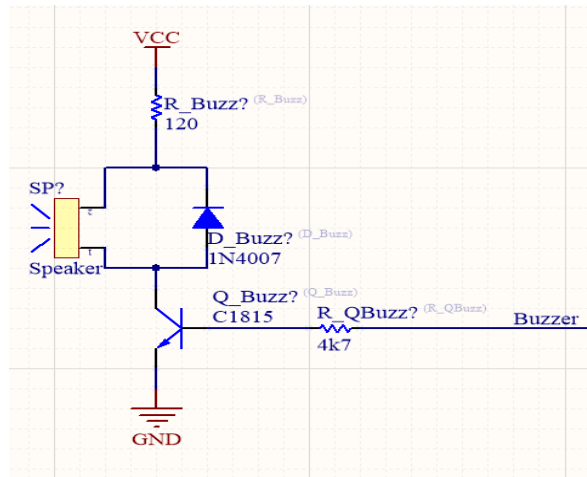
3.2.3 Khối còi báo hiệu

VĐK trung tâm dùng chân Buzzer để điều khiển tắt/mở BJT Q_Buzz.

Q_Buzz hoạt động ở chế độ bão hòa.

Nếu chân Buzzer ở mức cao (3.3V), Q_Buzz được mở, có dòng điện đổ từ nguồn Vcc (3.3V) qua loa, qua Q_Buzz về GND. Khi đó, loa phát âm thanh.

Nếu chân Buzzer ở mức thấp (0V), BJT Q_Buzz bị tắt, không có dòng chạy qua loa. Khi đó loa tắt, không có âm thanh.



Hình 3.3 Sơ đồ nguyên lý khối còi báo hiệu

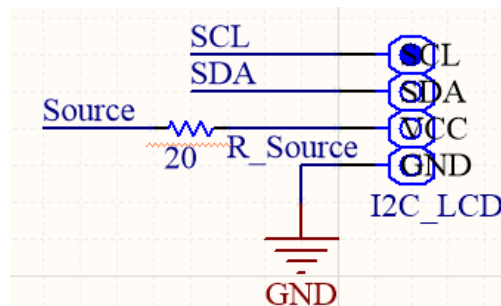
Diode D_Buzz được mắc song song với loa để tránh dòng điện cảm ứng do loa tạo ra lúc hoạt động chạy ngược về BJT làm hỏng BJT.

3.2.4 Khối hiển thị

Kết nối màn hình LCD với VĐK thông qua module giao tiếp I2C để tiết kiệm chân VĐK. Chỉ cần sử dụng 2 chân tín hiệu là SDA và SCL.

Chân SDA và SCL trên module I2C kết nối tương ứng đến chân SDA và SCL ở khối VĐK trung tâm. Hai chân này có chức năng nhận dữ liệu từ VĐK để giải mã và hiển thị lên màn hình LCD.

Chân Vcc ở module được nối vào khối nguồn qua một điện trở nhỏ.

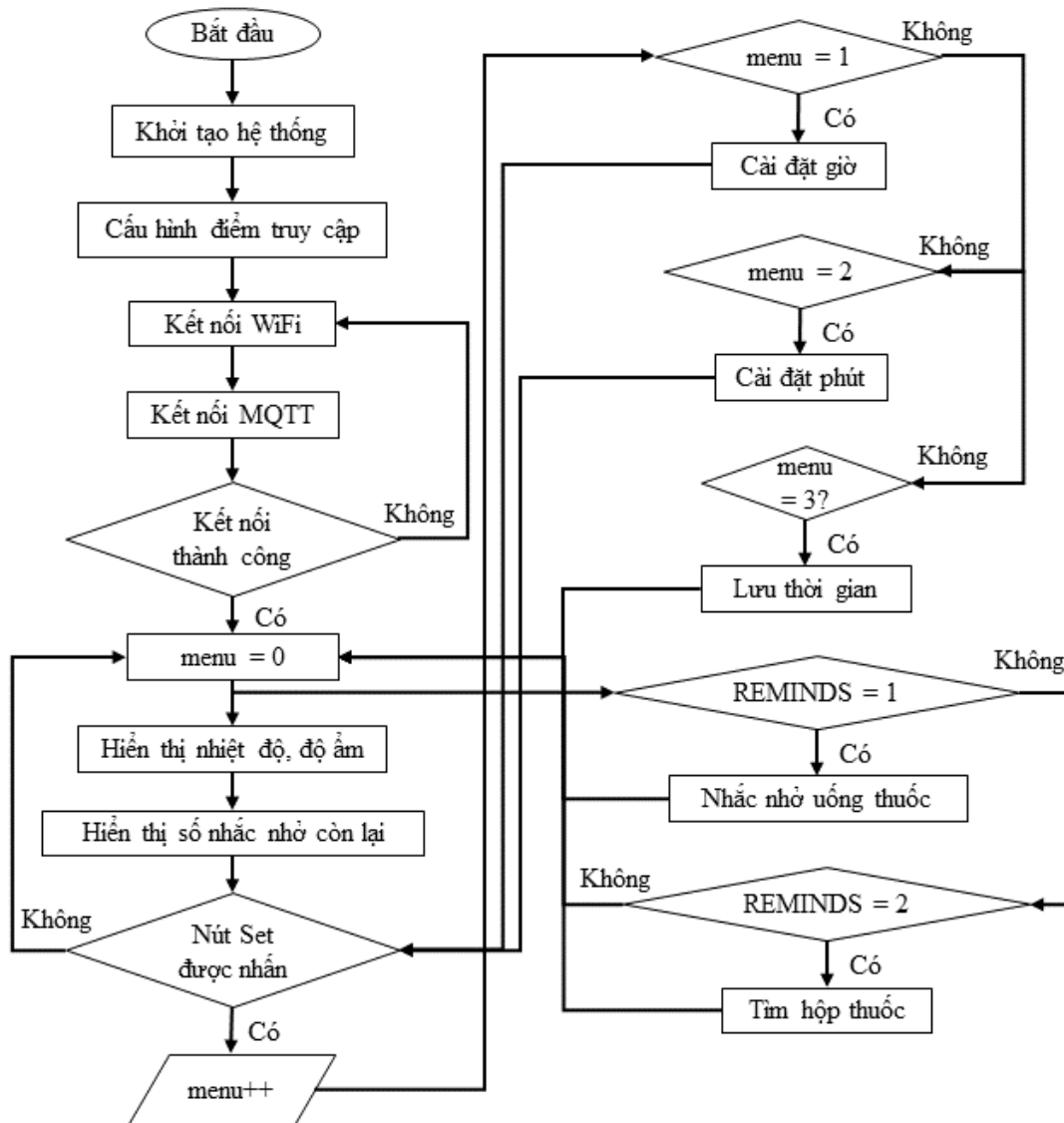


Hình 3.4 Sơ đồ nguyên lý khối hiển thị

3.3 Thiết kế phần mềm

3.3.1 Phần mềm cho vi điều khiển trung tâm

❖ **Chương trình chính:**



Hình 3.5 Sơ đồ thuật toán chương trình chính

❖ **Giải thích về hình ảnh:**

Khi được cấp nguồn, VDK trung tâm tiến hành khởi tạo hệ thống mặc định ban đầu, sau đó khởi tạo điểm truy cập để kết nối WiFi và MQTT, nếu kết nối không thành công thì sẽ kết nối lại đến lúc thành công.

Khi menu = 0, chương trình “Hiển thị nhiệt độ - độ ẩm” và “Số nhắc nhở còn lại” được thực thi.

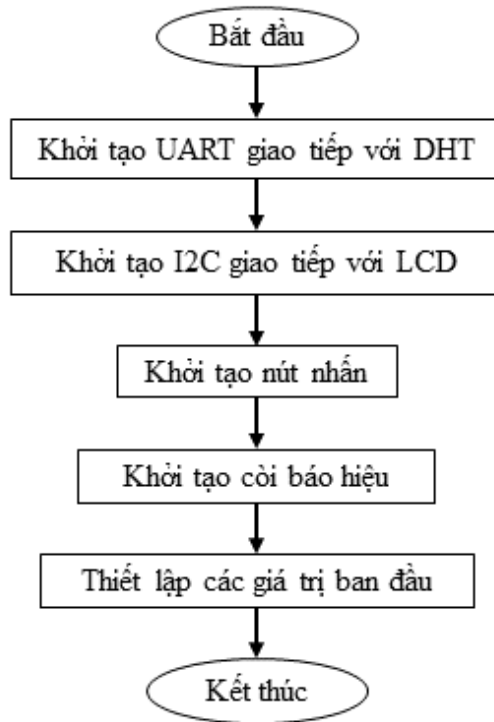
Nếu nút Set được nhấn thì tăng giá trị của menu lên 1 đơn vị.

Kiểm tra xem menu bằng bao nhiêu, nếu menu = 1, 2 hoặc 3 thì sẽ tiến hành cài đặt thời gian theo các chương trình lần lượt là “Cài đặt giờ”, “Cài đặt phút” và “Lưu thời gian”.

Nếu REMINDS = 1 thì chương trình phát cảnh báo nhắc nhở uống thuốc được thực thi, nếu REMINDS = 2 thì chương trình tìm hộp thuốc được thực thi. Nếu không, trở về lại chương trình menu = 0.

❖ **Chương trình con:**

Thuật toán chương trình con khởi tạo hệ thống:

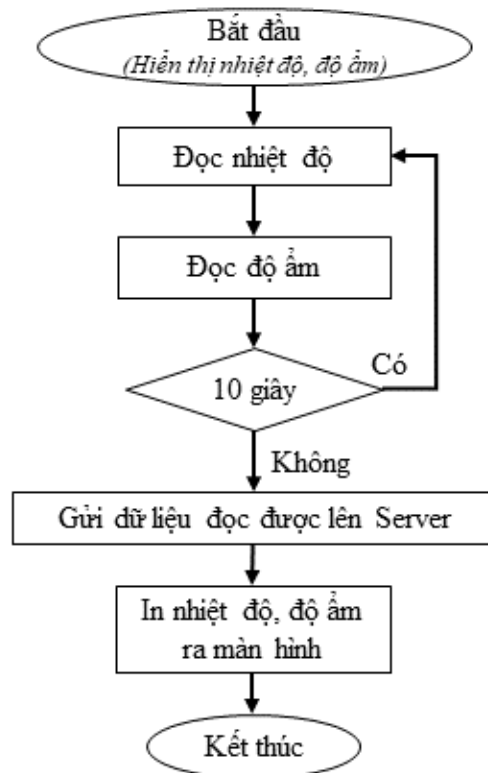


Hình 3.6 Sơ đồ thuật toán chương trình khởi tạo hệ thống

❖ **Giải thích về hình ảnh:**

Hình mô tả thuật toán khởi tạo hệ thống. VĐK sẽ cấu hình các chân GPIO theo chuẩn truyền thông nối tiếp UART để giao tiếp với các ngoại vi (DHT, LCD), chân vào/ra cho các nút nhấn và còi đi vào hoạt động ổn định. Tiếp đến, VĐK sẽ thiết lập các giá trị mặc định ban đầu.

Thuật toán hiển thị nhiệt độ, độ ẩm:

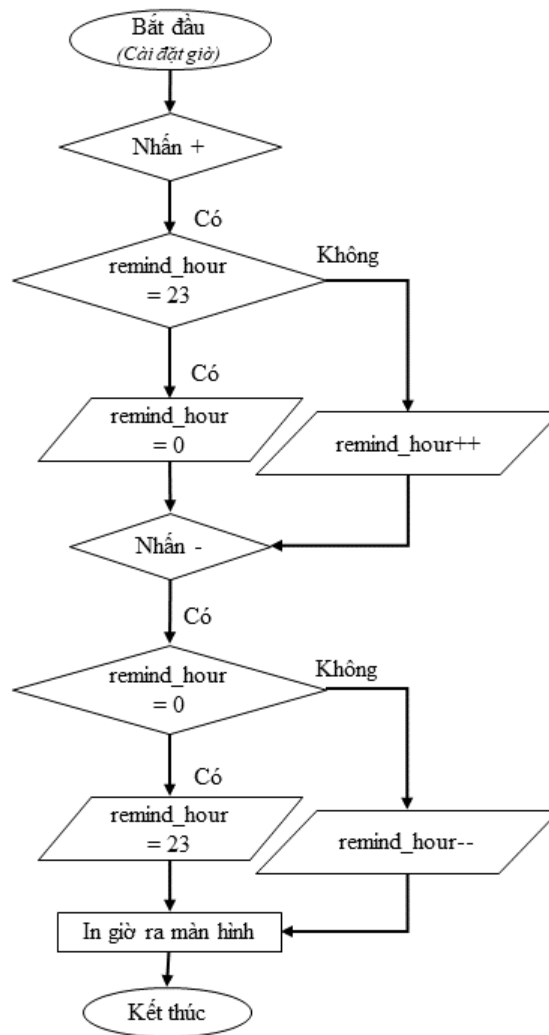


Hình 3.7 Sơ đồ thuật toán chương trình hiển thị độ ẩm, nhiệt độ

❖ **Giải thích về hình ảnh:**

Là thuật toán đọc và hiển thị nhiệt độ, độ ẩm lên màn hình LCD. Cảm biến DHT11 đọc thông tin nhiệt độ và độ ẩm sau đó gửi về cho VDK, cứ sau 10 giây đọc một lần. Thông tin về nhiệt độ và độ ẩm được VDK gửi lên Server và in ra màn hình LCD với mục đích giúp người dùng có thể theo dõi sự thay đổi nhiệt độ và độ ẩm, qua đó có phương pháp bảo quản thuốc phù hợp.

Thuật toán cài đặt giờ:



Hình 3.8 Sơ đồ thuật toán chương trình cài đặt giờ

❖ **Giải thích về hình ảnh:**

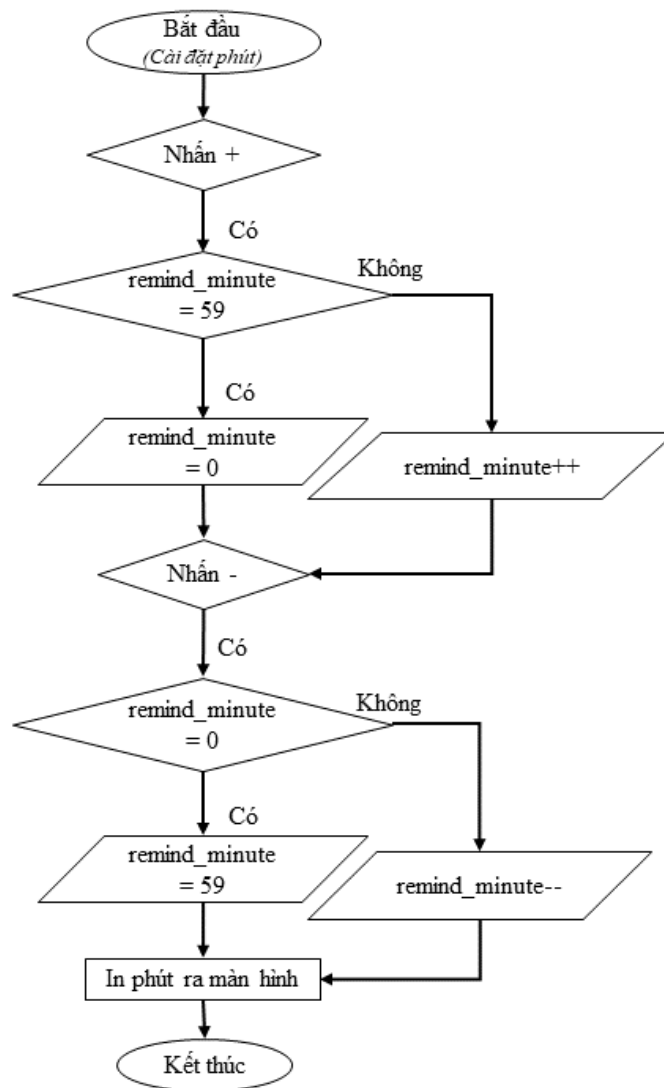
Khi vào tiến trình cài đặt giờ (menu = 1):

Nếu nút Plus (+) được nhấn thì tăng biến remind_hour (là biến lưu giá trị giờ) lên 1 đơn vị (nếu remind_hour đang có giá trị 23 thì khi nhấn nút Plus sẽ về 0).

Ngược lại, nếu nút Minus (-) được nhấn thì giảm biến remind_hour đi 1 đơn vị (nếu remind_hour đang có giá trị 0 thì khi nhấn nút Plus sẽ về 23).

Sau đó, in giá trị của biến remind_hour ra màn hình LCD.

Thuật toán cài đặt phút:



Hình 3.9 Sơ đồ thuật toán chương trình cài đặt phút

❖ **Giải thích về hình ảnh:**

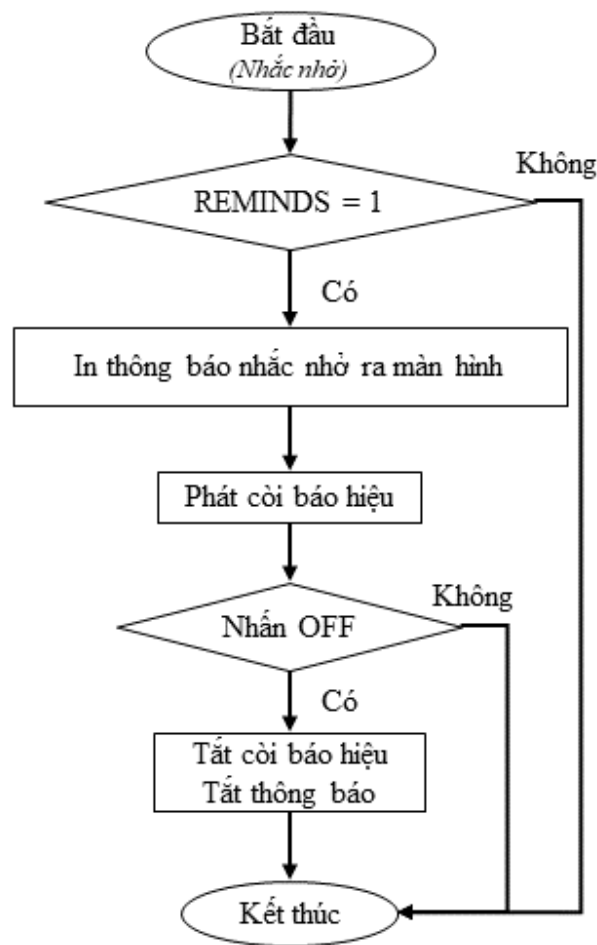
Tương tự như chương trình cài đặt giờ. Khi menu = 2 thì chương trình con cài đặt phút được thực thi:

Nếu nút Plus (+) được nhấn thì tăng biến remind_minute (là biến lưu giá trị phút) lên 1 đơn vị (nếu remind_minute đang có giá trị 59 thì khi nhấn nút Plus sẽ về 0).

Ngược lại, nếu nút Minus (-) được nhấn thì giảm biến remind_minute đi 1 đơn vị (nếu remind_minute đang có giá trị 0 thì khi nhấn nút Plus sẽ về 59).

Sau đó, in giá trị của biến remind_minute ra màn hình LCD.

Thuật toán nhắc nhở uống thuốc:



Hình 3.10 Sơ đồ thuật toán chương trình nhắc nhở

❖ **Giải thích về hình ảnh:**

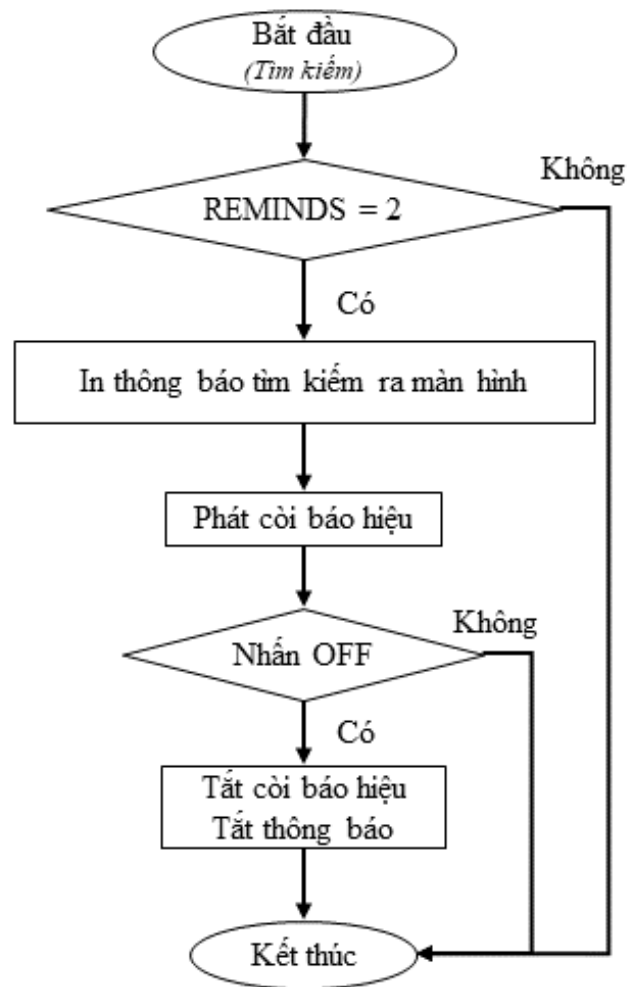
Là thuật toán chương trình con nhắc nhở uống thuốc theo thời gian đã cài đặt.

VĐK sẽ tiếp nhận giá trị biến REMINDS từ Server thông qua giao thức MQTT và kết nối WiFi.

Nếu $REMINDS = 1$ thì sẽ in thông báo nhắc nhở uống thuốc “Time to take medicines!” lên màn hình LCD, đồng thời phát cảnh báo bằng còi báo hiệu.

Bên cạnh đó, VĐK cũng sẽ lắng nghe xem nút nhấn Off có được nhấn hay không, nếu có thì sẽ tắt trạng thái nhắc nhở.

Thuật toán tìm hộp thuốc:



Hình 3.11 Sơ đồ thuật toán chương trình tìm hộp thuốc

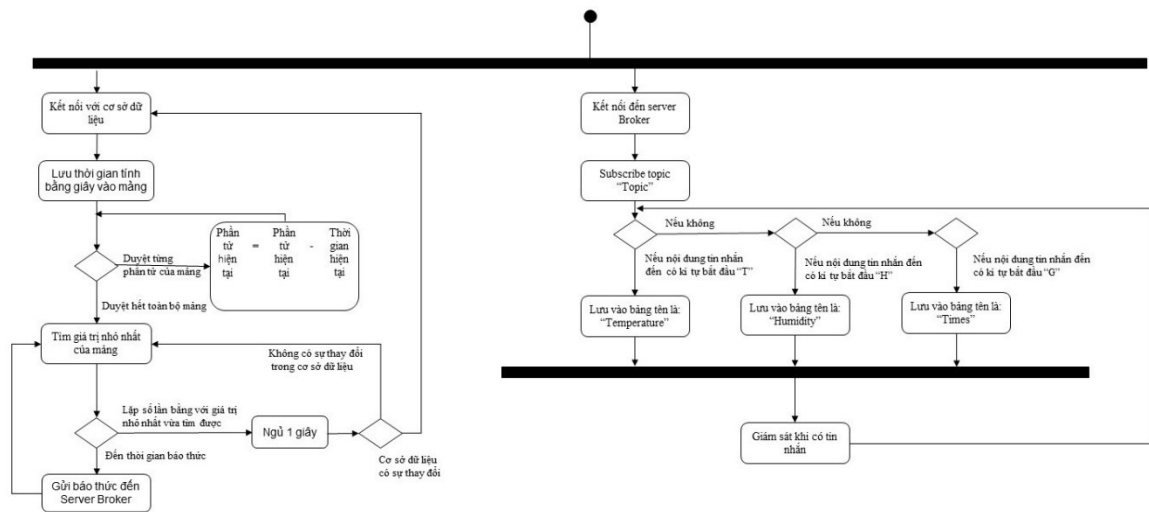
❖ **Giải thích về hình ảnh:**

Là thuật toán chương trình con tìm vị trí hộp thuốc.

Nếu $REMINDS = 2$ thì chương trình tìm hộp thuốc được thực thi và sẽ in dòng tin nhắn “I’m here!” lên màn hình LCD, đồng thời phát cảnh báo bằng còi báo hiệu.

Bên cạnh đó, VĐK cũng sẽ lắng nghe xem nút nhấn Off có được nhấn hay không, nếu có thì sẽ tắt trạng thái báo hiệu.

3.3.2 Phần mềm cho server



Hình 3.12 Giản đồ activity của server

❖ Giải thích về hình ảnh:

Trong quá trình xử lý server vừa lấy được thông tin thời gian từ cơ sở dữ liệu mà vẫn nhận được thông tin nhiệt độ và độ ẩm.

Giản đồ chia thành hai phần chính:

- Xử lý về thời gian báo thức. Server sẽ lấy dữ liệu từ cơ sở dữ liệu về xử lý sau đó tìm ra thời gian báo thức gần nhất phải thực hiện sau đó ngủ trong khoảng thời gian từ thời điểm hiện tại đến thời gian báo thức và sẽ gửi một tin nhắn đến Broker, nếu trong quá trình thực hiện mà kiểm tra thấy có sự thay đổi về thời gian nhắc nhở nó sẽ kết nối lại vào cơ sở dữ liệu để kiểm tra và lặp lại quá trình.
- Xử lý nhận thông tin về nhiệt độ và độ ẩm. Broker tiến hành đăng ký nhận dữ liệu từ ESP32 thông qua topic "Topic". Broker MQTT sẽ kiểm tra xem có nhận dữ liệu từ topic đã đăng ký hay không. Nếu không, sẽ đợi đến khi có dữ liệu đến.

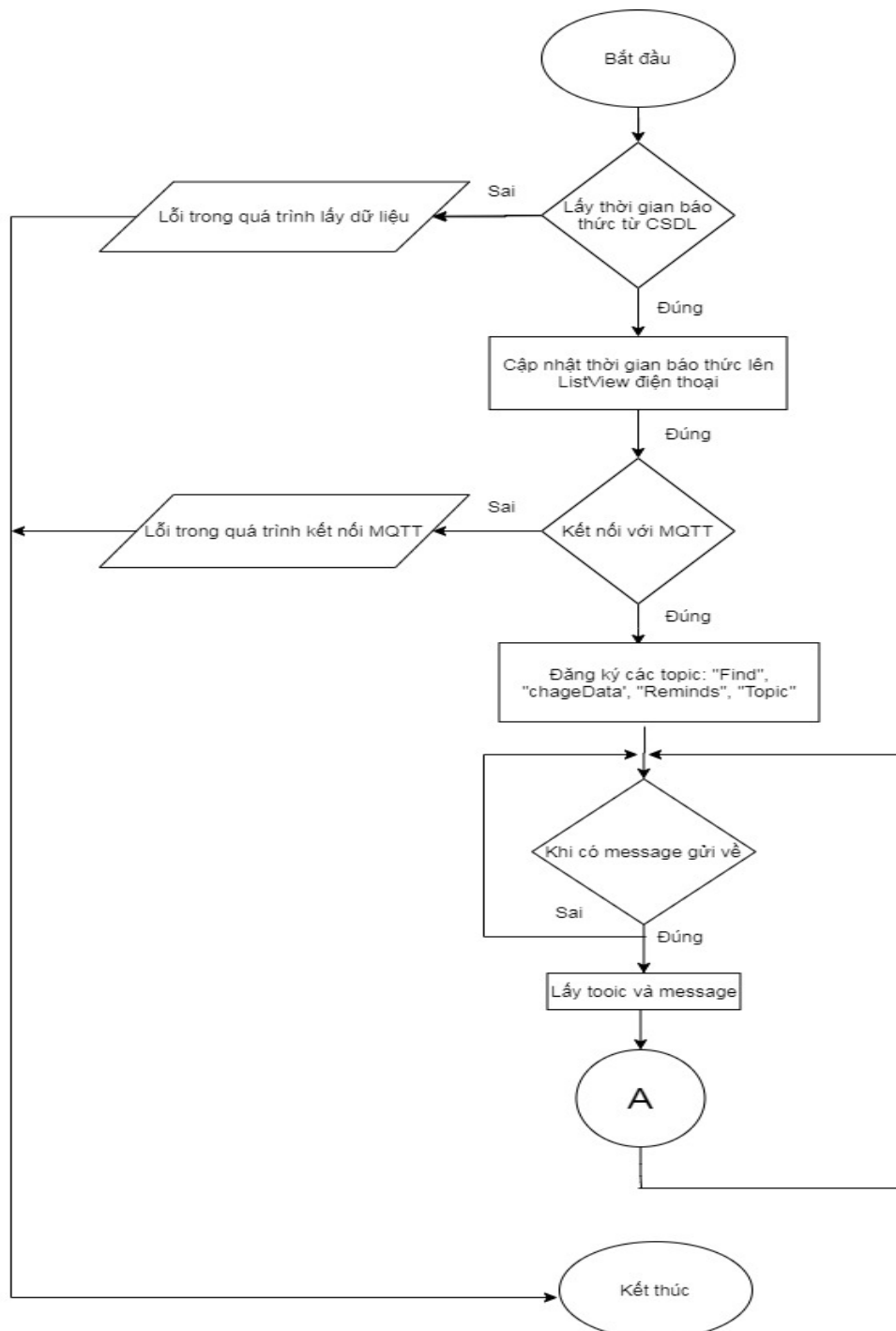
Dữ liệu gửi đến Broker có dạng:

- Chuỗi có dạng "H86" tức là độ ẩm 86%.
- Chuỗi có dạng "T33" tức là nhiệt độ 33°C.
- Chuỗi có dạng "G29160" tức là 29160 giây hay 08 giờ 06 phút.

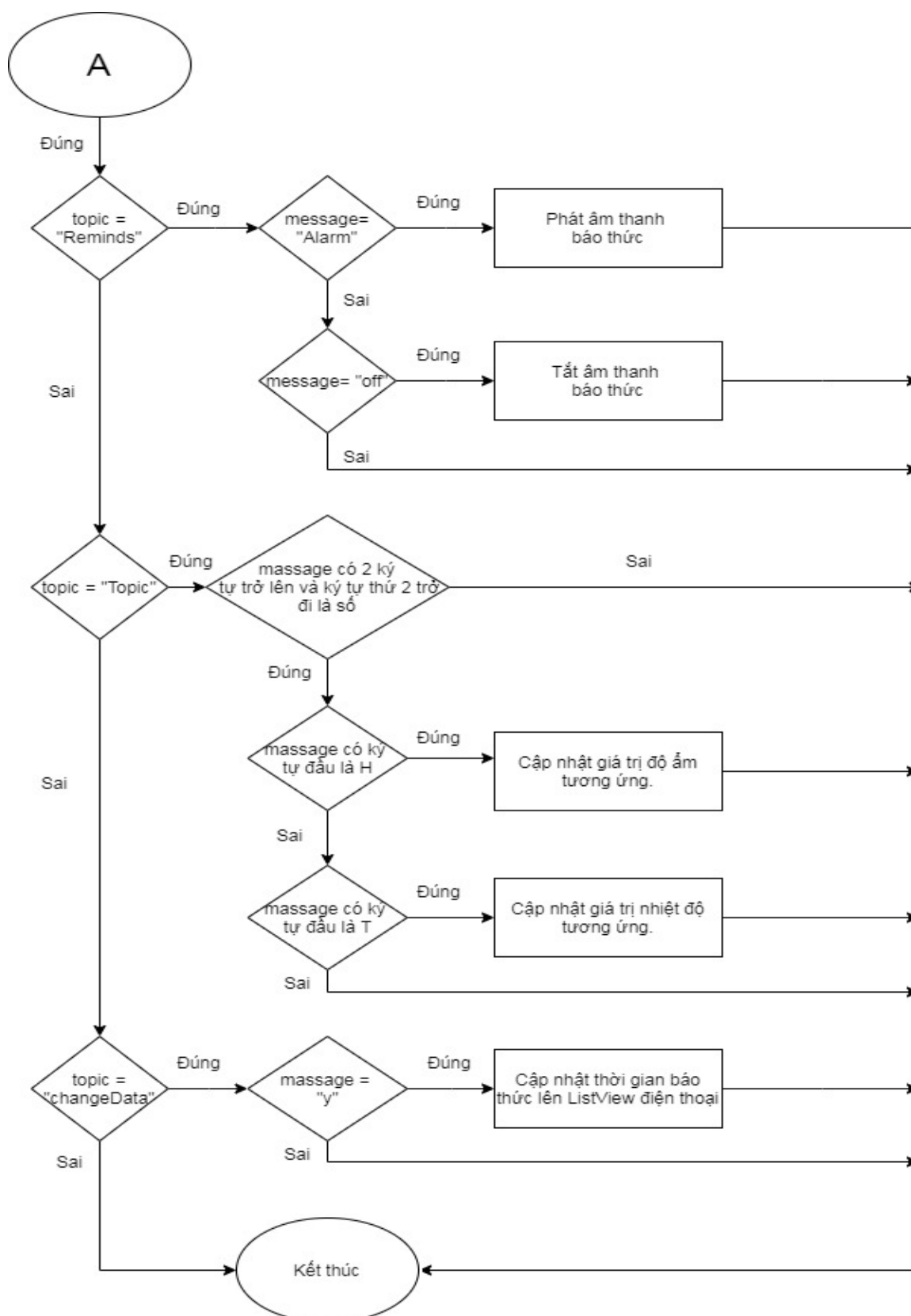
3.3.3 Phần mềm ứng dụng Android

Ứng dụng hỗ trợ cài đặt trên hệ điều hành Android. Ứng dụng có chức năng cài đặt báo thức và hiển thị nhiệt độ, độ ẩm kết hợp với tìm kiếm hộp đựng và phát ra thông báo nhắc nhở đến cho người dùng ngay cả khi điện thoại đang tắt.

❖ **Lưu đồ thuật toán:**



Hình 3.13 Sơ đồ thuật toán ứng dụng trên điện thoại (1)



Hình 3.14 Sơ đồ thuật toán ứng dụng trên điện thoại (2)

❖ Giải thích về hình ảnh

Nguyên lý hoạt động:

- Khi ứng dụng vừa được khởi động thì nó sẽ kết nối với database.
- Nếu kết nối thành công thì lấy dữ liệu thời gian ở database và đổ ra ListView của ứng dụng.
- Nếu kết nối không thành công thì sẽ báo lỗi “Lỗi trong quá trình lấy dữ liệu”.

Kết nối với MQTT:

- Nếu kết nối thành công thì đăng ký các topic để phục vụ cho việc đồng bộ dữ liệu và tìm kiếm hộp thuốc.
- Nếu kết nối không thành công thì sẽ báo lỗi “Lỗi trong quá trình kết nối MQTT”.
- Khi có dữ liệu gửi về từ các topic đã đăng ký ở trên thì xử lý.
- Nếu topic là “Reminds”.
- Nếu message là “Alarm” thì phát âm thanh báo thức.
- Nếu message là “off” thì tắt âm thanh báo thức.
- Nếu topic là “Topic”.
- Nếu message có 2 ký tự trở lên và ký tự thứ 2 trở đi là số.
- Nếu ký tự đầu tiên là “H” thì cập nhật giá trị độ ẩm.
- Nếu ký tự đầu tiên là “T” thì cập nhật giá trị nhiệt độ.
- Nếu topic là changeData.
- Nếu message là “y” thì đồng bộ dữ liệu ở database và ListView.

Bảng 3.1 Các topic đăng ký và nhiệm vụ tương ứng

Topic	Nhiệm vụ
changeData	Cập nhật lại ListView khi thay đổi database.
Reminds	Bật, tắt báo thức.
Topic	Nhiệt độ và độ ẩm được gửi về.

3.4 Thiết kế Server và Website

3.4.1 Server

Là một máy tính, đã cài đặt MQTT broker, nơi chạy thuật toán xử lý, truyền, nhận và xử lý gói tin đến từ client (ESP32), lưu trữ dữ liệu sau khi xử lý vào các bảng cơ sở dữ liệu. Python là ngôn ngữ chính được dùng trên server. Ngoài ra, server còn chứa những website và các thông tin liên quan khác.

Phương thức truyền dữ liệu: Server và khối truyền nhận dữ liệu qua WiFi (ESP32) sẽ giao tiếp truyền dữ liệu thông qua giao thức MQTT, nghĩa là client (ESP32) sẽ publish 1 gói tin A lên Broker qua topic B, phía bên server sẽ nhận được gói tin đó (do server subscribe topic B) sau đó phân tích dữ liệu nhận được, tiến hành xử lý rồi lưu vào cơ sở dữ liệu sau đó sẽ phản hồi (publish) lại cho ESP32 gói tin C thông qua topic B. ESP32 cũng subscribe topic B nên sẽ nhận được gói tin C. ESP32 nhận được rồi xử lý.

3.4.2 Website

Website, còn gọi là trang web hoặc trang mạng, là một tập hợp trang web, thường chỉ nằm trong một tên miền hoặc tên miền phụ trên World Wide Web của Internet. Một trang web là tập tin HTML hoặc XHTML có thể truy nhập dùng giao thức HTTP. Trang web được lưu trữ trên máy chủ (server) có thể truy cập thông qua Internet hoặc dưới local (các máy tính có kết nối chung với server trên cùng 1 mạng).

❖ **Các chức năng chính của website:**

Hiện thị giao diện để người dùng có thể thiết lập các nhắc nhở uống thuốc cá nhân. Giám sát nhiệt độ và độ ẩm, đưa ra cảnh báo bằng màu sắc ở phía bên dưới hộp thuốc.

❖ **Cấu trúc các bảng trong cơ sở dữ liệu “day”:**

Bảng 3.2 Bảng danh sách nhắc nhở

Số thứ tự	Tên trường	Giải thích	Kiểu	Ghi chú
1	Id	Số thứ tự	Int(11)	Khóa chính
2	Hour	Giờ	Int(255)	
3	Minute	Phút	Int(255)	
4	Seconds	Giây	Int(255)	

Bảng 3.3 Bảng độ ẩm

Số thứ tự	Tên trường	Giải thích	Kiểu	Ghi chú
1	Id	Số thứ tự	Int(11)	Khóa chính
2	Humidity	Độ ẩm	Int(255)	
3	Time	Thời gian	Timestamp	

Bảng 3.4 Bảng nhiệt độ

Số thứ tự	Tên trường	Giải thích	Kiểu	Ghi chú
1	Id	Số thứ tự	Int(11)	Khóa chính
2	Temperature	Nhiệt độ	Int(255)	
3	Time	Thời gian	Timestamp	

3.5 Kết luận chương

Trên đây nhóm đã trình bày xong về các thiết kế phần cứng và phần mềm của hệ thống, lưu đồ thuật toán và các thành phần liên quan, thiết kế của server và website của hệ thống. Chương kế tiếp chúng em sẽ trình bày về thi công, kiểm tra kết quả, đánh giá hệ thống.

CHƯƠNG 4: THI CÔNG, KIỂM TRA KẾT QUẢ VÀ ĐÁNH GIÁ HỆ THỐNG

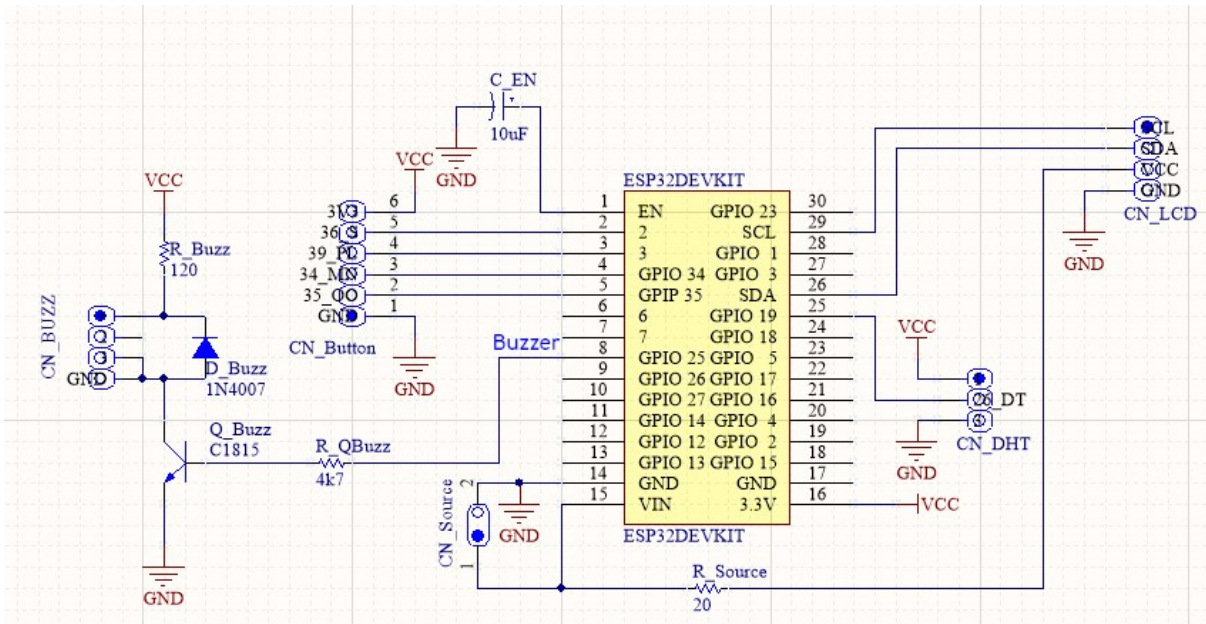
4.1 Giới thiệu chương

Trong chương này sẽ trình bày kết quả thi công của hệ thống gồm phần cứng, phần mềm; đánh giá và đưa ra hướng phát triển của đề tài.

4.2 Thi công mạch

4.2.1 Vẽ mạch trên phần mềm

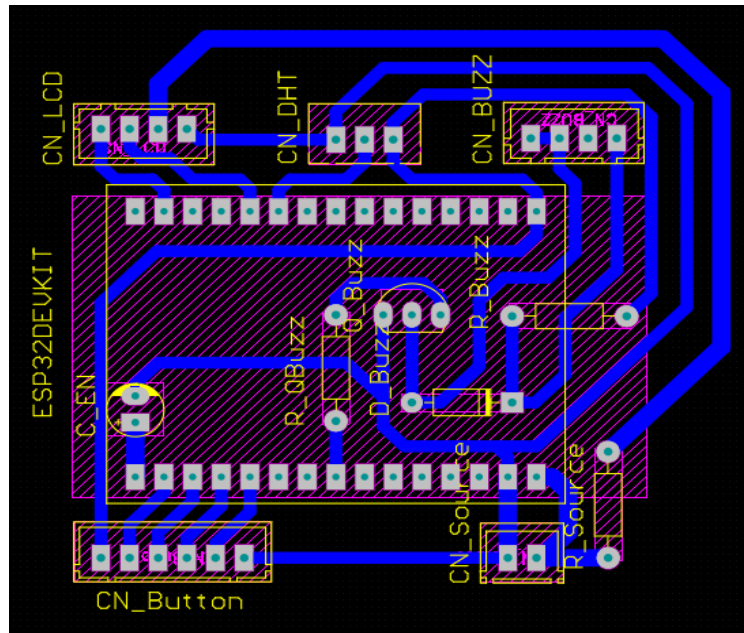
Mạch được vẽ bằng phần mềm Altium Designer (phiên bản 18.1.6). Dưới đây là những hình ảnh của mạch:



Hình 4.1 Sơ đồ nguyên lý của mạch

❖ Giải thích về hình ảnh:

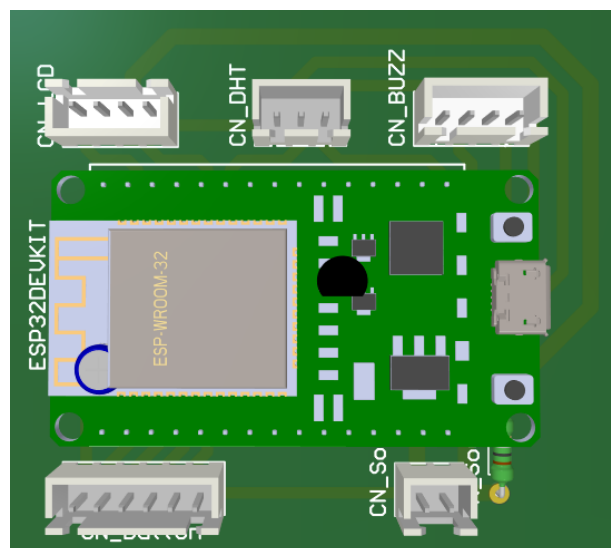
Là sơ đồ nguyên lý của mạch, hình ảnh mô tả kết nối giữa vi điều khiển trung tâm với các thiết bị theo đúng chức năng các chân của linh kiện.



Hình 4.2 Sơ đồ đi dây trên mạch in

❖ **Giải thích về hình ảnh:**

Hình ảnh của PCB, mô tả sơ đồ nối dây từ chân (pin) của linh kiện này đến chân (pin) của linh kiện khác trong mạch. Sơ đồ này được sử dụng để làm mạch in, in lên bảng mạch đồng và hàn các linh kiện lên bảng mạch đó (xem ảnh thực tế).

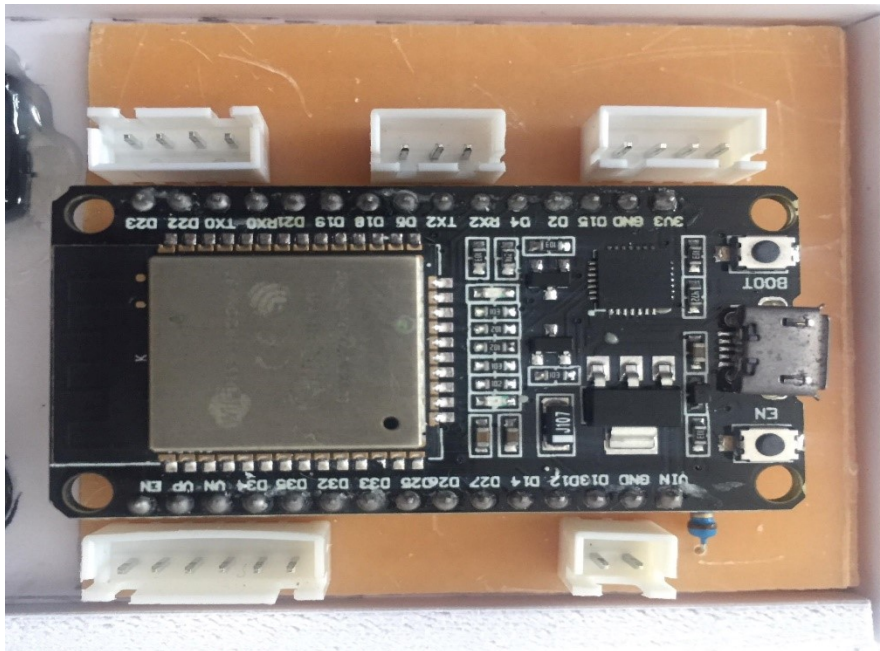


Hình 4.3 Hình ảnh 3D của mạch

❖ **Giải thích về hình ảnh:**

Là hình ảnh mô phỏng 3D của mạch trên phần mềm Altium Designer. Thông qua hình ảnh mô phỏng, chúng ta có thể hình dung trước được cách sắp xếp các linh kiện sau khi hoàn thành sẽ như thế nào, qua đó cũng có thể hiệu chỉnh vị trí linh kiện sao cho phù hợp, gọn gàng, thẩm mỹ.

4.2.2 Mạch thực tế



Hình 4.4 Board mạch xử lý trung tâm sau khi hoàn thiện

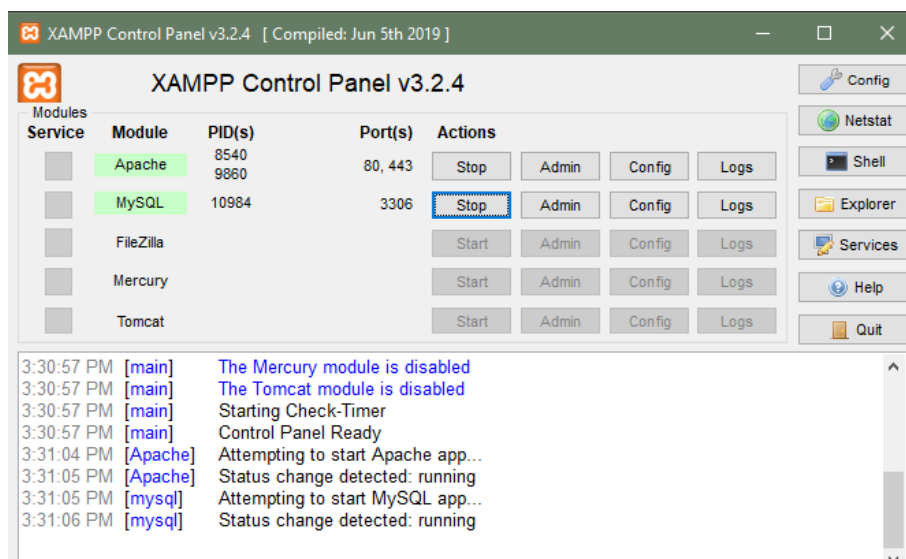
❖ Giải thích về hình ảnh:

Là hình ảnh thực tế của board mạch xử lý trung tâm sau khi đã hàn các linh kiện vào bảng mạch đồng. Hình ảnh này giống với hình ảnh mô phỏng 3D đến 99%.

4.3 Thi công website và tạo cơ sở dữ liệu

4.3.1 XAMPP

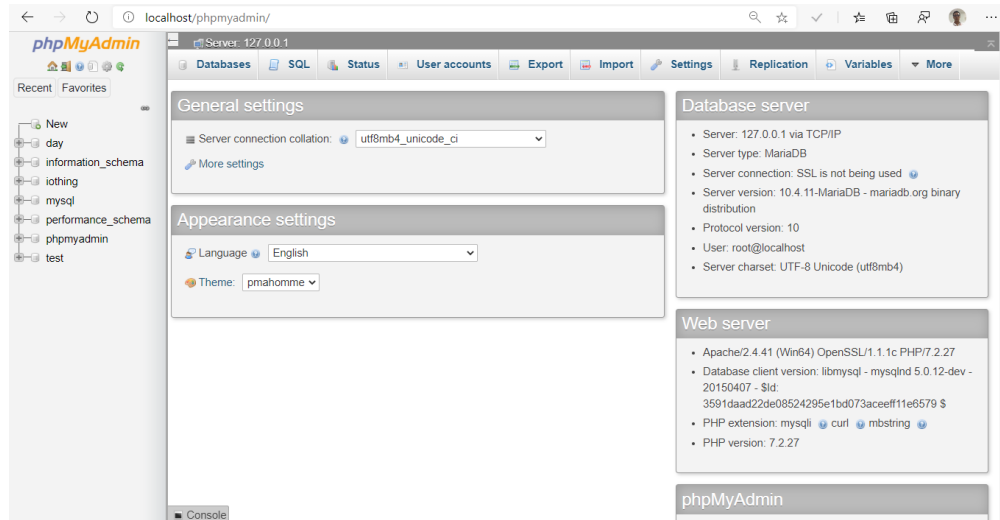
XAMPP là chương trình tạo máy chủ Web được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server và các công cụ như phpMyAdmin. Localhost là một máy chủ được vận hành trên máy tính.



Hình 4.5 Giao diện phần mềm XAMPP

❖ **Giải thích về hình ảnh:**

Để có thể tạo ra trang web thì người dùng phải lưu mã nguồn vào thư mục htdocs trong xampp tại địa chỉ C:\xampp\htdocs của máy tính.



Hình 4.6 Giao diện hệ quản trị cơ sở dữ liệu

❖ **Giải thích về hình ảnh:**

Để khởi tạo cơ sở dữ liệu thì truy cập vào trình duyệt theo địa chỉ <http://localhost/phpmyadmin/>.

4.3.2 Cơ sở dữ liệu

Table	Action	Rows	Type	Collation	Size	Overhead
humidity	Browse Structure Search Insert Empty Drop	666	InnoDB	utf8mb4_general_ci	48.0 KiB	-
temperature	Browse Structure Search Insert Empty Drop	1,127	InnoDB	utf8mb4_general_ci	64.0 KiB	-
times	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	16.0 KiB	-
3 tables	Sum	1,801	InnoDB	utf8mb4_general_ci	128.0 KiB	0 B

Hình 4.7 Danh sách các bảng trong cơ sở dữ liệu

❖ **Giải thích về hình ảnh :**

Phía bên cạnh trái là bố cục của cơ sở dữ liệu. Dưới đây là tên cơ sở dữ liệu và các bảng độ ẩm (humidity), nhiệt độ (temperature), thời gian (time) thành phần nằm bên trong.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	humidity	int(255)			No	None			Change Drop More
3	time	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Hình 4.8 Cấu trúc bảng độ ẩm

❖ **Giải thích về hình ảnh :**

Bảng mô tả kiểu dữ liệu và thành phần các cột của bảng độ ẩm.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	temperature	int(255)			No	None			Change Drop More
<input type="checkbox"/> 3	time	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

☐ Check all With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#)

Hình 4.9 Cấu trúc bảng nhiệt độ

❖ **Giải thích về hình ảnh :**

Bảng mô tả kiểu dữ liệu và thành phần các cột của bảng nhiệt độ.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	hour	int(255)			No	None			Change Drop More
<input type="checkbox"/> 3	minute	int(255)			No	None			Change Drop More
<input type="checkbox"/> 4	seconds	int(255)			No	None			Change Drop More


☐ Check all With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#)

Hình 4.10 Cấu trúc bảng thời gian

❖ **Giải thích về hình ảnh :**

Bảng mô tả kiểu dữ liệu và thành phần các cột của bảng thời gian.

4.3.3 Trang chủ web

 SMART BOX Home Sensor About us

CONTROL PANEL

set up alarm and viewing humidities and temperatures

Select your time

Chọn giờ

Select hour

Chọn phút

Select Minute

Submit

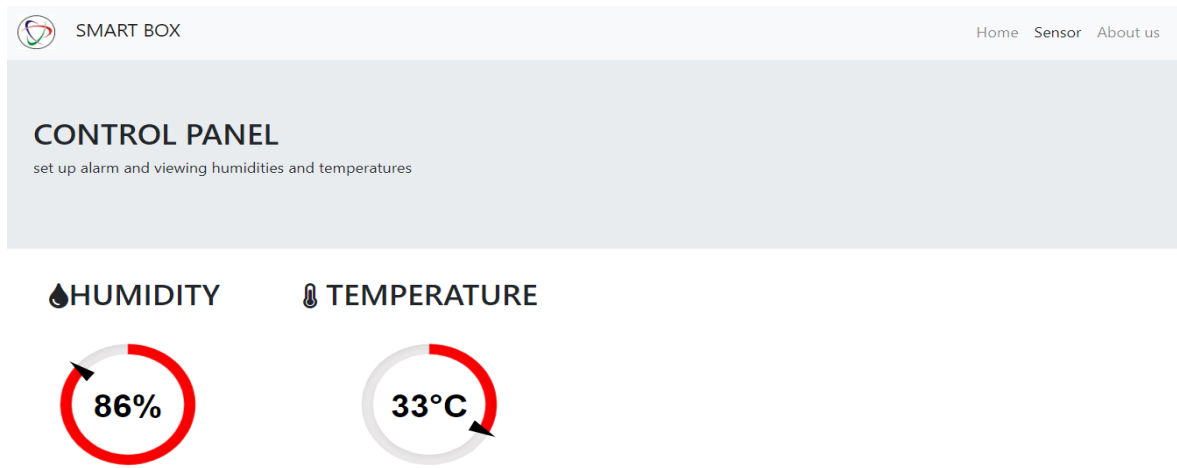
List of time

STT	Hour	Minute	Action
1	14	43	DELETE
2	14	50	DELETE

Hình 4.11 Giao diện chính

❖ **Giải thích về hình ảnh:**

Theo mặc định khi truy cập vào trang chủ thì người dùng sẽ thấy giao diện như trên hình. Tại đây người dùng có thể thêm mới báo thức ngay trên giao diện của web sau đó trên danh sách sẽ hiện ra những nhắc nhở mà người dùng đã cài đặt.

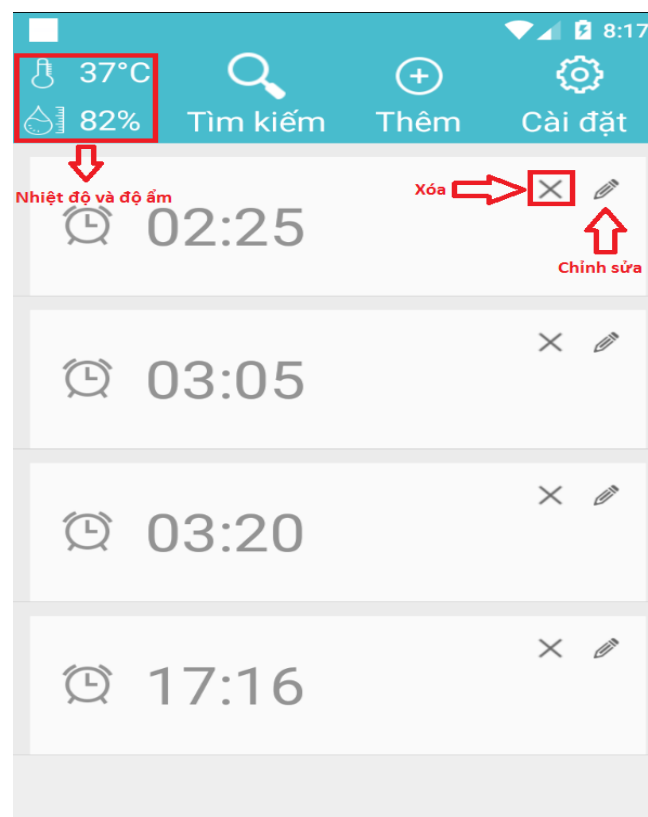


Hình 4.12 Hiển thị thông tin về độ ẩm và nhiệt độ

❖ **Giải thích về hình ảnh:**

Khi nhấn vào thanh menu trên cùng ta sẽ thấy thẻ Sensor đây là nơi người dùng có thể quan sát thông tin về độ ẩm và nhiệt độ. Hệ thống cũng đưa ra cảnh báo bằng màu sắc nếu độ ẩm và nhiệt độ vượt quá định mức, tiện lợi cho việc bảo quản thuốc.

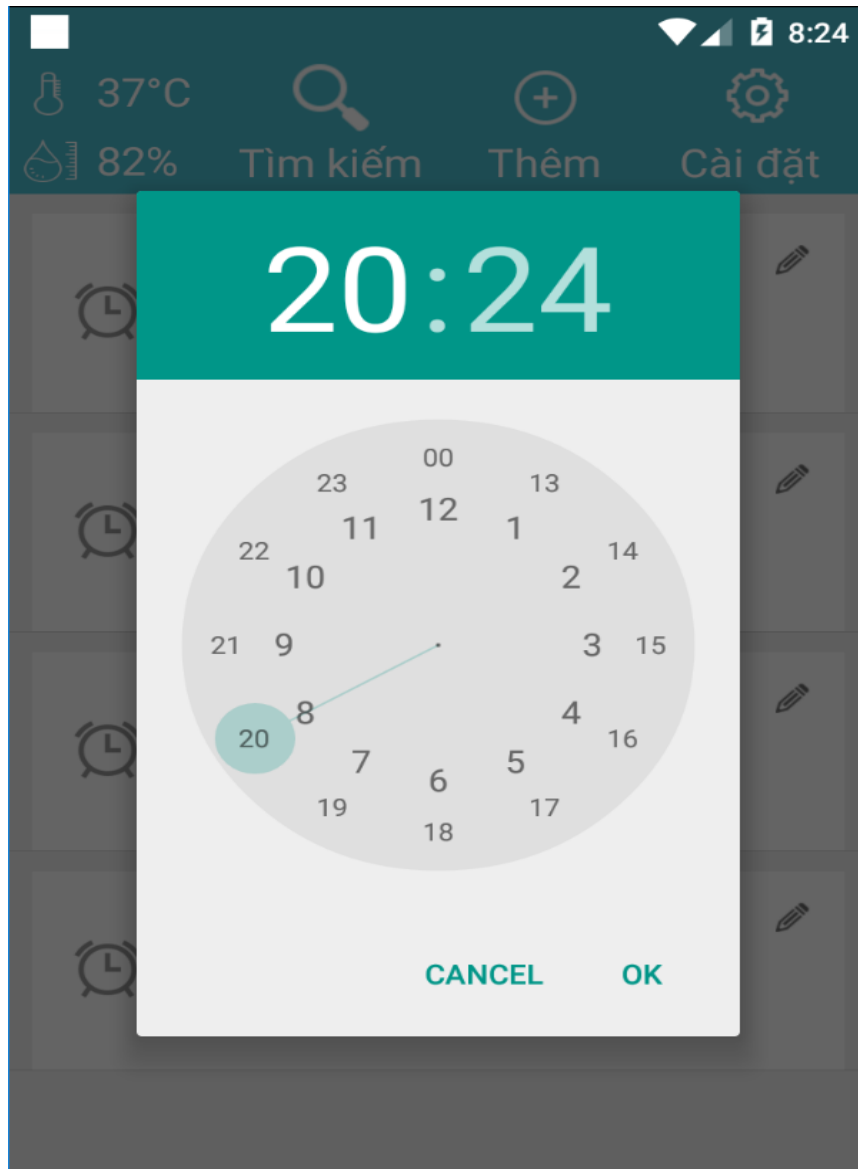
4.4 Xây dựng ứng dụng Android



Hình 4.13 Giao diện quản lý chung

❖ **Giải thích về hình ảnh:**

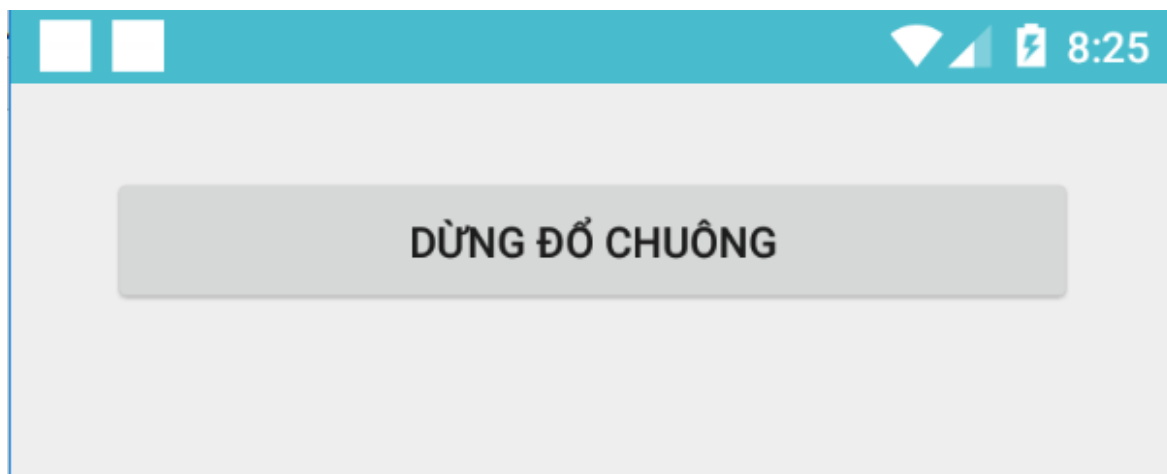
Người dùng có thể thêm, xóa, chỉnh sửa thời gian thông báo, tìm kiếm hộp thuốc và xem thông tin về nhiệt độ và độ ẩm hiện tại ở bên trong hộp thuốc.



Hình 4.14 Giao diện khi muốn thêm một báo thức mới

❖ **Giải thích về hình ảnh:**

Giao diện người dùng muốn thêm báo thức. Nếu nhấn OK thì báo thức sẽ được thêm vào database và sẽ cập nhật lên giao diện listview của điện thoại. Nếu nhấn vào CANCEL hoặc chạm tay vào vị trí ngoài khung lựa chọn giờ thì sẽ hủy việc thêm.



Hình 4.15 Giao diện khi đến giờ uống thuốc

❖ **Giải thích về hình ảnh:**

Khi đến giờ uống thuốc, server sẽ gửi thông báo về điện thoại, lúc đó điện thoại sẽ chuyển sang màn hình như hình 4.15 kết hợp với đồ chuông.

4.5 Kết quả thực nghiệm và đánh giá hệ thống

4.5.1 Kết quả thực nghiệm

❖ **Mô hình sản phẩm**



Hình 4.16 Mô hình sản phẩm mặt trước và sau

❖ **Màn hình chính:**



Hình 4.17 Giao diện mặc định

❖ **Cài đặt giờ:**



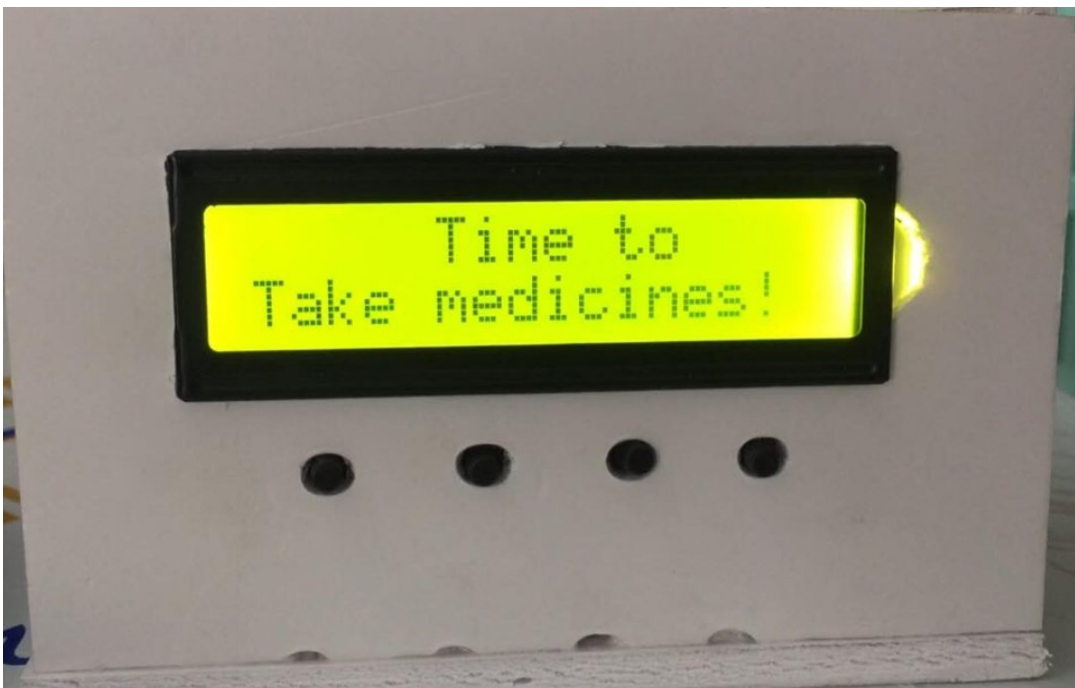
Hình 4.18 Giao diện cài đặt giờ

❖ **Cài đặt phút:**



Hình 4.19 Giao diện cài đặt phút

❖ **Nhắc nhở uống thuốc:**



Hình 4.20 Giao diện nhắc nhở

❖ **Tìm hộp thuốc:**



Hình 4.21 Giao diện tìm hộp thuốc

4.5.2 Đánh giá hệ thống

❖ **Ưu điểm:**

Hoàn thành hết chức năng ban đầu đặt ra cho hệ thống.

Hệ thống hoạt động khá ổn định, tốc độ xử lý nhanh do sử dụng giao thức MQTT.

Giao diện website đơn giản, bảng biểu trực quan, sinh động, thông báo rõ ràng, khách hàng dễ dàng sử dụng.

❖ **Nhược điểm:**

Thời gian thực hiện đề tài còn hạn chế nên một số chức năng không thể tránh khỏi những thiếu sót nhất định.

Cần hoạt động trong môi trường có WiFi.

Mô hình phần cứng chưa thật sự nhỏ gọn, thẩm mỹ.

4.6 Kết luận chương

Như vậy, chúng em đã trình bày xong toàn bộ những điểm chính của đề tài qua bốn chương. Tiếp theo là phần kết luận và hướng phát triển.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

❖ Kết luận:

Với đề tài “**Nhắc nhở uống thuốc dựa trên nền tảng IoT**” nếu có thể tạo nhiều điều kiện tốt để nghiên cứu, phát triển hoàn thiện cùng với những chức năng cần thiết, đề tài này sẽ rất có triển vọng, nhất là đối với những cá nhân luôn phải mang theo thuốc bên mình.

Kết quả đạt được: Đã xây dựng được website tương tác với người dùng, ứng dụng điện thoại hỗ trợ thao tác thiết lập nhắc nhở, mô hình phần cứng tương tác cơ học với người dùng.

❖ Hướng phát triển cho đề tài:

Nâng cấp các phiên bản tiếp theo cho hệ thống ở 3 mảng: website, mô hình phần cứng và ứng dụng điện thoại.

Thu nhỏ kích cỡ của hộp thuốc phù hợp để tiện lợi khi mang theo bên mình. Đồng thời, thiết kế các loại mô hình thẩm mỹ, gọn gàng hơn. Sử dụng mạch với linh kiện dân dụng thay vì linh kiện cấm.

Sử dụng một phương thức truyền khác song song thay vì chỉ sử dụng được WiFi, đề phòng trường hợp sử dụng tại địa điểm không có sóng WiFi.

Thêm khả năng định vị hộp thuốc để nâng cấp tính năng tìm hộp thuốc.

Thêm thông tin về điều kiện bảo quản thuốc, cũng như tối ưu ứng dụng của cảm biến nhiệt độ, độ ẩm.

Tối ưu phần mềm hơn và thêm một số chức năng thiết yếu khác sau quá trình nghiên cứu thêm.

Phát triển hệ thống trên tên miền Internet thay vì sử dụng localhost.

Toàn bộ dữ liệu về đề tài này sẽ được đưa lên trang GitHub cá nhân với địa chỉ: <https://github.com/lhp1507/medication-reminder-system> để phục vụ cho mục đích nghiên cứu và phát triển.

TÀI LIỆU THAM KHẢO

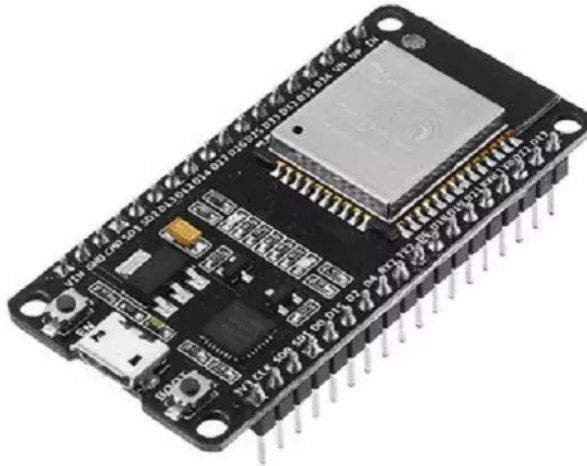
- [1] <https://codelearn.io/learning/java-co-ban>
- [2] <https://codelearn.io/learning/javascript-co-ban>
- [3] <https://codelearn.io/learning/python-co-ban>
- [4] [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
- [5] <https://en.wikipedia.org/wiki/JavaScript>
- [6] [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [7] <https://en.wikipedia.org/wiki/PHP>
- [8] <https://en.wikipedia.org/wiki/HTML>
- [9] <https://quantrimang.com/gioi-thieu-ve-html-149822>
- [10] <https://tapit.vn/http-request-va-http-response-phuong-thuc-giao-tiep-giua-server-client/>
- [11] <https://tapit.vn/hieu-dung-ve-moi-quan-qos-giua-publisher-va-subscriber-trong-giao-thuc-mqtt/>
- [12] https://viettelidc.com.vn/tin-tuc/server-la-gi-may-chu-la-gi-nhung-dieu-can-biet-ve-server?gclid=CjwKCAjw5vz2BRAtEiwAbcVIL88Kg6WPENyWNYn8MoaP7cjX Xk6kDI7k6iUxVnaJ4Py7BHBRIJBWKR0C6jwQAvD_BwE
- [13] https://www.bogotobogo.com/python/Multithread/python_multithreading_subclassing_Timer_Object.php
- [14] <https://getbootstrap.com/docs/4.5/getting-started/introduction/>
- [15] <https://github.com/bblanchon/ArduinoJson>
- [16] <https://espeasy.readthedocs.io/en/latest/Reference/GPIO.html#best-pins-to-use-on-esp32>
- [17] <https://bandardalat.com/2018/02/22/p5-dong-ho-hen-gio-bao-thuc/>
- [18] https://github.com/khoih-prog/ESP_WiFiManager
- [19] <https://tapit.vn/chuc-nang-nhap-xuat-tin-hieu-gpio-tren-nodemcu-esp32-dev-kit-va-nhung-luu-y-khi-su-dung/>
- [20] <https://stackoverflow.com/questions/52399683/attempting-mqtt-connection-failed-rc-2-try-again-in-5-seconds>
- [21] <http://arduino.vn/bai-viet/1550-loi-pho-thong-khi-dung-lcd-i2c-va-luu-y-cach-su-dung-voi-thu-vien-lcd-i2c>
- [22] <https://arduinookit.vn/giao-tiep-i2c-lcd-arduino>
- [23] <http://arduino.vn/tutorial/1570-gioi-thieu-module-esp32-va-huong-dan-cai-trinh-bien-dich-tren-arduino-ide>
- [24] <https://arduinookit.vn/doc-cam-bien-nhiet-do-do-am-dht11-arduino>
- [25] <https://www.arduino.cc/en/Tutorial.StringToIntExample>
- [26] <https://xemtailieu.com/tai-lieu/giam-sat-dieu-khien-thiet-bi-dien-chieu-sang-trong-nha-tu-xa-qua-mang-dien-thoai-1490408.html>
- [27] <https://playground.arduino.cc/Main/I2cScanner/>
- [28] <https://www.dientuspider.com/> (Hình ảnh và thông tin linh kiện)
- [29] <https://www.google.com/> (Thông tin linh kiện)
- [30] <https://lastminuteengineers.com/esp32-arduino-ide-tutorial/>

- [31] <https://randomnerdtutorials.com/solved-failed-to-connect-to-esp32-timed-out-waiting-for-packet-header/>
- [32] <https://www.slideshare.net/trongthuy2/luan-van-he-thong-giam-sat-nong-nghiep-bang-cong-nghe-iot-hay>

PHỤ LỤC

Giới thiệu về linh kiện

❖ Kit RF WiFi + Bluetooth ESP32



Hình PL.1 Kit RF WiFi + Bluetooth ESP32

- Module trung tâm: WiFi BLE SoC ESP32 ESP-WROOM-32.
- Nguồn sử dụng: 5VDC từ cổng micro USB hoặc 7V-12V (khuyến dùng) vào chân Vin trên Kit.
- Tích hợp mạch nạp và giao tiếp UART CP2102.
- Ra chân đầy đủ module ESP32, chuẩn chân cắm 2.54mm.
- Tích hợp LED Status, nút nhấn BOOT và ENABLE.
- Kích thước: 28.33 x 51.45mm.

❖ Màn hình LCD 1602



Hình PL.2 Màn hình LCD 1602

- Điện áp: -0.3V – 7V.
- Hoạt động ổn định: 2.7V – 5.5V.

- Điện áp ra mức cao: $>2.4V$.
- Điện áp ra mức thấp: $<0.4V$.
- Dòng điện cấp nguồn: $350\mu A - 600\mu A$.
- Nhiệt độ hoạt động: $-30^{\circ} - 75^{\circ}C$.
- Hiển thị 2 dòng, mỗi dòng 16 ký tự.
- Chữ đen, nền xanh lá.
- Kích thước: $80 \times 36 \times 12.5mm$.

❖ Mạch chuyển đổi I2C cho LCD



Hình PL.3 Module I2C

- Điện áp hoạt động: $3 - 6V$.
- Chuẩn giao tiếp: I2C.
- Dải địa chỉ I2C: $0x20$ đến $0x27$.
- Địa chỉ mặc định: $0x27$.
- Có điều chỉnh độ tương phản bằng vi trở.
- Kích thước: $41.5 \times 19 \times 15.3mm$.

❖ Module cảm biến nhiệt độ và độ ẩm DHT11



Hình PL.4 Module DHT11

- Điện áp hoạt động: $3 - 5V$.
- Dải độ ẩm hoạt động: $20 - 90\%$, sai số $\pm 5\%$.
- Dải nhiệt độ hoạt động: $0 - 50^{\circ}C$, sai số $\pm 2^{\circ}C$.

- Khoảng cách truyền tối đa: 20m.
- Tần số lấy mẫu tối đa: 1Hz (1 giây/ 1 lần).
- Kích thước: 28 x 12 x 10mm.

❖ Còi chip 5V



Hình PL.5 Còi chip 5V

- Điện áp hoạt động: 3.5 – 5.5V.
- Dòng điện tiêu thụ: < 25mA.
- Tần số cộng hưởng: 2300 Hz \pm 500 Hz.
- Biên độ âm thanh: > 80dB.
- Nhiệt độ hoạt động: -20 – 70°C.
- Kích thước: Đường kính 12mm, cao 9.7mm.

❖ Nguồn sử dụng



Hình PL.6 Pin Lithium Battery 3.7V, 2200mAh

- Điện áp: 3.7V (Khi đầy pin lên đến 4.2V).
- Pin được sử dụng nhiều lần.
- Kiểu: 18650.
- Dung lượng: 2200mAh.
- Khuyến cáo: Không sử dụng pin < 2.5V.
- Kích thước: 18 x 65mm.