

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH

VIÊN MINH TÂN
PHẠM QUỐC ĐĂNG
PHẠM TRUNG QUỐC
PHẠM TRỌNG HUỖNH

ĐỒ ÁN MÔN HỌC THIẾT KẾ HỆ THỐNG NHÚNG
ĐỒNG HỒ THÔNG MINH
SMART CLOCK

SINH VIÊN NGÀNH KỸ THUẬT MÁY TÍNH

TP. HỒ CHÍ MINH, 2021

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH

VIÊN MINH TÂN - 19520928
PHẠM QUỐC ĐĂNG - 19520036
PHẠM TRUNG QUỐC - 19520887
PHẠM TRỌNG HUỖNH - 19521651

ĐỒ ÁN MÔN HỌC THIẾT KẾ HỆ THỐNG NHÚNG
ĐỒNG HỒ THÔNG MINH

SMART CLOCK

SINH VIÊN NGÀNH KỸ THUẬT MÁY TÍNH

GIẢNG VIÊN HƯỚNG DẪN
LÊ HOÀI NGHĨA

TP. HỒ CHÍ MINH, 2021

MỤC LỤC

Chương 1. TỔNG QUAN ĐỀ TÀI.....	1
1.1. Mục tiêu và sơ lược về cách hoạt động của đề tài.....	1
1.2. Phương pháp chung để thực hiện đề tài.....	2
1.3. Các nội dung chính và giới hạn của đề tài đồng hồ thông minh.....	2
1.4. Phân chia thời gian thực hiện các nội dung.....	6
Chương 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG.....	7
2.1. Xác định yêu cầu đề tài.....	7
2.2. Đặc tả sản phẩm.....	8
2.3. Đặc tả kỹ thuật.....	8
2.4. Hệ thống phần cứng.....	9
2.4.1. Cơ sở lý thuyết.....	9
2.4.1.1. Module ESP8266 NodeMCU Lua CP2102.....	9
2.4.1.2. Module matrix LED MAX7219.....	11
2.4.1.3. Cảm biến nhiệt độ, độ ẩm DHT11.....	13
2.4.1.4. Thiết bị phát âm thanh Buzzer.....	14
2.4.1.5. Module LED RGB WS2812B.....	15
2.4.2. Sơ đồ hoạt động của phần cứng.....	16
2.4.3. Chức năng cụ thể.....	16
2.5. Hệ thống phần mềm.....	17
2.5.1. Cơ sở lý thuyết.....	17
2.5.1.1. Phần mềm Arduino IDE.....	17
2.5.1.2. Dịch vụ web trung gian IFTTT.....	18
2.5.1.3. Adafruit IO (MQTT API).....	19
2.5.1.4. Google Assistant.....	20
2.5.2. Sơ đồ hoạt động của phần mềm.....	22
2.5.3. Chức năng cụ thể.....	22
2.5.4. Các hàm chức năng.....	23
2.6. Thiết kế giao diện.....	24

2.6.1. Giao diện của Google Assistant.....	24
2.6.2. Giao diện kết nối WiFi.....	25
Chương 3. HƯỚNG DẪN SỬ DỤNG.....	26
3.1. Hướng dẫn cài đặt ban đầu.....	26
3.2. Hướng dẫn sử dụng.....	26
3.2.1. Tắt/mở thiết bị.....	26
3.2.2. Hiển thị nhiệt độ, độ ẩm.....	26
3.2.3. Thay đổi màu sắc led theo sở thích.....	27
3.2.4. Cài đặt báo thức.....	27
3.3. Một số lỗi thường gặp, lưu ý khi sử dụng.....	27
Chương 4. KẾT QUẢ THỰC NGHIỆM.....	28
4.1. Các nội dung thực nghiệm.....	28
4.2. Kết quả thực nghiệm mong muốn.....	28
4.3. Kết quả thực nghiệm.....	28
4.4. Đánh giá kết quả thực nghiệm.....	28
Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	29
5.1. Kết luận về đề tài.....	29
5.2. Ưu điểm và nhược điểm.....	29
5.3. Hướng phát triển của đề tài.....	30
5.4. Mức độ đóng góp của các thành viên trong nhóm.....	30
TÀI LIỆU THAM KHẢO.....	31

DANH MỤC HÌNH VẼ

Hình 1.1	Sơ đồ khối của đề tài đồng hồ thông minh.....	1
Hình 2.1	Kit RF thu phát ESP8266 Node MCU Lua CP2102.....	9
Hình 2.2	Một số thành phần và chức năng cơ bản trên ESP8266 NodeMCU. .	9
Hình 2.3	Sơ đồ các GPIO cơ bản trên ESP8266 NodeMCU.....	10
Hình 2.4	Matrix LED MAX7219.....	12
Hình 2.5	Cảm biến nhiệt độ, độ ẩm DHT11.....	13
Hình 2.6	Buzzer.....	14
Hình 2.7	IC LED RGB WS2812.....	15
Hình 2.8	Sơ đồ hoạt động của phân cứng.....	16
Hình 2.9	Phần mềm Arduino IDE.....	17
Hình 2.10	Giao diện chính của trang web IFTTT.....	18
Hình 2.11	Giao diện chính của web Adafruit IO.....	20
Hình 2.12	Giao diện tương tác của Google Assistant.....	21
Hình 2.13	Sơ đồ hoạt động của các hàm chức năng.....	24
Hình 2.14	Sơ đồ thiết kế giao diện của đề tài.....	24
Hình 2.15	Giao diện kết nối WiFi cho đồng hồ.....	25
Hình 2.16	Giao diện của Google Assistant khi sử dụng.....	24

DANH MỤC BẢNG

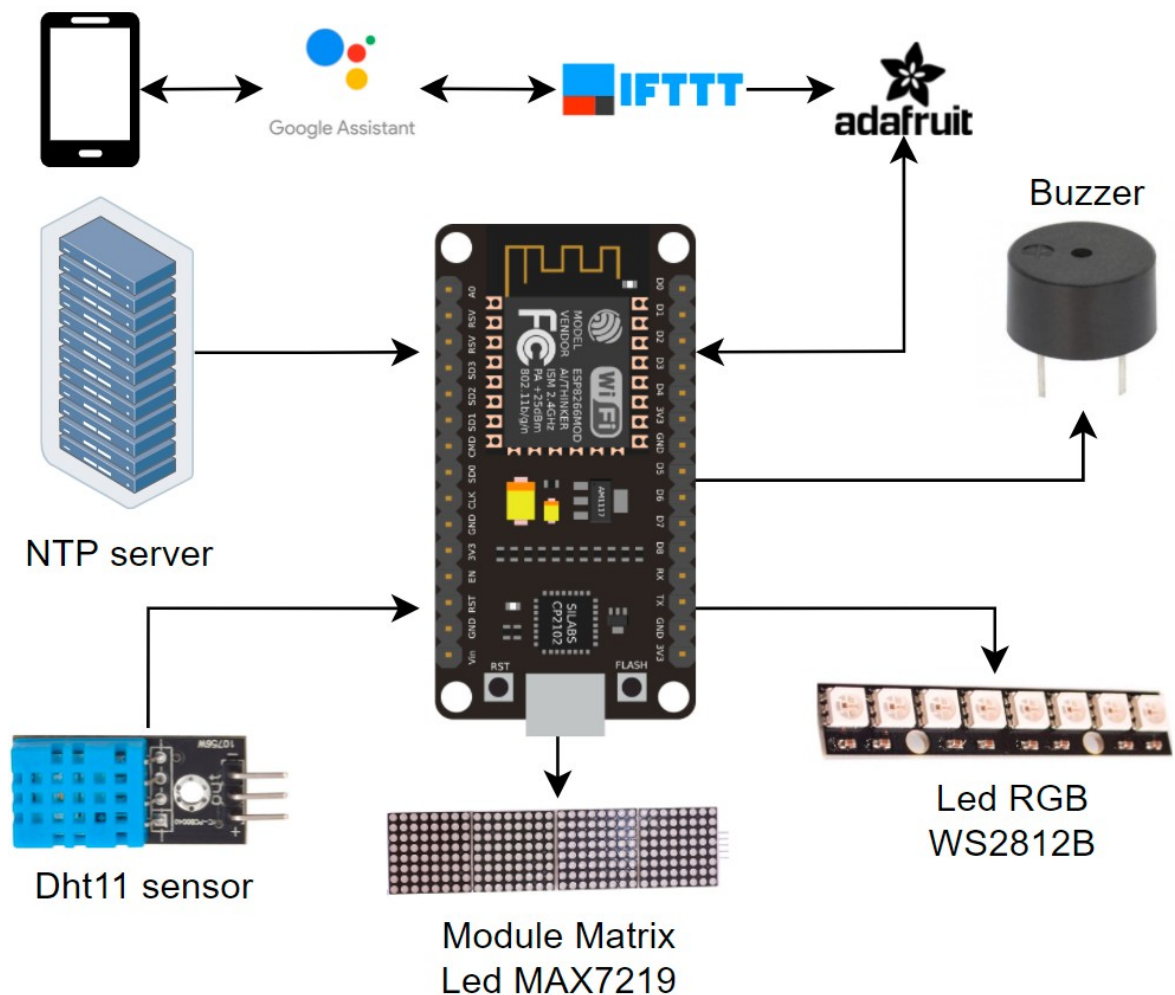
Bảng 1.1	Danh sách các “user story” và các nhiệm vụ cần làm.....	4
Bảng 1.2	Bảng phân chia thời gian thực hiện các công việc.....	6
Bảng 2.1	Bảng mô tả các yêu cầu của đề tài.....	7
Bảng 2.2	Mô tả chức năng các chân trên module.....	11
Bảng 2.3	Sơ đồ chân kết nối giữa Matrix LED và Module ESP.....	13
Bảng 2.4	Sơ đồ chân kết nối giữa DHT11 và Module.....	14
Bảng 2.5	Sơ đồ chân kết nối giữa Buzzer và ESP8266.....	14
Bảng 2.6	Chức năng các chân của IC WS2812.....	15
Bảng 5.1	Ưu điểm và nhược điểm của đề tài.....	29
Bảng 5.2	Mức độ đóng góp của các thành viên.....	30

Chương 1. TỔNG QUAN ĐỀ TÀI

1.1. Mục tiêu và sơ lược về cách hoạt động của đề tài

Hiện tại trên thị trường công nghệ có rất nhiều loại đồng hồ thông minh với nhiều chức năng đa dạng và mẫu mã đẹp. Nhận thấy đề tài đồng hồ thông minh là một đề tài phù hợp và có thể thực hiện được nên nhóm đã quyết định chọn đề tài là thiết kế một chiếc đồng hồ thông minh với các chức năng cơ bản như “hiện thời gian thực, thông báo nhiệt độ, độ ẩm, bật và tắt, hẹn giờ, điều chỉnh màu sắc trang trí, hiển thị màu sắc theo nhiệt độ phòng; tất cả đều sẽ được thực hiện thông qua thao tác đơn giản bằng giọng nói với công cụ Google Assistant”.

Sơ đồ khối của đề tài



Hình 1.1 Sơ đồ khối của đề tài đồng hồ thông minh

Mô tả sơ lược về cách hoạt động của đề tài: Như đã được liệt kê về các chức năng cơ bản của đồng hồ ở trên, đồng hồ thông minh sẽ hiển thị thời gian thực hoặc

nhiệt độ, độ ẩm trên một ma trận led 8x32. Việc hẹn giờ, điều chỉnh màu sắc trang trí, chuyển đổi chế độ từ hiện thị giờ sang hiển thị nhiệt độ, độ ẩm sẽ được thực hiện thông qua công cụ Google Assistant, người dùng sẽ nhập cú pháp yêu cầu về chức năng (các cú pháp và hướng dẫn sử dụng sẽ được cung cấp cùng với sản phẩm) và Google Assistant sẽ phản hồi đã nhận được thông tin. Sau đó thông tin được nhận sẽ qua các bước xử lý trung gian để gửi lại thông tin yêu cầu đồng hồ thực hiện. Đối với chế độ hẹn giờ, khi đến giờ hẹn thì đồng hồ sẽ phát lên tiếng cảnh báo trong vòng 15s.

1.2. Phương pháp chung để thực hiện đề tài

Cần phải xác định rõ nội dung đề tài hướng đến là gì, xác định đối tượng mà đề tài hướng đến, nhiệm vụ nghiên cứu và phân tích cũng như phạm vi nghiên cứu.

Chia nhỏ công việc thành các giai đoạn và các nội dung cần nghiên cứu, như giai đoạn chuẩn bị, giai đoạn tìm hiểu nghiên cứu, giai đoạn hiện thực cơ bản, ghép nối và chạy thử, giai đoạn hoàn thiện và in mạch, đóng gói.

Để đạt được mục tiêu trên, ta cần chia nhỏ nội dung nghiên cứu thực hiện thành các phần bao gồm phần cứng, chương trình (phần lập trình), phần dữ liệu, và phần tương tác với hệ thống, mỗi phần sẽ là một giai đoạn trong việc làm đề tài và đòi hỏi phải nghiêm túc nghiên cứu và thực hiện.

Nghiên cứu đề tài từ các nguồn tài liệu trực tuyến, các trang tìm kiếm lớn, có thể tìm kiếm thêm sự trợ giúp của giảng viên môn học, giảng viên hướng dẫn đồ án, sách,... thêm nữa là sự hỗ trợ và trao đổi giữa các thành viên trong nhóm với nhau.

Chuẩn bị sẵn các phần cứng cũng như các thiết bị đáp ứng cho nhu cầu làm việc và thực hiện đề tài. Đối với các sản phẩm phần cứng cần tìm hiểu kĩ datasheet, nắm rõ cách hoạt động để dễ dàng hơn trong việc kết nối và thực hiện đề tài.

1.3. Các nội dung chính và giới hạn của đề tài đồng hồ thông minh

Đối với việc thực hiện một đề tài thì một trong những bước cơ bản và quan trọng nhất đó chính là phân chia nội dung thực hiện cho đề tài. Đề tài trên sẽ được chia nhỏ thành các nội dung chi tiết dưới đây.

- Tìm hiểu về các mẫu đồng hồ thông minh đã có mặt trên thị trường hiện tại, liệt kê danh sách các chức năng có thể phù hợp với đề tài đã chọn, tìm hiểu về cách để hiện thực các chức năng đã liệt kê.
- Chọn một vi điều khiển phù hợp với đề tài đã chọn (ở đây nhóm đã sử dụng vi điều khiển ESP8266 sử dụng platform Arduino), đọc các tài liệu hướng dẫn kết nối và sử dụng phần cứng như datasheet, cách sử dụng Arduino lập trình cho ESP8266, ...
- Sau khi chọn và hiểu được cách hoạt động, sử dụng của vi điều khiển, tiến hành xác định các linh kiện, ngoại vi cần thiết để thiết kế đề tài, cũng như cách sử dụng và tương tác giữa chúng với vi điều khiển, tất cả được liệt kê trong phần cơ sở lý thuyết.
- Kiểm tra tính chính xác trong hoạt động của các linh kiện như kết nối và sử dụng, tương tác giữa các phần cứng với nhau thông qua những mã nguồn có sẵn, sau đó áp dụng và chỉnh sửa để chúng ta có một mã nguồn hoàn chỉnh để nạp vào vi điều khiển.
- Nạp mã nguồn của toàn bộ chương trình vào vi điều khiển và bắt đầu kiểm tra xem hệ thống đã thực hiện đúng các chức năng đã được đề ra hay chưa, qua đó có những chỉnh sửa nhanh chóng và cải thiện độ hiệu quả cho thiết bị, đặc biệt cần kiểm tra kỹ xem liệu các chức năng có xung đột gì hay không, đảm bảo sản phẩm thể hiện các chức năng tốt nhất.
- Bước cuối cùng, sau khi đã kiểm tra hệ thống hoạt động hiệu quả và không xảy ra lỗi gì, tiến hành đóng gói sản phẩm một cách hoàn chỉnh, chỉnh chu về mẫu mã, đưa ra các thông tin cần thiết cho người sử dụng như hướng dẫn sử dụng.

Một số giới hạn của đề tài như: các chức năng còn chưa được đa dạng, người dùng muốn sử dụng chính xác các chức năng thì cần phải phát âm đúng cú pháp tiếng Anh của từng chức năng đó, ...

Để việc thực hiện các chức năng dễ điều chỉnh và quản lý hơn, nhóm đã liệt kê một danh sách các “user story” và các nhiệm vụ cần làm như sau:

Bảng 1.1 Danh sách các “user story” và các nhiệm vụ cần làm

No.	I want to	So that	Describes	Tasks
1	Hiển thị thời gian	Biết được thời gian hiện tại, phục vụ cho nhu cầu cơ bản (quản lý thời gian học tập, làm việc, ...) của người dùng	Hiển thị thời gian hiện tại trên ma trận LED	Lấy thời gian thực từ server NTP về esp8266 (1.1)
				Hiển thị thời gian lên ma trận LED (1.2)
2	Biết được nhiệt độ, độ ẩm hiện tại	Theo dõi và kiểm soát điều kiện phòng từ đó đưa ra các giải pháp để cải thiện môi trường sống	Sử dụng cảm biến nhiệt độ, độ ẩm để lấy thông số nhiệt, độ ẩm của phòng	Kết nối cảm biến với esp8266(2.1)
				Hiển thị thông số nhiệt, độ ẩm ra ma trận LED (2.2)
3	Màu sắc của LED trang trí thay đổi theo nhiệt độ phòng	Phục vụ mục đích chính là trang trí; thuận tiện cho việc theo dõi thông số môi trường (nhiệt, độ ẩm) của người dùng	Xử lý các thông số nhận được từ cảm biến nhiệt độ, độ ẩm; từ đó điều chỉnh màu sắc của các LED trang trí phù hợp	Lấy thông số nhiệt độ từ cảm biến (2.1); xử lý theo từng trường hợp (3.1)
				Điều chỉnh màu LED RGB theo các trường hợp ở (3.1)
4	Điều chỉnh màu	Người dùng có thể điều chỉnh màu sắc của các LED trang trí	Dùng Google Assistant trên smartphone để thực	Dùng IFTTT để kiểm tra việc thực thi các lệnh được gửi từ Google Assitant (4.1)

	sắc của LED trang trí theo sở thích	theo sở thích cá nhân thông qua smartphone	hiện việc thay đổi màu sắc của LED trang trí	(4.1) ra lệnh cho cập nhật dữ liệu lên Server Adafruit (4.2)
				Esp8266 cập nhật dữ liệu từ server Adafruit (4.3)
				Xử lý dữ liệu được lấy từ (4.3) để thực hiện việc điều chỉnh màu sắc của LED trang trí (4.4)
5	Cài đặt báo thức (bằng giọng nói)	Giúp người sử dụng chủ động về thời gian, tránh trễ giờ trong các hoạt động	Dùng Google Assistant trên smartphone để thực hiện việc cài đặt báo thức cho đồng hồ	(4.1), (4.2), (4.3)
				Kết nối Esp8266 với loa (5.3)
				Xử lý dữ liệu từ (4.3), điều khiển loa phát khi đến thời gian đã được cài đặt
6	Bật / tắt đồng hồ (bằng giọng nói)	Khi không cần sử dụng, người dùng có thể tắt đồng hồ thông qua giọng nói.	Dùng Google Assistant trên smartphone để bật, tắt đồng hồ	(4.1), (4.2), (4.3)
				Xử lý dữ liệu từ (4.3) theo các trường hợp (6.2)
				Thực hiện bật, tắt ma trận LED; ngắt kết nối wifi theo các trường hợp ở (6.2)

1.4. Phân chia thời gian thực hiện các nội dung

Bảng 1.2 Bảng phân chia thời gian thực hiện các công việc

Thời gian	Công việc	Người thực hiện
Tuần 1 – 2	- Chọn đề tài	- Cả nhóm
Tuần 3 – 4 (27/9 - 10/10)	- Xây dựng user story	- Cả nhóm
Tuần 5 – 6 (11/10 -24/10)	- Tìm hiểu về Esp8266, DHT11 - Hoàn thành user story 1, 2	- Huỳnh, Quốc - Tân, Đăng
Tuần 7 – 8 (25/10 - 7/11)	- Tìm hiểu về IFTTT, Adafruit, Google Assistant	- Huỳnh, Quốc
	- Hoàn thành user story 6	- Tân, Đăng
Tuần 9 – 10 (8/11 - 21/11)	- Hoàn thành user story 3, 4, 5	- Cả nhóm
Tuần 11 – 12 (22/11 - 13/12)	- Kiểm tra, hoàn thiện sản phẩm	- Cả nhóm

Nhóm đã quyết định việc chọn đề tài thông qua bỏ phiếu và chọn được đề tài là đồng hồ thông minh. Giai đoạn này tốn cũng khá nhiều thời gian vì chưa có sự nhất quán của các thành viên và thiết bị, linh kiện dùng cho việc thực hiện các đề tài còn hạn chế nên phải tìm một đề tài phù hợp nhất cả về điều kiện lẫn chuyên môn.

Bước tiếp theo của việc xây dựng đề tài đó là xây dựng một danh sách các “user story”, các chức năng mà mình mong muốn, ứng dụng của nó, mô tả các chức năng, các nhiệm vụ để xây dựng được chức năng đó. Đây được xem như bước quan trọng nhất vì nó sẽ định hình cách xây dựng một đề tài theo đúng các chức năng mà chúng ta đặt ra, các công việc cần làm để hoàn thiện đề tài một cách có hiệu quả và cụ thể.

Các tuần còn lại sẽ dựa vào các “user story” để nghiên cứu về lý thuyết cần sử dụng, hiện thực từng chức năng, ...

Bước cuối cùng là kiểm tra chức năng, tính chính xác của từng chức năng cũng như hoàn thiện sản phẩm, cụ thể là mẫu mã sản phẩm.

Chương 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Xác định yêu cầu đề tài

Bảng 2.3 Bảng mô tả các yêu cầu của đề tài

No.	I want to	So that	Describes
1	Hiển thị thời gian thực	Biết được thời gian hiện tại, phục vụ cho nhu cầu cơ bản (quản lý thời gian học tập, làm việc, ...) của người dùng	Hiển thị thời gian hiện tại trên ma trận LED
2	Biết được nhiệt độ, độ ẩm hiện tại	Theo dõi và kiểm soát điều kiện phòng từ đó đưa ra các giải pháp để cải thiện môi trường sống	Sử dụng cảm biến nhiệt độ, độ ẩm để lấy thông số nhiệt, độ ẩm của phòng
3	Màu sắc của LED trang trí thay đổi theo nhiệt độ phòng	Phục vụ mục đích chính là trang trí; thuận tiện cho việc theo dõi thông số môi trường (nhiệt, độ ẩm) của người dùng	Xử lý các thông số nhận được từ cảm biến nhiệt độ, độ ẩm; từ đó điều chỉnh màu sắc của các LED trang trí phù hợp
4	Điều chỉnh màu sắc của LED trang trí theo sở thích	Người dùng có thể điều chỉnh màu sắc của các LED trang trí theo sở thích cá nhân thông qua smartphone	Dùng Google Assistant trên smartphone để thực hiện việc thay đổi màu sắc của LED trang trí.
5	Cài đặt báo thức bằng giọng nói	Giúp người sử dụng chủ động về thời gian, tránh trễ giờ trong các hoạt động	Dùng Google Assistant trên smartphone để thực hiện việc cài đặt báo thức cho đồng hồ
6	Bật / tắt đồng hồ (bằng giọng nói)	Khi không cần sử dụng, người dùng có thể tắt đồng hồ thông qua giọng nói.	Dùng Google Assistant trên smartphone để bật, tắt đồng hồ

2.2. Đặc tả sản phẩm

- Chức năng: Đồng hồ hiển thị thời gian thực, cài đặt báo thức, điều khiển màu sắc led, bật tắt đồng hồ bằng giọng nói.
- Input: cảm biến nhiệt độ, độ ẩm, câu lệnh điều khiển (thông qua Google Assistant), thời gian thực lấy từ NTP server.
- Output: hiển thị thời gian thực, nhiệt độ độ ẩm, màu led, âm báo (buzzer)
- Giao diện người dùng: trang chủ đăng nhập WiFi, led Matrix8x32, Google Assistant.
- Chuẩn giao tiếp: SPI (led Matrix); Single-wire, TTL (LED RGB, DHT11); MQTT (Adafruit)

2.3. Đặc tả kỹ thuật

Phần cứng: ESP8266 NodeMCU Lua CP2102, cảm biến DHT11, Led Matrix 8x32, Buzzer, LED RGB WS2821B.

Yêu cầu phần cứng và phần mềm:

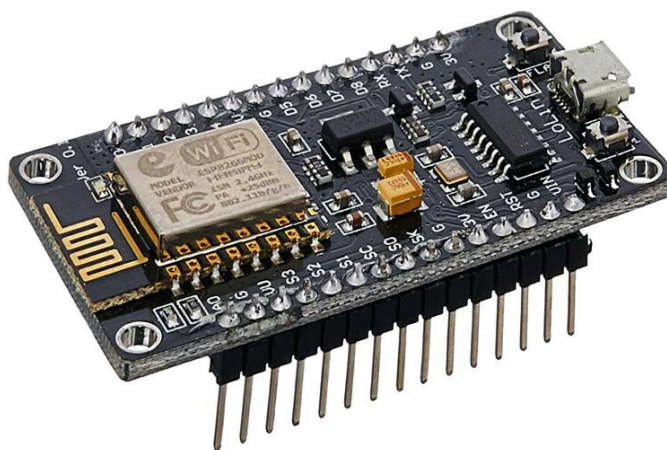
- Xử lý thời gian thực
- Hiển thị giá trị nhiệt độ độ ẩm
- Điều chỉnh led theo nhiệt độ, hoặc theo yêu cầu người dùng
- Cài đặt báo thức
- Điều khiển bằng giọng nói thông qua Google Assistant.

2.4. Hệ thống phần cứng

2.4.1. Cơ sở lý thuyết

2.4.1.1. Module ESP8266 NodeMCU Lua CP2102

Tổng quan



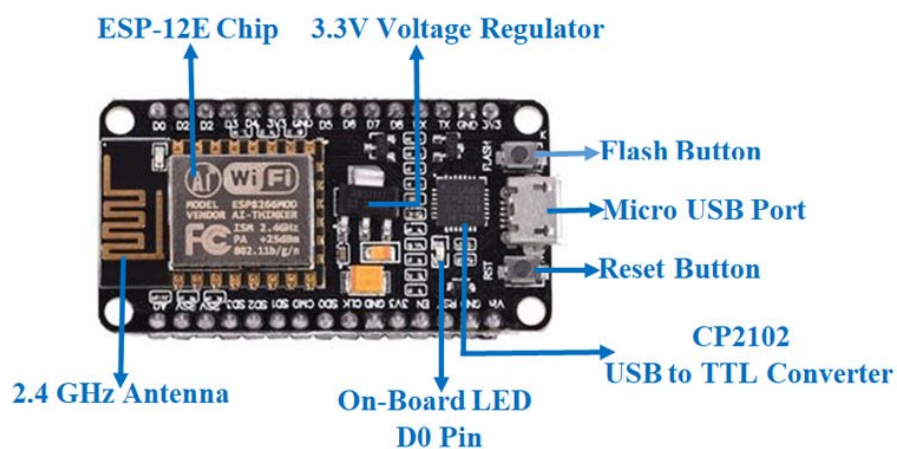
Hình 2.2 Kit RF thu phát ESP8266 Node MCU Lua CP2102

Kit RF thu phát ESP8266 Node MCU Lua CP2102 là kit phát triển dựa trên nền chip WiFi SoC ESP8266.

Sử dụng chip nạp CP2102 giao tiếp thông qua UART, có khả năng nhận Driver trên các hệ điều hành Windows và Linux, là bản nâng cấp của IC nạp CH340.

Sử dụng trực tiếp trình biên dịch Arduino IDE để lập trình và nạp code.

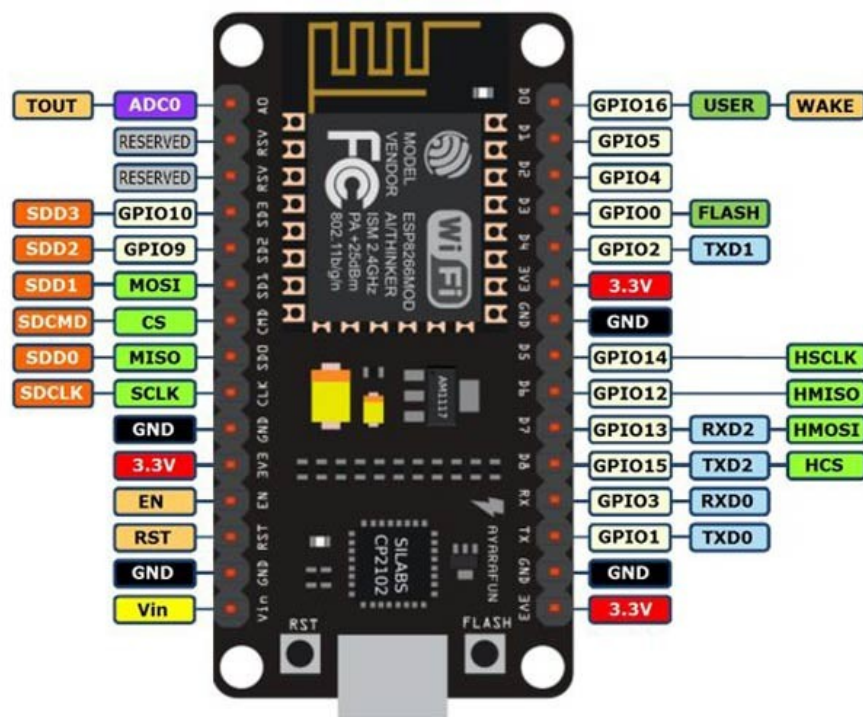
Thông số kỹ thuật



Hình 2.3 Một số thành phần và chức năng cơ bản trên ESP8266 NodeMCU

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- IC nạp: CP2102
- Điện áp đầu vào: 7-12V DC
- 16 GPIO Pins
- GPIO giao tiếp mức 3.3V
- Tương thích chuẩn WiFi: 802.11 b/g/n
- Hỗ trợ: Wi-Fi Direct (P2P), soft-AP
- Các chuẩn giao tiếp:
 - UARTs (x1)
 - SPIs (x1)
 - I2Cs (x1)
- Flash Memory: 4MB
- Kích thước: 25 x 50mm

Sơ đồ chân



Hình 2.4 Sơ đồ các GPIO cơ bản trên ESP8266 NodeMCU

Bảng 2.4 Mô tả chức năng các chân trên module

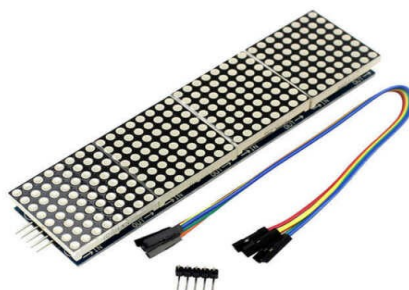
Pin	Tên	Mô tả chức năng
Power	Micro-USB, 3.3V, Vin, GND	Micro-USB: dùng để nạp chương trình cho module / cấp nguồn. 3.3V: đầu ra mặc định là 3.3V, có thể dùng để cấp nguồn cho thiết bị khác Vin: Cấp nguồn cho module GND: nối đất
Control Pins	EN, RST	Chân và nút nhấn để khởi động lại module
Analog Pin	A0	Dùng để nhận biết điện áp tương tự trong khoảng 0-3.3V
GPIOs	GPIO1 – GPIO16	16 chân vào / ra trên module
SPI Pins	SD1, CMD, SDD0 - 3, CLK	4 chân sử dụng cho kết nối thông qua giao thức SPI
UART Pins	TXD0, RXD0, TXD2, RXD2	Sử dụng cho kết nối thông qua UART, UART1 (TXD1, SDD1) dùng cho việc nạp chương trình
I2C Pins	GPIO14, GPIO2	Sử dụng cho giao tiếp thông qua I2C

2.4.1.2. Module matrix LED MAX7219**Tổng quan về Module matrix LED MAX7219**

Module là tổ hợp 4 Led Matrix 8x8, với mỗi led Matrix được điều khiển bởi một IC MAX7219 (tương tự như IC 74HC595 nhưng chuyên dùng cho các ứng dụng Led Driver.).

4 IC MAX7219 được mắc nối tiếp và giao tiếp với nhau bằng chuẩn SPI giúp tiết kiệm số chân cho vi điều khiển (chỉ sử dụng 3 chân để điều khiển).

Ngoài ra IC MAX7219 còn được tích hợp Static RAM 8x8 từ đó việc hiển thị trở nên mượt mà hơn, đồng thời Module còn có thể mở rộng thêm bằng việc mắc nối tiếp các mạch Led Matrix 8x8 MAX7219 lại với nhau.



Hình 2.5 Matrix LED MAX7219

Thông số kỹ thuật:

- 4 Led Matrix 8x8 (Cathode chung), 4 IC MAX7219
- IC giao tiếp: MAX7219

Chuẩn giao tiếp: SPI

- Điện áp sử dụng: 5VDC
- Kích thước (cm): 6.3 x 3.5 x 1.5

Mạch hiển thị 4 Led ma trận MAX7219 có ngõ vào tín hiệu nối tiếp, ghi dịch chuyển tín hiệu nên chỉ cần 3 GPIO để điều khiển, mạch có khả năng nối tiếp thêm nhiều mạch bằng cổng nối tiếp nên dễ dàng để mở rộng.

Sơ đồ chân:

- DIN: là chân truyền dữ liệu từ vi điều khiển đến Max7219 (chân MOSI của vi điều khiển).
- CLK: là chân cấp xung nhịp hoạt động cho Max7219 được nối với chân SCK của vi điều khiển.
- CS: là chân lựa chọn hoạt động, chân này nối với bất kì một chân I/O nào của vi điều khiển.
- VCC: là chân cấp nguồn: sử dụng nguồn 5V / 3.3V DC.
- GND: kết nối với chân GND của vi điều khiển.

⇒ Kết nối với ESP8266

Bảng 2.5 Sơ đồ chân kết nối giữa Matrix LED và Module ESP

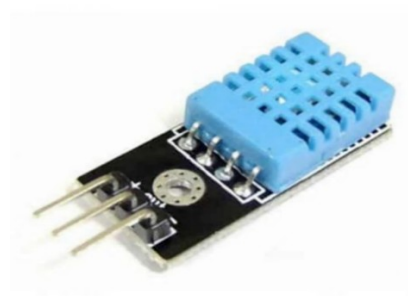
Matrix LED	Esp8266
VCC	3.3V
GND	GND
DIN	D7
CS	D4
CLK	D5

2.4.1.3. Cảm biến nhiệt độ, độ ẩm DHT11

Tổng quan về DHT11:

Cảm biến độ ẩm nhiệt độ DHT11 ra chân được tích hợp sẵn điện trở 5,1k ohm giúp người dùng dễ dàng kết nối và sử dụng hơn so với cảm biến DHT11 chưa ra chân, module lấy dữ liệu thông qua giao tiếp 1 wire (giao tiếp 1 dây).

Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không cần phải qua bất kỳ tính toán nào. Module được thiết kế hoạt động ở mức điện áp 5VDC.



Hình 2.6 Cảm biến nhiệt độ, độ ẩm DHT11

Thông số kỹ thuật:

- Điện áp hoạt động: 5V / 3.3V DC
- Chuẩn giao tiếp: TTL, 1 wire.
- Khoảng đo độ ẩm: 20%-80%RH sai số $\pm 5\%RH$
- Khoảng đo nhiệt độ: 0-50°C sai số $\pm 2^{\circ}C$
- Tần số lấy mẫu tối đa 1Hz (1 giây / lần)
- Kích thước: 28mm x 12mm x 10mm

Kết nối với ESP8266:

Bảng 2.6 Sơ đồ chân kết nối giữa DHT11 và Module

DHT11	Esp8266
VCC	3.3V
GND	GND
OUT	D1

2.4.1.4. Thiết bị phát âm thanh Buzzer

Tổng quan về Buzzer:

Buzzer hay còn gọi là còi chip hoặc còi xung, là thiết bị phát ra âm thanh (tiếng bíp bíp) hay dùng trong các mạch điện tử.

Cấu tạo của Buzzer gồm 2 chân, chân dài là chân (+) và chân ngắn là chân (-). Trong quá trình sử dụng cần chú ý mắc đúng chân để tránh làm hỏng buzzer.

Ứng dụng này sử dụng Buzzer 5V, có khả năng phát ra âm thanh có tần số tối đa 2.5kHz.



Hình 2.7 Buzzer

a) Kết nối với ESP8266:

Bảng 2.7 Sơ đồ chân kết nối giữa Buzzer và ESP8266

Buzzer	Esp8266
(+)	D0
(-)	GND

2.4.1.5. Module LED RGB WS2812B

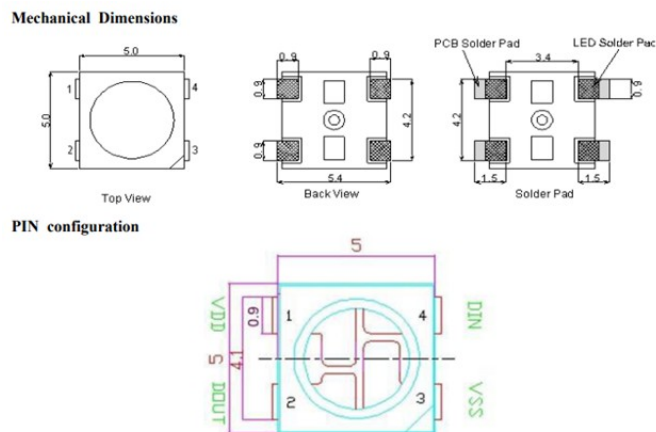
Tổng quan

Module LED RGB WS2812B bao gồm 8 LED RGB WS2812 giao tiếp qua chuẩn giao tiếp 1 wire (chỉ cần 1 chân data duy nhất), mạch rất dễ sử dụng với bộ thư viện hỗ trợ đi kèm với khả năng hiển thị của mỗi Led lên đến 16777216 màu và 256 cấp độ sáng khác nhau.

- IC: LED RGB WS2812B
- Điện áp đầu vào: 3.3V ~ 7V DC (tốt nhất 5V)
- Cỡ LED: SMD 5050
- Số LED: 8

IC LED RGB WS2812B

Là mạch điều khiển LED dùng chung 1 nguồn điện. Được tích hợp chung với 1 LED RGB 5050 để tạo thành 1 điểm (pixel). Chỉ cần dùng 1 chân trên vi điều khiển để điều chỉnh màu sắc của LED.

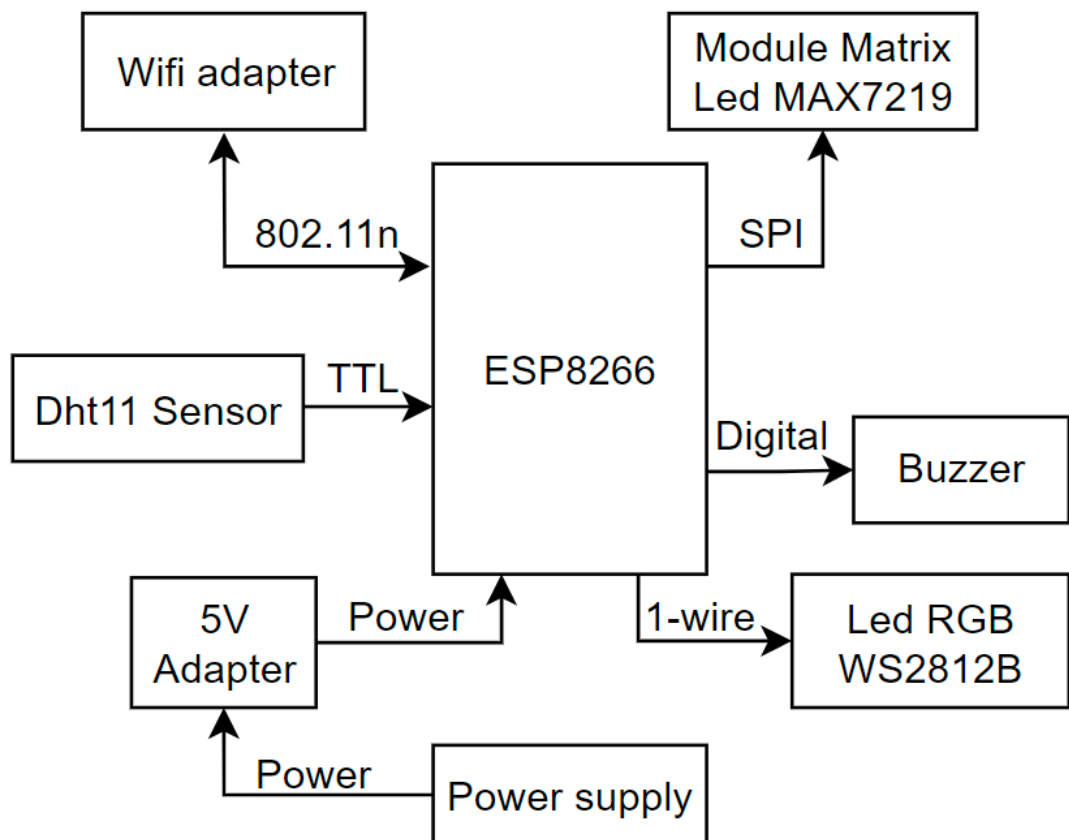


Hình 2.8 IC LED RGB WS2812

Bảng 2.8 Chức năng các chân của IC WS2812

STT	Ký hiệu	Mô tả chức năng
1	VDD	Cấp nguồn cho Module
2	DOUT	Dữ liệu đầu ra
3	VSS	Nối đất
4	DIN	Dữ liệu điều khiển đầu vào

2.4.2. Sơ đồ hoạt động của phần cứng



Hình 2.9 Sơ đồ hoạt động của phần cứng

2.4.3. Chức năng cụ thể

ESP8266 hoạt động với tần số 80MHz, cảm biến nhiệt độ, độ ẩm DHT11, LED Matrix MAX7219 (8x32), buzzer, LED RGB WS2812B.

ESP8266 được kết nối với WiFi (nơi người dùng).

ESP8266 lấy thời gian thực từ server NTP. Hiển thị giá trị thời gian lên LED Matrix; điều chỉnh hiệu ứng của LED RGB theo mô chương trình có sẵn.

ESP8266 liên tục cập nhật giá trị từ server Adafruit để bắt các sự kiện từ người dùng: yêu cầu hiển thị nhiệt độ, độ ẩm; cài đặt báo thức; bật hoặc tắt sản phẩm; đổi màu LED RGB trang trí.

ESP8266 đọc giá trị nhiệt độ từ cảm biến nhiệt độ, độ ẩm DHT11, sau đó điều chỉnh màu LED RGB WS2812B theo như giá trị nhiệt độ đọc được; hiển thị thời gian và nhiệt độ, độ ẩm lên LED Matrix MAX7219.

ESP8266 đọc giá trị báo thức từ server Adafruit, so sánh giá trị báo thức với giá trị thời gian thực; điều khiển buzzer reo khi hai giá trị bằng nhau.

ESP8266 đọc giá trị từ server Adafruit, thực hiện tắt LED Matrix, LED RGB (khi người dùng yêu cầu tắt sản phẩm) hoặc mở theo yêu cầu người dùng.

ESP8266 đọc giá trị từ server Adafruit, thực hiện đổi màu LED RGB trang trí theo thuật toán được cài đặt sẵn.

2.5. Hệ thống phần mềm

2.5.1. Cơ sở lý thuyết

2.5.1.1. Phần mềm Arduino IDE

Tổng quan về Arduino IDE

Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linh kiện. Phần mềm giúp bạn có thể sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.

Cách Arduino IDE hoạt động

Cách Arduino IDE hoạt động là: khi người dùng viết mã và biên dịch, IDE sẽ tạo file Hex cho mã. File Hex là các file thập phân Hexa được Arduino hiểu và gửi đến bo mạch bằng cáp USB. Mỗi bo Arduino đều được tích hợp một bộ vi điều khiển, bộ vi điều khiển sẽ nhận file Hex và chạy theo mã được viết.



Hình 2.10 Phần mềm Arduino IDE

Một số lý do nên sử dụng Arduino IDE:

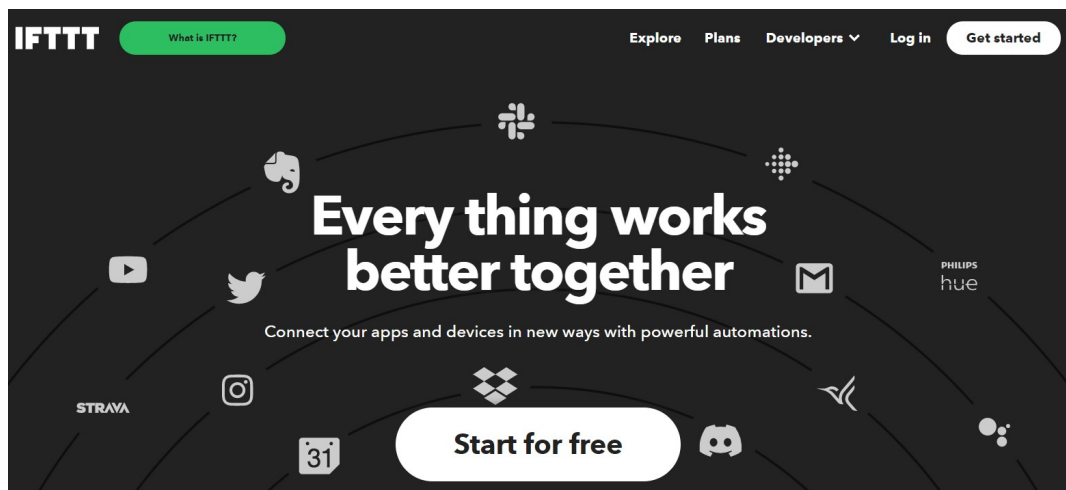
- Arduino IDE là phần mềm lập trình mã nguồn mở miễn phí
- Sử dụng ngôn ngữ lập trình C/C++ thân thiện với các lập trình viên
- Hỗ trợ lập trình tốt cho bo mạch Arduino
- Thư viện hỗ trợ phong phú
- Giao diện đơn giản, dễ sử dụng
- Hỗ trợ đa nền tảng như Windows, MacOS, Linux

2.5.1.2. Dịch vụ web trung gian IFTTT

Tổng quan về IFTTT

IFTTT là một dịch vụ trung gian cho phép người dùng dễ dàng lập trình một phản hồi dựa trên một sự kiện nào đó xảy ra. IFTTT viết đầy đủ là If This, Then That. IFTTT hợp tác với nhiều nền tảng dịch vụ để khi có thông báo về hoạt động, IFTTT sẽ thực hiện phản hồi trên một nền tảng khác đang được kết nối.

IFTTT dùng những Recipes (công thức IFTTT) còn gọi là các Applet để có thể kết nối những ứng dụng với nhau. Khi trên ứng dụng này có sự thay đổi thì IFTTT sẽ thay đổi trên ứng dụng kia.



Hình 2.11 Giao diện chính của trang web IFTTT

Các khái niệm cơ bản liên quan đến IFTTT

Recipe: Khi thực hiện một công việc nào đó trong IFTTT đây là câu lệnh khai báo. Chẳng hạn: bạn tạo ra 1 recipe yêu cầu trên IFTTT khi tôi đăng tải hình ảnh trên Facebook thì tất cả những hình ảnh này sẽ được tự động lưu về Google Drive.

Channel: là từ sử dụng để nhắc đến các dịch vụ mà IFTTT đang hỗ trợ như Gmail, Facebook, Youtube, Linkedin, ... Chỉ cần 1 lần kích hoạt channel thì từ đó trở về sau khi bạn tạo bất kỳ recipe nào liên quan đến channel đã tạo IFTTT sẽ hiểu tự động và làm việc mà không phải kích hoạt lại.

Trigger: Đây là điều kiện để kích hoạt hành động Nếu ... Thì. Có nghĩa là nếu điều kiện A được thỏa mãn thì công việc B sẽ được thực hiện.

Action: Được hiểu là hành động sau khi trigger đã được kích hoạt thành công.

Lợi ích khi dùng IFTTT SEO

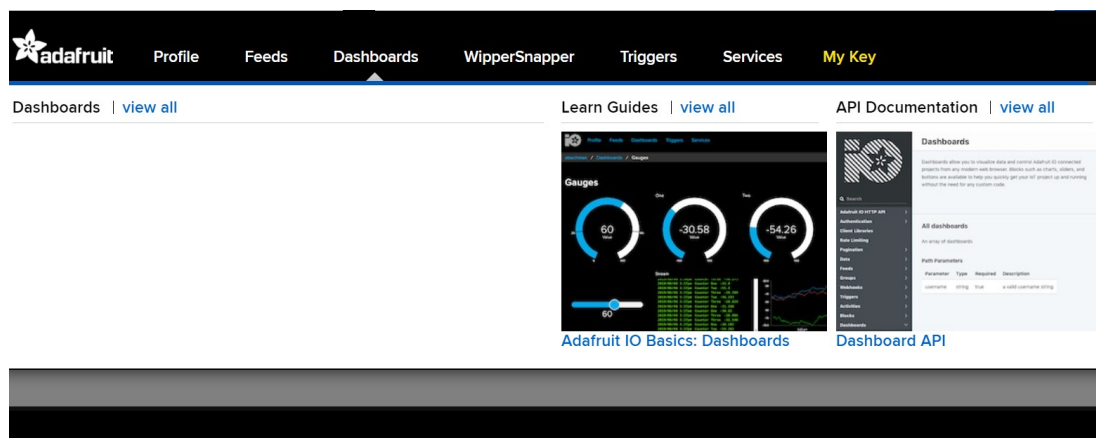
IFTTT giúp tiết kiệm các yếu tố chi phí, thời gian, công sức cho người dùng. Hơn nữa, công cụ này còn có khả năng làm việc mọi lúc, mọi nơi kể cả khi chúng ta bận hay đi ngủ. Nó còn cho chúng ta sử dụng 5 applet miễn phí và rất dễ dàng để cài đặt và sử dụng.

2.5.1.3. Adafruit IO (MQTT API)

Tổng quan về Adafruit IO

Adafruit đưa ra một MQTT miễn phí Cloud Service cho thí nghiệm IOT và người học gọi Adafruit IO trong năm 2015. Adafruit IO là một dịch vụ internet đơn giản dễ sử dụng, dễ dàng cho phép các thiết bị IoT nhận và gửi dữ liệu. Và Adafruit IO có thể làm các công việc sau:

- Hiện thị dữ liệu của bạn trong thời gian thực, trực tuyến
- Làm cho dự án của bạn kết nối internet: Điều khiển động cơ, đọc dữ liệu cảm biến và hơn thế nữa!
- Kết nối các dự án với các dịch vụ web như Twitter, nguồn cấp dữ liệu RSS, dịch vụ thời tiết, v.v.
- Kết nối dự án của bạn với các thiết bị hỗ trợ internet khác.
- Tất cả những điều trên có thể làm miễn phí với Adafruit IO.



Hình 2.12 Giao diện chính của web Adafruit IO

Khái niệm cơ bản trong Adafruit IO

Dashboard: là một bảng điều khiển kỹ thuật số hoặc là giao diện số. Trong bảng này với rất nhiều thông tin đã được thu thập và tổng hợp từ nhiều nguồn dữ liệu khác nhau của tổ chức và doanh nghiệp lớn, nhỏ vào màn hình.

Trigger: Sử dụng kích hoạt trong Adafruit IO để kiểm soát và phản ứng với dữ liệu của bạn. Cấu hình các kích hoạt để gửi email cho bạn khi hệ thống của bạn ngoại tuyến, phản ứng với cảm biến nhiệt độ quá nóng và xuất bản tin nhắn đến nguồn cấp dữ liệu mới.

2.5.1.4. Google Assistant

Tổng quan về Google Assistant

Google Assistant là trợ lý ảo của Google, là sự phát triển của một tính năng trên Android có tên là Google Now. Là đối thủ cạnh tranh với Siri của Apple, Alexa của Amazon và Cortana của Microsoft

Google Assistant ra mắt vào tháng 5 năm 2016, như một phần của ứng dụng nhắn tin thông minh Allo của Google. Ngay sau đó, Google Assistant đã chuyển sang điện thoại Pixel của Google. Hiện tại, Google Assistant có thể được tải xuống miễn phí trên các thiết bị chạy Android 5 lollipop trở lên.



Hình 2.13 Giao diện tương tác của Google Assistant

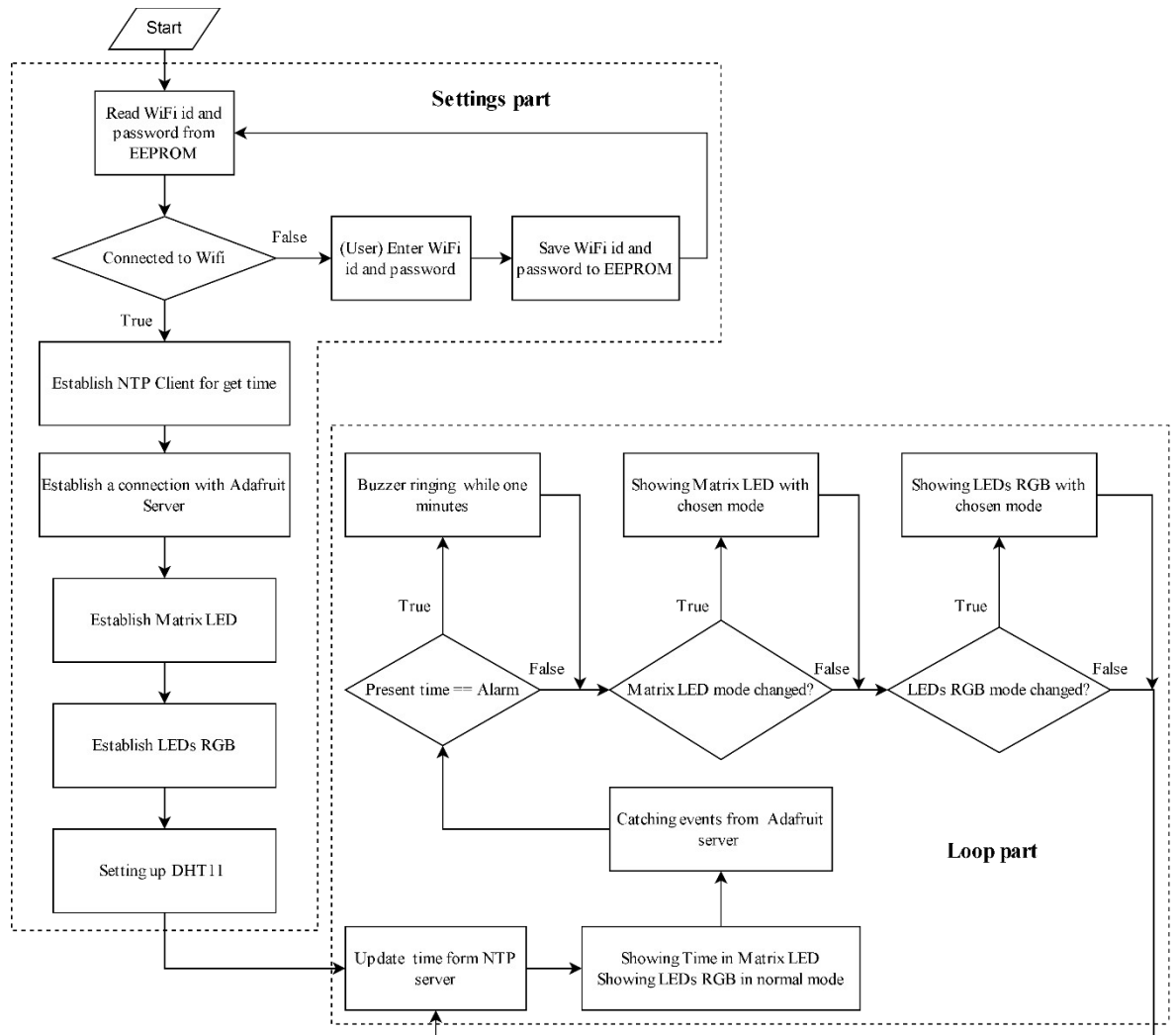
Google Assistant có thể làm gì?

Cung cấp lệnh thoại, tìm kiếm bằng giọng nói và kích hoạt thiết bị bằng giọng nói, cho phép thực hiện một số tác vụ khi người dùng nói từ “Ok Google” hoặc “Hey Google”

- Kiểm soát các thiết bị thông minh
- Truy cập thông tin từ lịch và thông tin cá nhân khác
- Kiểm soát playlist âm nhạc của bạn
- Tìm kiếm thông tin online như đặt chỗ nhà hàng, chỉ đường, thời tiết
- Phát nội dung trên Chromecast hoặc các thiết bị tương thích khác
- Chạy bộ hẹn giờ và lời nhắc
- Lên lịch hẹn và gửi tin nhắn
- Mở ứng dụng trên điện thoại
- Đọc thông báo cho bạn

Google Assistant trong các thiết bị Google Home tạo thành nền tảng của việc điều khiển các ngôi nhà thông minh.

2.5.2. Sơ đồ hoạt động của phần mềm



Hình 2.13: Sơ đồ hoạt động của phần mềm

2.5.3. Chức năng cụ thể

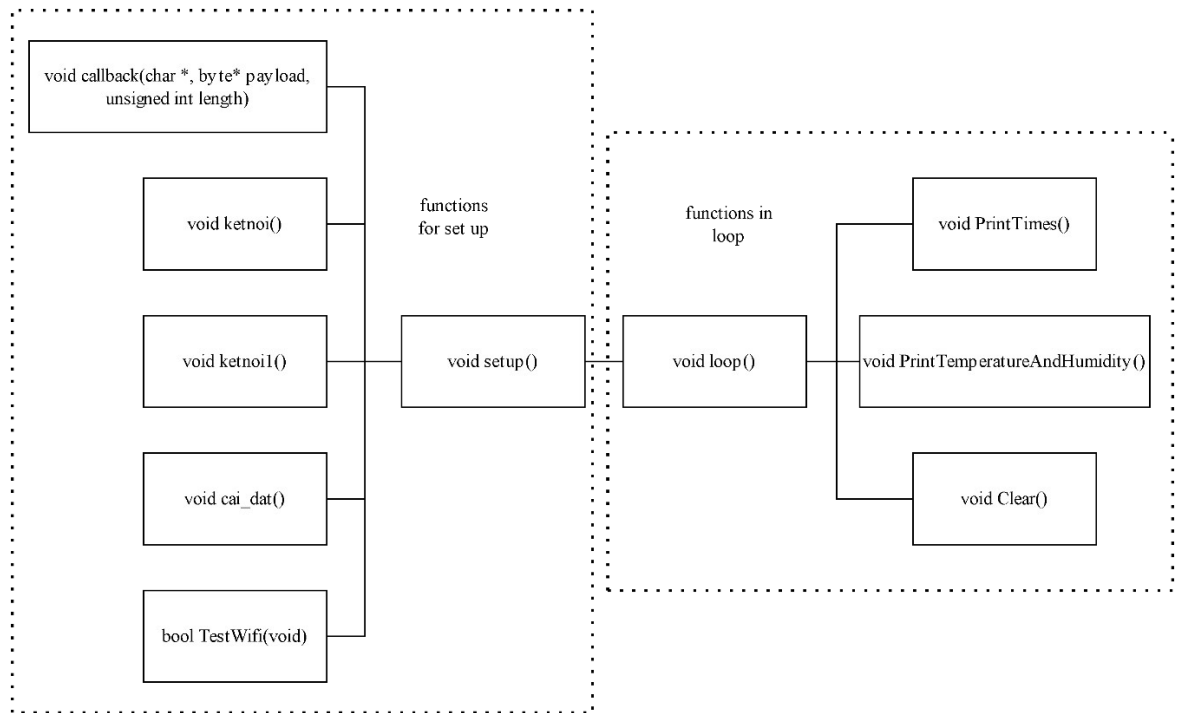
ESP8266 là bộ xử lý trung tâm có chức năng nhận dữ liệu thời gian (từ NTP server), nhận dữ liệu về lệnh của người dùng được gửi qua Google Assistant; điều khiển các thành phần phần cứng khác (LED Matrix, LED RGB, Buzzer) thực hiện các chức năng theo yêu cầu của người sử dụng.

Google Assistant nhận yêu cầu từ người dùng, các Applets trên IFTTT được kích hoạt và gửi dữ liệu đến các feeds trên Server Adafruit.

ESP8266 liên tục bắt sự kiện thay đổi về dữ liệu trên Server Adafruit lấy dữ liệu đó để xử lý; sau đó điều khiển các thành phần phần cứng khác thực hiện chức năng đã được cài đặt sẵn.

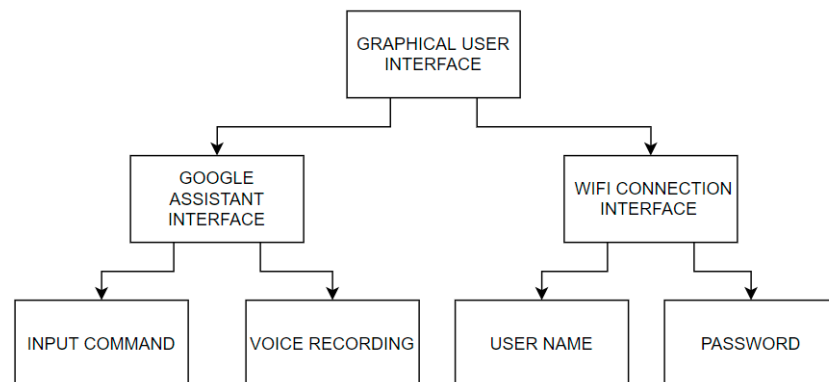
2.5.4. Các hàm chức năng

- void callback (char* topic, byte* payload, unsigned int length): hàm được gọi khi có sự kiện thay đổi dữ liệu xảy ra trên server Adafruit; xử lý dữ liệu được lấy về từ server Adafruit theo các trường hợp (hiển thị nhiệt độ, độ ẩm; cài đặt báo thức; bật, tắt đồng hồ, đổi màu LED RGB).
- String convert(byte* in, unsigned int length): hàm chuyển đổi kiểu byte của dữ liệu được lấy về từ server Adafruit thành kiểu string để thuận tiện cho việc xử lý trong hàm callback.
- void ketnoi(): hàm được gọi khi ESP8266 đang ở chế độ Station (STA)
- void ketnoi1(): hàm được gọi khi ESP8266 đã được kết nối với WiFi của người dùng.
- void cai_dat(): hàm có nhiệm vụ ghi dữ liệu WiFi (ID và mật khẩu) vào EEPROM khi người dùng nhập ID và mật khẩu WiFi trong lần đầu sử dụng.
- bool testWiFi(void): hàm kiểm tra việc kết nối WiFi.
- void setup(): hàm cài đặt cho ESP8266: cài đặt kết nối ESP8266 với server NTP; cài đặt kết nối với LED Matrix MAX7219; cài đặt kết nối với LED RGB WS2812B; cài đặt kết nối với server Adafruit; cài đặt kết nối với Buzzer.
- void loop(): lặp liên tục; thực hiện cập nhật thời gian thực từ server NTP, cập nhật dữ liệu từ server Adafruit và thực hiện các yêu cầu theo dữ liệu đã được xử lý.
- void PrintTimes(): thực hiện hiển thị thời gian thực lên LED Matrix MAX7219.
- void PrintTemperatureAnhHumidity(): thực hiện hiển thị nhiệt độ, độ ẩm lên LED Matrix MAX7219.
- void Clear(): thực hiện xóa các dữ liệu đang hiển thị trên LED Matrix MAX7219



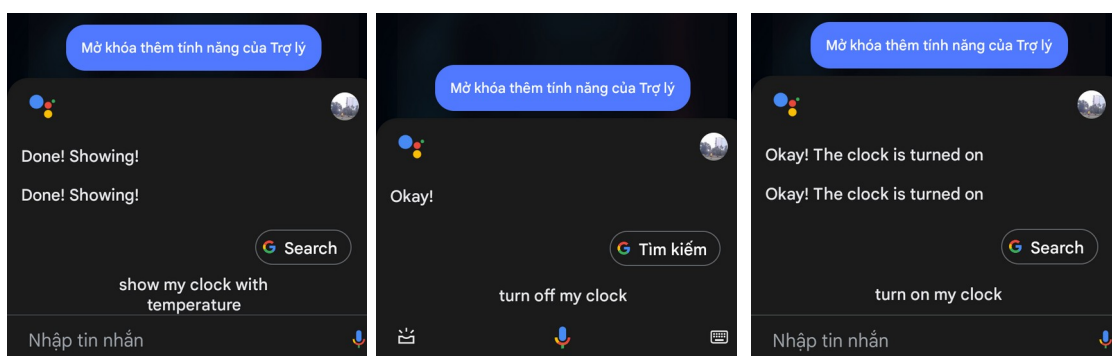
Hình 2.14 Sơ đồ hoạt động của các hàm chức năng

2.6. Thiết kế giao diện



Hình 2.15 Sơ đồ thiết kế giao diện của đề tài

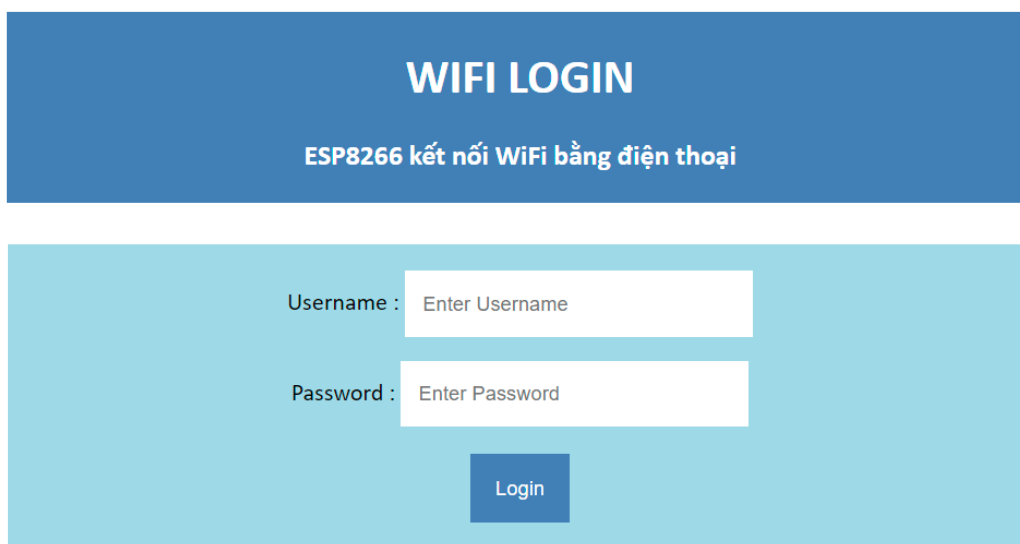
2.6.1. Giao diện của Google Assistant



Hình 2.16 Giao diện của Google Assistant khi sử dụng

Sau khi đã kết nối đồng hồ với wifi cá nhân thành công, để sử dụng được các chức năng khác, cần có thao tác sử dụng giọng nói hoặc ghi tên/cú pháp của chức năng vào dịch vụ Google Assistant. Thông qua giao diện này người dùng có thể tương tác và thay đổi trực tiếp các chức năng của đồng hồ, nếu cú pháp/chức năng hợp lệ, Google Assistant sẽ phản hồi Done hoặc Okay.

2.6.2. Giao diện kết nối WiFi



The image shows a web interface for WiFi login. At the top, a dark blue banner contains the text "WIFI LOGIN" in white, bold letters, and below it, "ESP8266 kết nối WiFi bằng điện thoại" in smaller white text. Below the banner is a light blue rectangular area. Inside this area, there are two input fields: "Username : Enter Username" and "Password : Enter Password". Below these fields is a dark blue button with the text "Login" in white.

Hình 2.17 Giao diện kết nối WiFi cho đồng hồ

Để đến với giao diện này, người dùng sử dụng điện thoại di động kết nối với WiFi có tên Smart Clock. Đây là sóng WiFi do chính ESP8266 phát ra khi được cấp nguồn.

Với giao diện ESP8266 kết nối với WiFi của người sử dụng. Nhập tên WiFi (Username) và mật khẩu WiFi (password), sau đó chọn Login để ESP8266 có thể nhận và gửi dữ liệu thời gian, nhiệt độ, độ ẩm, trạng thái điều khiển thiết bị thông qua các dịch vụ đám mây.

Sau khi kết nối thành công, thiết bị ngay lập tức cập nhật thời gian thực lên màn hình hiển thị.

Chương 3. HƯỚNG DẪN SỬ DỤNG

3.1. Hướng dẫn cài đặt ban đầu

Bước 1: Tiến hành cấp nguồn cho Smart Clock và tải ứng dụng Google Assistant từ Google Play hoặc App Store.

Bước 2: Kết nối Wifi có tên “SmartClock” bằng điện thoại.

Bước 3: Sau khi kết nối, mở trình duyệt, truy cập vào IP “192.168.4.1”, một giao diện đăng nhập hiện ra.

Bước 4: Nhập tài khoản và mật khẩu Wifi mà bạn đang sử dụng để đồng hồ kết nối được với Wifi của bạn.

Bước 5: Kết nối thành công, đồng hồ sẽ cập nhật đúng theo thời gian hiện tại.

3.2. Hướng dẫn sử dụng

Chúng ta sẽ sử dụng thiết bị thông qua Google Assistant vì thế mở Google Assistant hoặc dùng lệnh “Ok Google” hay “Hey Google” để chạy ứng dụng. Các lệnh dưới đây có thể voice hoặc nhập vào thanh tìm kiếm.

3.2.1. Tắt/mở thiết bị

Tắt thiết bị, cú pháp: “*turn off my clock*”. Thiết bị chỉ tắt khi chúng ta đã cấp nguồn. Để mở thiết bị, dùng lệnh: “*turn on my clock*”. Thiết bị chỉ hiển thị giờ và phút. Google Assistant trả về: “*Okay!*” thông báo câu lệnh chính xác và thực hiện yêu cầu.

3.2.2. Hiển thị nhiệt độ, độ ẩm

Cú pháp: “*Show my clock with temperature*”. Google Assistant trả về: “*Done! Showing!*” và ngay lập tức nhiệt độ và độ ẩm trong phòng sẽ lần lượt được hiển thị trên màn hình. Đồng thời 2 dây đèn sẽ thay đổi theo nhiệt độ phòng hoặc bạn có thể tùy chỉnh theo sở thích (được giải thích ở phần sau)

3.2.3. Thay đổi màu sắc led theo sở thích

Cú pháp: “*change my clock led with numbers x*”. Với *x* là chỉ số màu từ 0 đến 765. Google Assistant trả về: “*Done! Showing my color!*” và ngay lập tức màu sắc tương ứng với chỉ số được hiển thị và quay về chế độ xem thời gian.

3.2.4. Cài đặt báo thức

Báo thức trong ngày: “*turn on my smartclock at hh and mm*”. Với *hh* là giờ và *mm* là phút. Ví dụ: “*turn on my smartclock at 20 and 30*”.

Báo thức vào ngày mai: “*turn on my smartclock at hh and mm tomorrow*”. Với *hh*:giờ và *mm*:phút. Ví dụ: “*turn on my smartclock at 20 and 30 tomorrow*”.

Báo thức ngày cụ thể: “*turn on my smartclock at hh and mm dd/mm/yy*”. Với *hh*:giờ, *mm*:phút, *dd*:ngày, *mm*:tháng, *yy*:năm. Ví dụ: “*turn on my smartclock at 20 and 30 24/12/2021*”.

Báo thức vào thứ *x* (*x* từ thứ hai đến thứ chủ nhật) trong tuần: “*turn on my smartclock at hh and mm day of the week*”. Với *hh*:giờ, *mm*:phút, *day of the week*: Saturday, Sunday, ... Ví dụ: “*turn on my smartclock at 20 and 30 thursday*”. (Với lệnh này người dùng có thể cài đặt báo thức cho ngày thứ *x* gần nhất).

Google Assistant trả về: “Done!” để thông báo nhận đã nhận được lệnh cài đặt.

3.3. Một số lỗi thường gặp, lưu ý khi sử dụng

Kiểm tra tài khoản Google trên Google Assistant và tài khoản dùng được cung cấp từ nhà sản xuất phải là một tài khoản.

Sản phẩm có thể có lỗi (không cập nhật được thời gian thực, không kết nối được Google Assistant); cách khắc phục là rút nguồn và cắm lại.

Không để sản phẩm gần nguồn nhiệt cao hoặc tiếp xúc trực tiếp với chất lỏng, tránh va đập mạnh.

Chương 4. KẾT QUẢ THỰC NGHIỆM

4.1. Các nội dung thực nghiệm

- Thực nghiệm thứ nhất: Kết nối wifi cá nhân cho đồng hồ.
- Thực nghiệm thứ hai: Nhận và hiển thị thời gian thực của đồng hồ.
- Thực nghiệm thứ ba: Chuyển sang chế độ hiển thị nhiệt độ, độ ẩm.
- Thực nghiệm thứ tư: Yêu cầu tắt, mở đồng hồ bằng Google Assistant.
- Thực nghiệm thứ năm: Thay đổi màu sắc trang trí theo ý muốn của người dùng hoặc theo nhiệt độ của phòng bằng Google Assistant.
- Thực nghiệm thứ sáu: Hẹn giờ đồng hồ và kiểm tra xem đến đúng giờ thì đồng hồ có báo động không.

4.2. Kết quả thực nghiệm mong muốn

- Thực nghiệm thứ nhất: Đồng hồ kết nối với wifi cá nhân thành công.
- Thực nghiệm thứ hai: Hiển thị chính xác thời gian thực.
- Thực nghiệm thứ ba: Đồng hồ hiển thị chính xác nhiệt độ và độ ẩm của phòng hoặc nơi đặt đồng hồ.
- Thực nghiệm thứ tư: Tắt và mở được đồng hồ bằng Google Assistant.
- Thực nghiệm thứ năm: Thay đổi được màu sắc trang trí theo ý muốn của người dùng hoặc theo nhiệt độ của phòng bằng Google Assistant.
- Thực nghiệm thứ sáu: Hẹn giờ đồng hồ và báo thức đúng giờ đã hẹn.

4.3. Kết quả thực nghiệm

Video về sản phẩm, mô tả việc thực nghiệm theo trình tự và kết quả được đăng ở đường dẫn dưới đây: [Smart Clock - YouTube](#)

4.4. Đánh giá kết quả thực nghiệm

Đối với từng lần thử khác nhau, kết quả thực nghiệm đều cho ra đúng yêu cầu đã đặt ra của đề tài. Tuy nhiên nhóm chỉ mới kiểm tra một số lần nhất định, sau khi báo cáo kết thúc, nhóm sẽ tiếp tục kiểm tra để có thể phát hiện ngoại lệ hoặc sai sót nào đó khi sử dụng sản phẩm và có kế hoạch nâng cấp, sửa chữa cho sản phẩm một cách tốt nhất.

Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận về đề tài

Tính đến thời điểm hiện tại, sản phẩm của nhóm đã được hoàn thiện cả về mẫu mã bên ngoài lẫn chức năng bên trong, đúng với mục đích ban đầu của nhóm đặt ra cho sản phẩm.

Đây là một đề tài không quá mới mẻ và khá dễ tiếp cận cũng như tìm hiểu để thực hiện, tuy nhiên đối với các bạn sinh viên thì đây là một đề tài cần thiết để các bạn làm quen với môn học hiện tại và các môn học phía sau nó.

Đề tài yêu cầu việc tìm hiểu nghiêm túc cùng khối lượng công việc và kiến thức bao quát phù hợp với các bạn sinh viên bắt đầu nghiên cứu về vi điều khiển và giao tiếp với các thiết bị phần cứng khác.

5.2. Ưu điểm và nhược điểm

Bảng 5.9 Ưu điểm và nhược điểm của đề tài

Ưu điểm	Nhược điểm
Dịch vụ như IFTTT, Adafruit, Google Assistant hoàn toàn miễn phí, chỉ cần thiết bị có kết nối internet thì có thể truy cập dễ dàng	Dịch vụ trung gian IFTTT cho phép sử dụng hạn chế số applets (tối đa 5) đối với tài khoản miễn phí
Giá thành hợp lý đối với một sản phẩm công nghệ với các tính năng như trên (cụ thể là xấp xỉ 300k)	Bắt buộc phải phát âm đúng để nhận dữ liệu
Thiết bị nhỏ gọn, dễ dàng sử dụng đối với nhiều đối tượng khác nhau	Để đảm bảo dữ liệu được nhận chính xác thì người dùng vẫn phải nhập text
	Hiện tại số lượng chức năng còn hạn chế, khả năng tương tác chưa được cao

5.3. Hướng phát triển của đề tài

Sau khi đã đánh giá được các ưu điểm và nhược điểm của sản phẩm, nhóm đưa ra hướng phát triển cho đề tài trong tương lai như sau:

- Điều khiển thiết bị thông qua Google Assistant tiếng Việt, thêm các ngoại lệ hoặc từ khóa giúp việc sử dụng thuận tiện hơn.
- Bổ sung một số tiện ích như thêm nội dung cho báo thức, xác nhận thời gian đã cài đặt, chọn múi giờ để hiển thị, tương tác linh hoạt hơn với người dùng.
- Sử dụng ESP-IDF framework để can thiệp sâu hơn vào hệ thống.
- Ứng dụng trên điện thoại và website điều khiển và quản lý thiết bị.
- Có khả năng cập nhật được các phiên bản phần mềm từ xa.
- Tạo kết nối với nhiều server NTP để đảm bảo việc lấy dữ liệu thời gian thực.

5.4. Mức độ đóng góp của các thành viên trong nhóm

Bảng dưới đây được đánh giá dựa vào mức độ đóng góp cho đề tài của từng thành viên:

Bảng 5.10 Mức độ đóng góp của các thành viên

Tên thành viên	Mã số sinh viên	Mức độ đóng góp
Viên Minh Tân	19520928	25%
Phạm Quốc Đăng	19520036	25%
Phạm Trung Quốc	19520887	25%
Phạm Trọng Huỳnh	19521651	25%

TÀI LIỆU THAM KHẢO

- ["Randomnerd," [Online]. Available:
1] <https://randomnerdtutorials.com/getting-started-with-esp8266-wifi-transceiver-review/>. [Accessed 29 12 21].
- ["Last Minute Engineers," [Online]. Available:
2] <https://lastminuteengineers.com/esp8266-ntp-server-date-time-tutorial/>. [Accessed 29 12 21].
- ["Components101," 22 April 2020. [Online]. Available:
3] <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>. [Accessed 29 12 2021].
- ["Components101," 1 June 2015. [Online]. Available:
4] <https://lastminuteengineers.com/esp8266-ntp-server-date-time-tutorial/>. [Accessed 29 12 21].
- [Arduinokit, "ARDUINO KIT," 27 12 2020. [Online]. Available:
5] https://arduinokit.vn/esp8266-adafruit-ifttt-google-assistant-tieng-viet/?fbclid=IwAR1-5_jMFtiadB3LfqKOORJUU0rGORvMCwbFvZzG_gvtY9jn9q-lZbG00JM. [Accessed 29 12 21].