

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN NHẬT HUY

**XÂY DỰNG MÔ HÌNH NHÀ THÔNG MINH HƯỚNG
ĐẾN HỆ SINH THÁI THÀNH PHỐ THÔNG MINH**

Chuyên ngành : KỸ THUẬT VIỄN THÔNG

Mã số : 8.52.02.08

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2021

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN NHẬT HUY

**XÂY DỰNG MÔ HÌNH NHÀ THÔNG MINH HƯỚNG
ĐẾN HỆ SINH THÁI THÀNH PHỐ THÔNG MINH**

Chuyên ngành : KỸ THUẬT VIỄN THÔNG

Mã số : 8.52.02.08

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. NGUYỄN VIỆT HÙNG

HÀ NỘI - 2021

HÀ NỘI - 2021

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn qua phần mềm DoIT một cách trung thực và đạt kết quả độ tương đồng% toàn bộ nội dung luận văn. Bản luận văn kiểm tra qua phần mềm là bản cứng luận văn đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

....., ngày....tháng....năm.....

HỌC VIÊN CAO HỌC

Nguyễn Nhật Huy

MỤC LỤC

BẢN CAM ĐOAN.....	i
MỤC LỤC.....	ii
DANH MỤC BẢNG.....	1
DANH MỤC HÌNH.....	1
LỜI NÓI ĐẦU	1
CHƯƠNG 1: TỔNG QUAN VỀ CÁC CÔNG NGHỆ IOT.....	2
1.1. Tổng quan IoT.....	2
1.2. Những đặc điểm cơ bản và nền tảng hệ thống IoT	3
1.2.1. Đặc điểm cơ bản.....	3
1.2.2. Nền tảng của hệ thống IoT.....	3
1.3. Những ưu điểm của IoT	5
1.4. Những công nghệ IoT trong hệ sinh thái thành phố thông minh	6
1.5. Kết luận	7
CHƯƠNG 2: TỔNG QUAN VỀ ARDUINO VÀ CÁC ỨNG DỤNG CHO NHÀ THÔNG MINH	8
2.1. Tổng quan về Arduino	8
2.2. Các đặc tính cơ bản của Arduino Uno R3	9
2.2.1. Đặc điểm chung về Arduino Uno R3.....	9
2.2.2. Đặc điểm chung về Modul Wifi ESP8266.....	13
2.3. Nghiên cứu webserver cho nhà thông minh.....	17
2.3.1. Khái niệm.....	17
2.3.2. Nguyên lý hoạt động của webserver:.....	18
2.4. Nghiên cứu cơ sở dữ liệu firebase.....	19
2.4.1. Khái niệm.....	19
2.4.2. Lịch sử phát triển	19
2.4.3. Các chức năng chính của Firebase	20
2.4.4. Ưu nhược điểm của Firebase	20
2.5. Nghiên cứu và cài đặt các dịch vụ nền tảng trên smartphone.....	21

2.5.1. Trợ lý ảo Google Assistant	21
2.5.2. Loa thông minh Google Home	23
2.5.3. Nghiên cứu hệ sinh thái Blynk.....	25
2.6. Kết luận	26
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG NHÀ THÔNG MINH.....	27
3.1. Mô tả hoạt động của hệ thống.....	27
3.2. Thiết kế sơ đồ nguyên lý hệ thống	28
3.2.1. Thiết kế sơ đồ khối điều khiển bật tắt thiết bị.....	28
3.2.2. Thiết kế sơ đồ khối nhiệt độ, độ ẩm và chuyển động của ngôi nhà.	30
3.2.3. Tính toán và thiết kế toàn mạch.....	32
3.3. Phần mềm chạy trên hệ điều hành Blynk.....	38
3.4. Kết luận	39
CHƯƠNG 4: TỐI ƯU KỊCH BẢN VÀ XÂY DỰNG MÔ HÌNH NHÀ THÔNG MINH	40
4.1. Kịch bản tối ưu cho một ngôi nhà thông minh	40
4.2. Thi công hệ thống	42
4.2.1. Thi công phần cứng:	43
4.2.2. Lưu đồ giải thuật mạch điều khiển bật tắt thiết bị	46
4.2.3. Lưu đồ giải thuật mạch giám sát nhiệt độ, độ ẩm và chuyển động.	51
4.2.4. Sơ đồ hệ thống nhà thông minh.....	57
4.3. Ứng dụng ngôi nhà hướng đến hệ sinh thái thành phố thông minh.....	58
4.3.1. Bảo mật.....	59
4.3.2. Cảm biến thông minh	59
4.3.3. Mạng kết nối	59
4.3.4. Phân tích dữ liệu thông minh.....	60
KẾT LUẬN VÀ KIẾN NGHỊ.....	61
TÀI LIỆU THAM KHẢO.....	63

DANH MỤC BẢNG

Bảng 4.1 Danh sách các linh kiện sử dụng	43
--	----

DANH MỤC HÌNH

Hình 1.1: Kết nối mọi vật.....	3
Hình 1.2: Mô hình nền tảng IoT.....	4
Hình 2.1. Cấu trúc phần cứng của Arduino Uno R3.....	10
Hình 2.2. Sơ đồ chân trong ATmega 328	11
Hình 2.3: Đo điện áp bằng phần mềm Arduino	12
Hình 2.4: Hình ảnh thực tế của Chip ESP8266.....	14
Hình 2.5. Hình ảnh sơ đồ chân kết nối ESP8266.....	15
Hình 2.6. Cách thức giao tiếp với web server.....	18
Hình 2.7. Trao đổi dữ liệu giữa FIREBASE với các thiết bị.....	19
Hình 2.8: Sử dụng Google Assistant trên điện thoại (nguồn Google API).....	22
Hình 2.9 Trợ lý google hỗ trợ bạn kiểm soát nhiều thiết bị trong ngôi nhà	23
Hình 2.10: Google Home (nguồn Google API).....	24
Hình 2.11: Google Home Mini (nguồn Google API)	24
Hình 2.12: Sơ đồ hệ sinh thái Blynk.....	25
Hình 3.1: Sơ đồ hoạt động của hệ thống giám sát	27
Hình 3.2 Sơ đồ khối hệ thống điều khiển đèn bulb chiếu sáng.....	29
Hình 3.3 Sơ đồ khối mạch giám sát nhiệt độ, độ ẩm và chuyển động.....	31
Hình 3.4 Sơ đồ nguyên lý mạch xử lý trung tâm điều khiển bật tắt thiết bị.....	34
Hình 3.5 Sơ đồ nguyên lý mạch xử lý trung tâm giám sát nhiệt độ, độ ẩm và chuyển động.....	35
Hình 3.6 Sơ đồ kết nối mạch báo động chống trộm.....	36

Hình 3.7 Sơ đồ kết nối toàn mạch.....	38
Hình 3.8 Giao diện hiển thị và điều khiển độ sáng đèn, quạt, nhiệt độ, chuyển động	39
Hình 4.3 Sơ đồ mạch in.....	44
Hình 4.3 Mặt trước mạch in thực tế	45
Hình 4.4 Mặt sau mạch in thực tế	46
Hình 4.5: Lưu đồ giải thuật bật tắt thiết bị.....	46
.....	47
Hình 4.6 Lưu đồ giám sát nhiệt độ, độ ẩm và chuyển động.	52
Hình 4.7 Sơ đồ nguyên lý mạch thực tế.....	57

LỜI NÓI ĐẦU

Trong quá trình phát triển về cuộc cách mạng 4.0, khái niệm về IoT đã trở nên khá quen thuộc và được ứng dụng khá nhiều trong các lĩnh vực của đời sống con người, đặc biệt ở các nước phát triển có nền khoa học công nghệ tiên tiến. Đã làm cho cuộc sống của con người ngày càng trở lên hoàn thiện. Tuy nhiên, những công nghệ này chưa được áp dụng một cách rộng rãi ở nước ta, do những điều kiện về kỹ thuật, kinh tế, nhu cầu sử dụng của con người. Chính vì tầm quan trọng đó nên em đã quyết định chọn đề tài liên quan tới IOT.

Được sự định hướng và chỉ dẫn từ Tiến sĩ Nguyễn Việt Hưng, em đã chọn đề tài luận văn “Xây dựng mô hình Nhà thông minh hướng đến hệ sinh thái Thành phố thông minh” làm đề tài cho luận văn của mình.

Bố cục của luận văn gồm 4 chương:

Chương 1: Tổng quan về các công nghệ IOT

Chương 2: Tổng quan phần cứng và các ứng dụng cho nhà thông minh

Chương 3: Thiết kế hệ thống nhà thông minh

Chương 4: Tối ưu hóa kịch bản và xây dựng mô hình ngôi nhà thông minh

Trong quá trình thực hiện luận văn của mình, dưới sự hướng dẫn khoa học: TS. Nguyễn Việt Hưng em đã cố gắng hết sức để hoàn thiện bài luận văn một cách tốt nhất. Em mong thầy cô và các bạn sẽ đóng góp ý kiến giúp đề tài của em có thể hoàn thiện hơn.

CHƯƠNG 1: TỔNG QUAN VỀ CÁC CÔNG NGHỆ IOT

Trong chương này, em xin được trình bày tổng quan về IoT, vai trò cũng như thực trạng về IoT ở Việt Nam và trên thế giới

1.1. Tổng quan IoT

❖ Khái niệm

Thiết bị (device):

Đối với Internet of Things, đây là một phần của cả hệ thống với chức năng bắt buộc là truyền thông và chức năng không bắt buộc là: cảm biến, thực thi, thu thập dữ liệu, lưu trữ và xử lý dữ liệu.

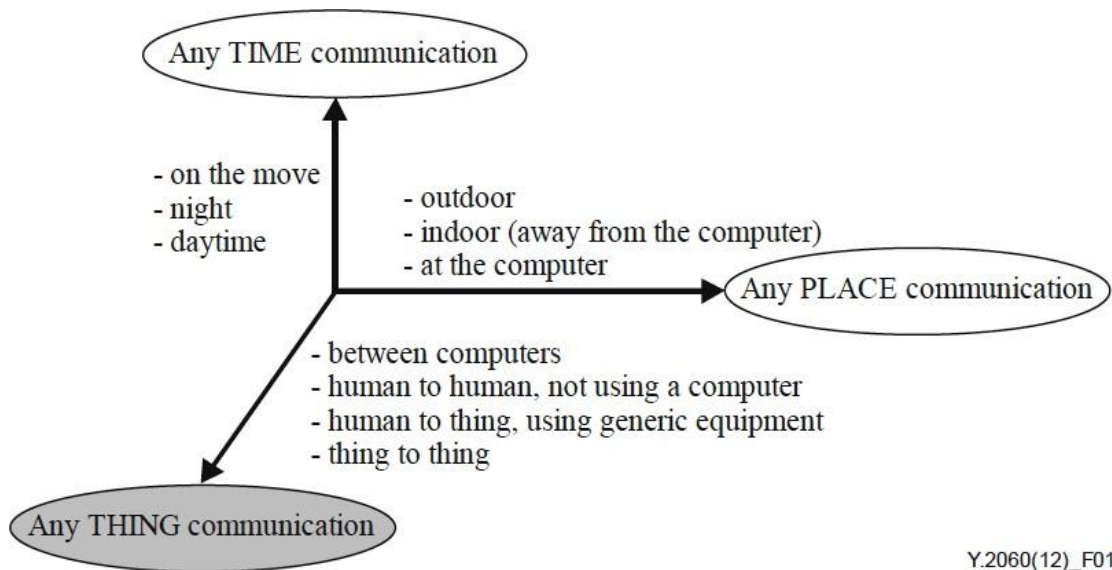
Internet of Things:

Là một cơ sở hạ tầng mang tính toàn cầu cho xã hội thông tin, mang đến những dịch vụ tiên tiến bằng cách kết nối các “Things” (cả physical lẫn virtual) dựa trên sự tin cậy của thông tin, dựa trên khả năng tương tác của các thông tin đó, và dựa trên các công nghệ truyền thông.

Things:

Đối với Internet of Things, “Thing” là một đối tượng của thế giới vật chất (physical things) hay thế giới thông tin ảo (virtual things). “Things” có khả năng được nhận diện, và “Things” có thể được tích hợp vào trong mạng lưới thông tin liên lạc

IoT có thể được coi là một tầm nhìn sâu rộng của công nghệ và trong cuộc sống. Từ quan điểm hiện nay, IoT đã được xem như là một cơ sở hạ tầng mang tính toàn cầu cho xã hội thông tin, tạo điều kiện cho các dịch vụ tiên tiến thông qua các sự liên kết. IoT sẽ được tích hợp rất nhiều công nghệ mới, chẳng hạn như các công nghệ thông tin machine-to-machine, mạng tự trị, khai thác dữ liệu và ra quyết định, bảo vệ sự an ninh và sự riêng tư, điện toán đám mây.



Hình 1.1: Kết nối mọi vật

1.2. Những đặc điểm cơ bản và nền tảng hệ thống IoT

1.2.1. Đặc điểm cơ bản

Đặc điểm cơ bản của IoT bao gồm:

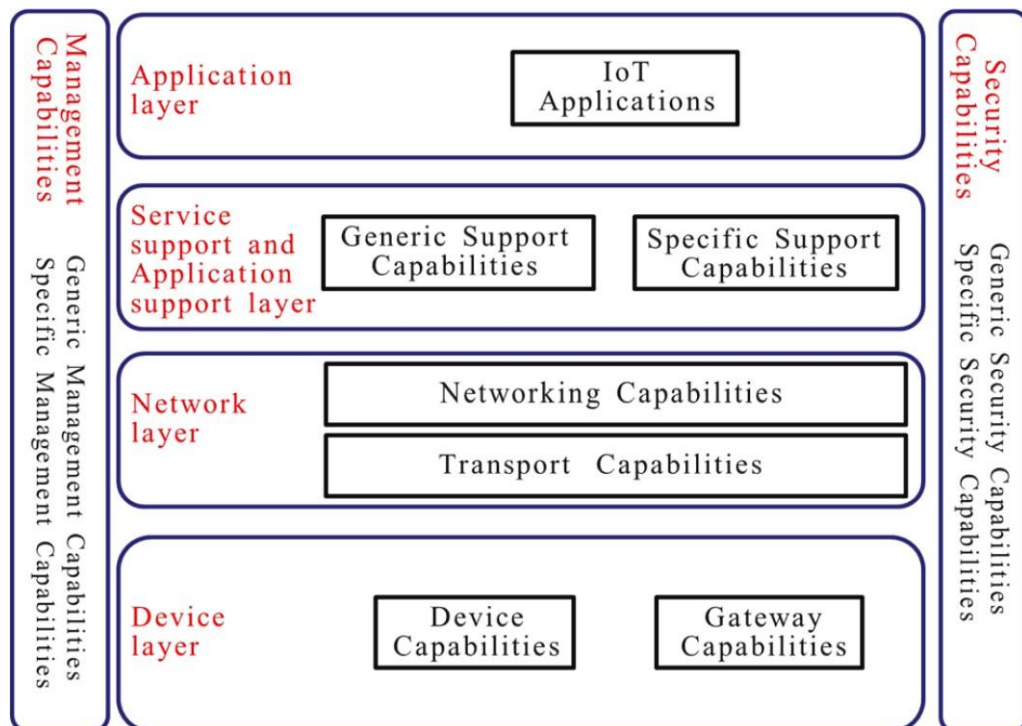
- Tính kết nối liên thông: là khả năng các thiết bị đều có thể kết nối với nhau
- Tính không đồng nhất: Các thiết bị trong mạng lưới IoT sở hữu phần cứng cũng như network khác nhau nên không đồng nhất
- Thay đổi linh hoạt: Số lượng và trạng thái thiết bị đều có thể thay đổi
- Quy mô lớn: mạng lưới IoT có rất nhiều các thiết bị kết nối với nhau thông qua Internet
- Đáp ứng đủ nhu cầu liên quan đến “Things”

1.2.2. Nền tảng của hệ thống IoT

Nền tảng IOT luôn được xây dựng lên từ sự kết hợp của 4 layer sau:

- Application Layer (Lớp ứng dụng).

- Service support and application support layer (Lớp Hỗ trợ dịch vụ và hỗ trợ ứng dụng).
- Network Layer (Lớp mạng).
- Device Layer (Lớp thiết bị).



Hình 1.2: Mô hình nền tảng IoT

1.2.2.1. Application Layer

Lớp ứng dụng cũng tương tự như trong mô hình OSI 7 lớp, lớp này tương tác trực tiếp với người dùng để cung cấp một chức năng hay một dịch vụ cụ thể của một hệ thống IOT.

1.2.2.2. Service support and application support layer

Nhóm dịch vụ chung: Các dịch vụ hỗ trợ chung, phổ biến mà hầu hết các ứng dụng IOT đều cần, ví dụ như xử lý dữ liệu hoặc lưu trữ dữ liệu.

Nhóm dịch vụ cụ thể, riêng biệt: Những ứng dụng IOT khác nhau sẽ có nhóm dịch vụ hỗ trợ khác nhau và đặc thù. Trong thực tế, nhóm dịch vụ cụ thể riêng biệt là tính toán độ tăng trưởng của cây mà đưa ra quyết định tưới nước hoặc bón phân.

1.2.2.3. Network layer

Lớp Network có 2 chức năng:

- *Chức năng Networking*: cung cấp chức năng điều khiển các kết nối kết nối mạng, chẳng hạn như tiếp cận được nguồn tài nguyên thông tin và chuyển tài nguyên đó đến nơi cần thiết, hay chứng thực, uỷ quyền...

- *Chức năng Transporting*: tập trung vào việc cung cấp kết nối cho việc truyền thông tin của dịch vụ/ứng dụng IOT.

1.2.2.4. Device layer

Lớp Device chính là các phần cứng vật lý trong hệ thống IOT. Device có thể phân thành hai loại như sau:

- *Thiết bị thông thường*: Device này sẽ tương tác trực tiếp với network: Các thiết bị có khả năng thu thập và tải lên thông tin trực tiếp (nghĩa là không phải sử dụng gateway) và có thể trực tiếp nhận thông tin (ví dụ, lệnh) từ các network. Device này cũng có thể tương tác gián tiếp với network: Các thiết bị có thể thu thập và tải network gián tiếp thông qua khả năng gateway. Ngược lại, các thiết bị có thể gián tiếp nhận thông tin (ví dụ, lệnh) từ network. Trong thực tế, các Thiết bị thông thường bao gồm các cảm biến, các phần cứng điều khiển motor, đèn...

- *Thiết bị Gateway*: Gateway là các thiết bị biên có thể giao tiếp với hệ thống ngược dòng một trong hai cách: có hoặc không có gateway. Ngoài việc cung cấp cơ chế truyền tải, một thiết bị gateway cũng có thể cung cấp các chức năng tùy chọn như phân tách dữ liệu, dọn dẹp, tổng hợp, khử trùng lặp và tính toán biên...

1.3. Những ưu điểm của IoT

Những vấn đề quan trọng nhất của hệ thống IoT bao gồm trí thông minh nhân tạo, kết nối, cảm biến và các thiết bị nhỏ nhưng mang tính cơ động cao, chúng được mô tả sơ lược như bên dưới:

- *AI (Artificial Intelligence)*: Hệ thống IoT về cơ bản được hiểu là làm cho mọi thiết bị trở nên “thông minh”, nghĩa là nó giúp nâng cao mọi khía cạnh của cuộc sống bằng những dữ liệu thu thập được, thông qua các thuật toán tính toán nhân tạo và kết nối mạng. Một ví dụ đơn giản như hộp đựng gạo của bạn, khi biết rằng gạo sắp hết, hệ thống tự động đặt một đơn hàng mới cho nhà cung cấp.

- *Connectivity*: Là một đặc trưng cơ bản của IoT, hiện nay các mạng thiết bị đang trở nên phổ biến, nhiều mạng thiết bị ngày càng nhỏ hơn, rẻ hơn và được phát triển phù hợp với thực tế cũng như nhu cầu của người dùng.

- *Sensor*: IoT sẽ mất đi sự quan trọng của mình nếu không có sensors. Các cảm biến hoạt động giống như một công cụ giúp IoT chuyển từ mạng lưới các thiết bị thụ động sang mạng lưới các thiết bị tích cực, đồng thời có thể tương tác với thế giới thực.

- *Active Engagement*: Ngày nay, phần lớn các tương tác của những công nghệ kết nối xảy ra 1 cách thụ động. IoT được cho là sẽ đem đến những hệ thống mang tích tích cực về nội dung, sản phẩm cũng như các dịch vụ gắn kết.

- *Small Devices*: Như đã được dự đoán từ trước, các thiết bị ngày càng được tối ưu với mục đích nâng cao độ chính xác, khả năng mở rộng cũng như tính linh hoạt. Nó được thiết kế ngày càng nhỏ hơn, rẻ hơn và mạnh mẽ hơn theo thời gian.

1.4. Những công nghệ IoT trong hệ sinh thái thành phố thông minh

Thành phố thông minh là một khu vực mà ở đó, các nguồn lực, tài sản hiện hữu trong thành phố cùng các mặt hoạt động của thành phố được thực hiện hiệu quả và bền vững nhờ sử dụng công nghệ thông tin và truyền thông (ICT) cùng các công nghệ thông minh khác. Thành phố thông minh sử dụng những thành tựu mới nhất của công nghệ thông tin truyền thông, các thiết bị được kết nối với nhau theo nguyên lý của Internet vạn vật (Internet of Things) nhằm tối ưu hóa quá trình vận hành của thành phố, cung ứng hàng hóa, dịch vụ tốt nhất cho công dân, gắn kết giữa chính quyền và người dân.

Những công nghệ chủ chốt được ứng dụng trong xây dựng thành phố thông minh (hoặc chuyển đổi từ cách vận hành thành phố truyền thống sang vận hành/quản trị thành phố theo mô hình thành phố thông minh) bao gồm:

1. Trí tuệ nhân tạo (AI) và tự động hóa ở mức cao (super automation).
2. Giao tiếp giữa máy với máy (M2M communication) và dịch vụ băng thông rộng di động được sử dụng phổ biến (pervasive broadband mobile).
3. Hệ thống truyền tải năng lượng “thông minh” (“smart” energy grids).
4. Các trợ lý ảo (talking & serviceable “bots”).
5. Phương tiện giao thông tự hành (không người lái - driverless transport).
6. Internet vạn vật (Internet of Everything hoặc Internet of things - IoT).
7. An ninh mạng ở trình độ tân tiến (advanced cybersecurity).
8. Tương tác người-máy (human-machine interface - hiện tại, tương tác giữa máy và người ở nhiều thành phố được thực hiện thông qua các màn hình cảm ứng).
9. Làm việc từ xa (telework), giáo dục từ xa (tele-education) và chữa bệnh/chăm sóc y tế từ xa (tele-health services).
10. Công ty ảo (virtual companies).

Nhờ ứng dụng công nghệ cao, các công chức của thành phố thông minh có thể tương tác trực tiếp với cộng đồng, các cơ sở hạ tầng kỹ thuật của thành phố, theo dõi được những gì đang diễn ra và những diễn biến, trưởng thành, tiến bộ, xu hướng vận động của cả thành phố. Với việc quản trị thành phố theo mô hình thành phố thông minh, nhu cầu của người dân thành phố được đáp ứng tối đa.

1.5. Kết luận

Chương 1 đã trình bày khái quát tổng quan về nền tảng IoT càng lúc càng phát triển với những đặc điểm, tính năng và ứng dụng thực tế cho cuộc sống. Đồng thời chương đầu cũng đã phân tích và giải thích động lực triển khai hệ thống IoT cho hệ sinh thái thành phố thông minh và đưa ra lý do nghiên cứu xây dựng ngôi nhà thông minh dựa trên nền tảng Arduino và các ứng dụng cho ngôi nhà trở nên thông minh để góp phần hoàn thiện hơn cho thành phố thông minh

CHƯƠNG 2: TỔNG QUAN VỀ ARDUINO VÀ CÁC ỨNG DỤNG CHO NHÀ THÔNG MINH

Trong chương này em sẽ giới thiệu tổng quan Arduino; nghiên cứu web server; cơ sở dữ liệu firebase, trợ lý ảo Google Assistant; ứng dụng hệ sinh thái blynk;

2.1. Tổng quan về Arduino

Arduino cơ bản là một mã nguồn mở về điện tử được tạo thành từ phần cứng và phần mềm. Về mặt kỹ thuật có thể coi Arduino là một bộ điều khiển logic có thể lập trình được. Đơn giản hơn, Arduino là thiết bị có thể tương tác với ngoại cảnh thông qua các cảm biến và hành vi được lập trình sẵn. Với thiết bị này việc lắp ráp và điều khiển các thiết bị điện tử sẽ dễ dàng hơn bao giờ hết.

Hiện tại có rất nhiều loại vi điều khiển và đa số được lập trình bằng ngôn ngữ C/C++ hoặc Assembly nên rất khó khăn cho những người có ít kiến thức sâu về điện tử và lập trình. Nó là trở ngại cho mọi người muốn tạo riêng cho mình một món đồ mang tính công nghệ. Song Arduino đã giải quyết được vấn đề này, Arduino được phát triển nhằm đơn giản hóa việc thiết kế, lắp ráp linh kiện điện tử cũng như lập trình trên vi điều khiển và mọi người có thể tiếp cận dễ dàng hơn với thiết bị điện tử mà không cần nhiều về kiến thức điện tử và thời gian.

Những thế mạnh của Arduino so với các nền tảng vi điều khiển khác:

- *Chạy trên đa nền tảng:* Việc lập trình Arduino có thể thực hiện trên các hệ điều hành khác nhau như Windows, Mac Os, Linux trên Desktop, Android trên di động.
- *Ngôn ngữ lập trình đơn giản dễ hiểu.*
- *Mã nguồn mở:* Arduino được phát triển dựa trên nguồn mở nên phần mềm chạy trên Arduino được chia sẻ dễ dàng và tích hợp vào các nền tảng khác nhau.
- *Mở rộng phần cứng:* Arduino được thiết kế và sử dụng theo dạng modul nên việc mở rộng phần cứng cũng dễ dàng hơn.
- *Đơn giản và nhanh:* Rất dễ dàng lắp ráp, lập trình và sử dụng thiết bị.

- *Đễ dàng chia sẻ*: Mọi người dễ dàng chia sẻ mã nguồn với nhau mà không lo lắng về ngôn ngữ hay hệ điều hành mình đang sử dụng.

Arduino được chọn làm bộ não xử lý của rất nhiều thiết bị từ đơn giản đến phức tạp. Trong số đó có một vài ứng dụng thực sự chứng tỏ khả năng vượt trội của Arduino do chúng có khả năng thực hiện nhiều nhiệm vụ rất phức tạp.

Arduino được biết đến nhiều nhất là phần cứng của nó, nhưng phải có phần mềm để lập trình phần cứng. Cả phần cứng và phần mềm gọi chung là Arduino. Để có tạo mạch cho ngôi nhà em đã sử dụng 2 linh kiện chính để thiết kế là Arduino Uno R3 (với chip điều khiển như bộ não ngôi nhà) và module ESP8266 là module wifi được đánh giá rất cao cho các ứng dụng liên quan đến Internet và Wifi cũng như các ứng dụng truyền nhận sử dụng thay thế cho các module RF khác với khoảng cách truyền lên tới 100 mét (môi trường không có vật cản). Trên 400m với anten và router thích hợp.

2.2. Các đặc tính cơ bản của Arduino Uno R3

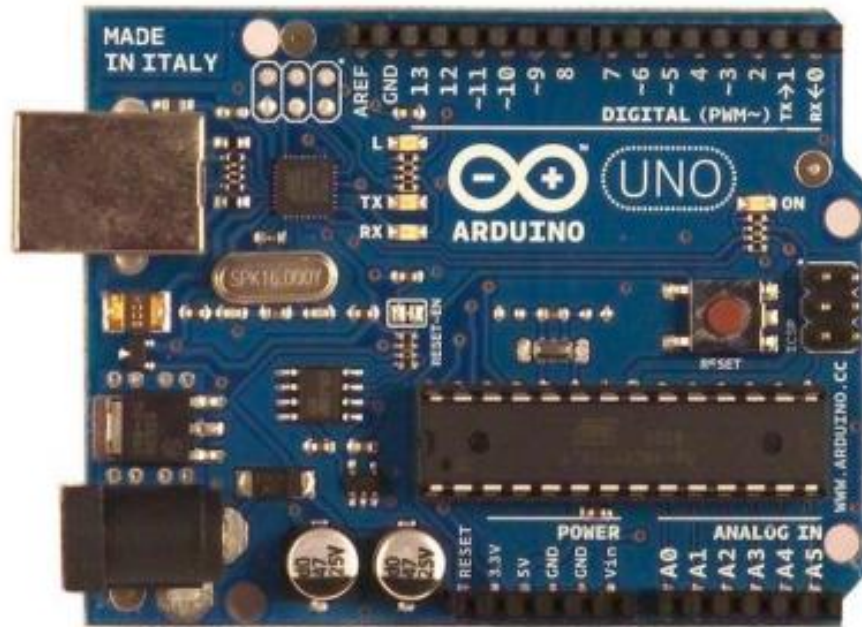
2.2.1. Đặc điểm chung về Arduino Uno R3

2.2.1.1. Khái niệm Arduino Uno R3

Arduino Uno là một bo mạch vi điều khiển dựa trên chip ATmega168 hoặc ATmega 328. Cấu trúc chung bao gồm:

- 14 chân vào ra bằng tín hiệu số, trong đó có 6 chân có thể sử dụng để điều chế độ rộng xung.
- Có 6 chân đầu vào tín hiệu tương tự cho phép chúng ta kết nối với các bộ cảm biến bên ngoài để thu thập số liệu.
- Sử dụng một dao động thạch anh tần số dao động 16MHz.
- Có một cổng kết nối bằng chuẩn USB để chúng ta nạp chương trình vào bo mạch và một chân cấp nguồn cho mạch, một nút reset.
- Nó chứa tất cả mọi thứ cần thiết để hỗ trợ các vi điều khiển, nguồn cung cấp cho Arduino có thể là từ máy tính thông qua cổng USB hoặc là từ bộ nguồn

chuyên dụng được biến đổi từ xoay chiều sang một chiều hoặc là nguồn lấy từ pin.



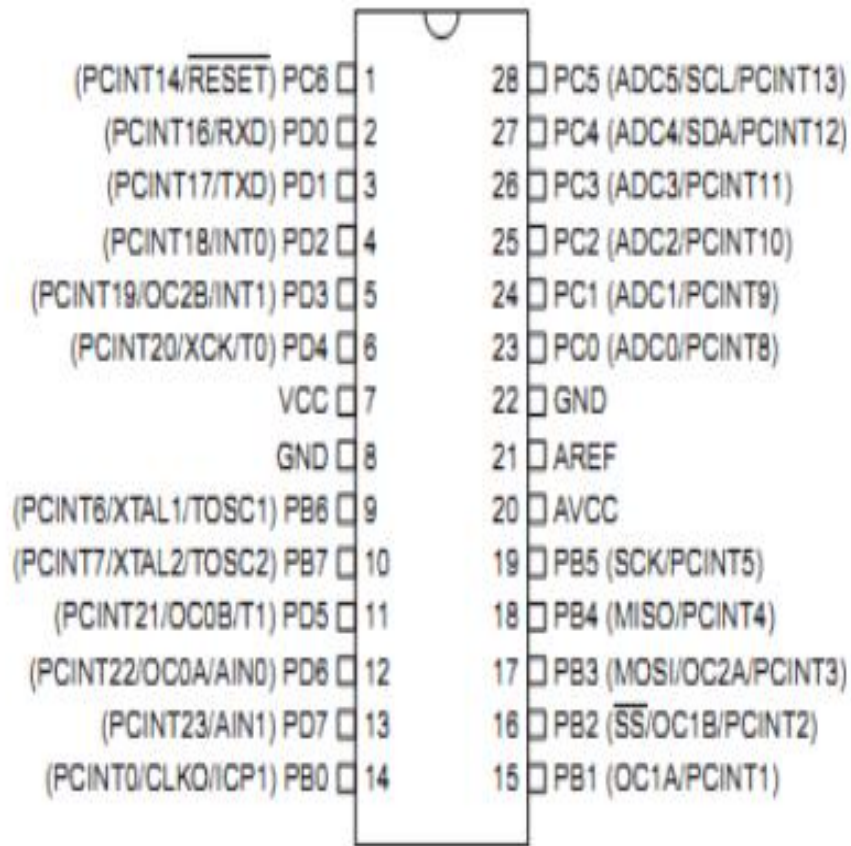
Hình 2.1. Cấu trúc phần cứng của Arduino Uno R3

❖ **Thông số kỹ thuật của Arduino Uno:**

- Khối xử lý trung tâm là vi điều khiển Atmega328.
- Điện áp hoạt động 5V.
- Điện áp đầu vào khuyến nghị là 5-12V.
- Điện áp đầu vào giới hạn 6-20V.
- Dòng điện một chiều trên các chân vào ra là 40mA.
- Dòng điện một chiều cho chân 3.3V là 50mA.
- Clock Speed 16 MHz.
- Flash Memory 16 Kb (ATmega 168) hoặc 32 Kb (ATmega 328), SRAM 1 Kb (ATmega 168) hoặc 2 Kb (ATmega 328), EEPROM 512 bytes (ATmega 168) hoặc 1 Kb (ATmega 328).

❖ **Khối xử lý trung tâm:**

Trong bo mạch Arduino IC đóng vai trò xử lý trung tâm là Atmega328 cấu trúc sơ đồ chân của nó như sau:



Hình 2.2. Sơ đồ chân trong ATmega 328

- Chân VCC (chân số 7): Chân cung cấp điện áp dương nguồn 5V.
- Chân GND (chân số 8): Chân đất chung.
- Chân AREF (chân 21): Là chân tham chiếu để chuyển đổi tín hiệu tương tự sang số.
- Chân AVCC (chân 20): Chân cung cấp điện áp cho quá trình chuyển đổi ADC.
- Cổng B (chân 14 - chân 19, chân 9, chân 10): Bao gồm có 8 chân I/O từ (PB0÷PB7).
- Cổng C (chân 23 – chân 28, chân 1): Bao gồm có 7 chân I/O từ (PC0÷PC6) trong đó chân PC6 (chân số 1) làm chân reset.
- Cổng D (chân 2 – chân 6, chân 11 – chân 13): Bao gồm có 8 chân I/O từ chân (PD0÷PD7).

2.2.1.2. Ưu nhược điểm của Arduino

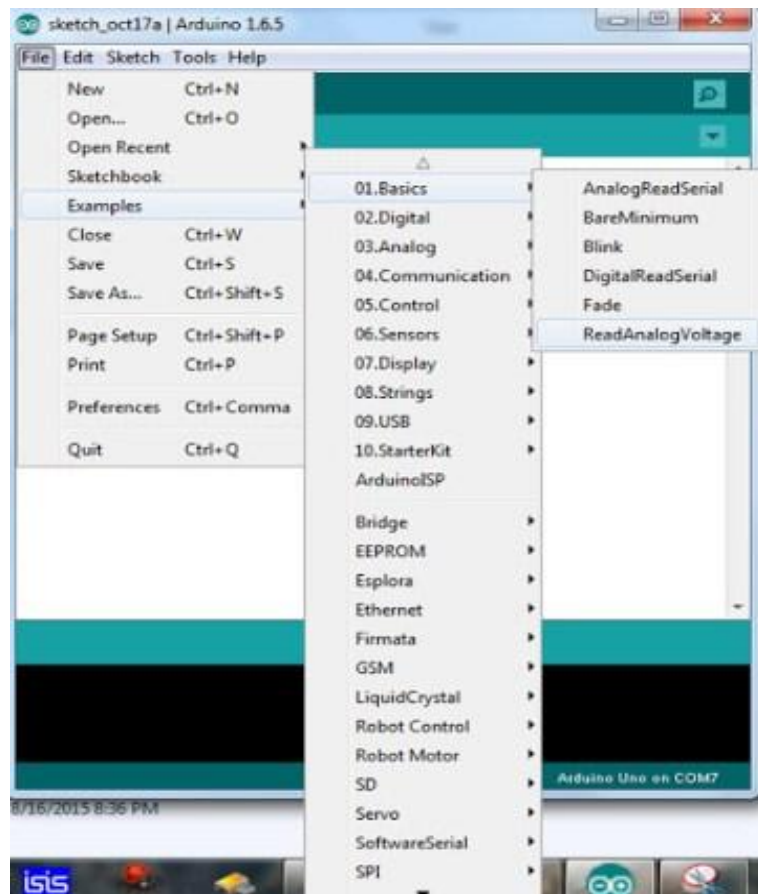
❖ Ưu điểm:

- Có thể sử dụng ngay:

Ưu điểm lớn nhất của Arduino là có thể sử dụng ngay. Vì Arduino là một bộ hoàn chỉnh gồm bộ nguồn 5V, một ổ ghi, một bộ dao động, một vi điều khiển, truyền thông nối tiếp, LED và các giắc cắm. Sẽ không cần phải suy nghĩ về các kết nối lập trình hoặc bất kỳ giao diện nào khác. Chỉ cần cắm nó vào cổng USB của máy tính.

- Các mẫu lệnh có sẵn:

Một ưu điểm lớn khác của Arduino là thư viện các mẫu có sẵn trong phần mềm Arduino. Để nói rõ hơn về ưu điểm này có thể lấy ví dụ về đo lường điện áp. Ví dụ nếu như muốn đo điện áp bằng cách sử dụng vi điều khiển ATmega8 bằng mở mẫu ReadAnalog Voltage có sẵn trong phần mềm.



Hình 2.3: Đo điện áp bằng phần mềm Arduino

- Các chức năng giúp đơn giản hóa công việc:

Trong quá trình mã hóa Arduino, một số chức năng giúp đơn giản hóa công việc. Một ưu điểm khác của Arduino là khả năng chuyển đổi đơn vị tự động của nó và chỉ cần chú ý vào các phần chính của project mà không phải lo lắng về các vấn đề phụ.

- Cộng đồng lớn:

Có rất nhiều diễn đàn trên internet nói về Arduino. Kỹ sư và các chuyên gia đang thực hiện dự án của họ thông qua Arduino.

❖ **Nhược điểm:**

- Cấu trúc:

Cấu trúc của Arduino cũng là nhược điểm của nó. Trong khi xây dựng một dự án bạn phải làm cho kích thước của nó càng nhỏ càng tốt. Nhưng với cấu trúc lớn của Arduino chúng ta phải gắn với PCB có kích thước lớn. Nếu bạn đang làm việc trên vi điều khiển nhỏ như ATmega8 bạn có thể dễ dàng làm PCB càng nhỏ càng tốt.

- Chi phí:

Yếu tố quan trọng nhất mà bạn không thể phủ nhận là chi phí. Đây là vấn đề mà mọi người kỹ sư hoặc chuyên gia phải đối mặt. Lúc này chúng ta phải xem chi phí cho Arduino có hiệu quả hay không.

- Dễ sử dụng:

Vì phần cứng và phần mềm của Arduino thân thiện với người dùng, cơ bản như giao tiếp nối tiếp, ADC, I2C... nên sẽ dễ dàng bị kẻ gian tấn công.

2.2.2. Đặc điểm chung về Modul Wifi ESP8266

2.2.2.1 Khái niệm ESP8266

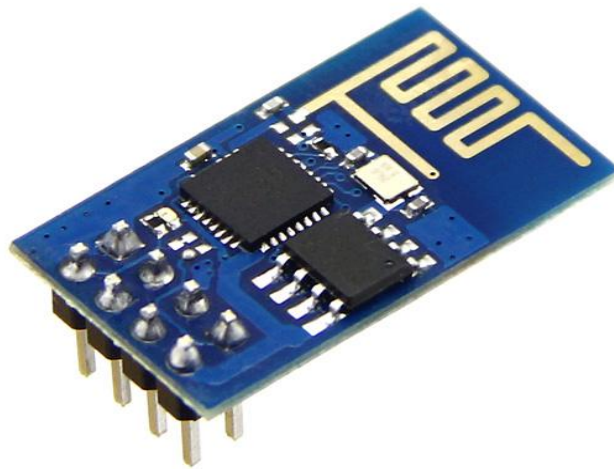
Module ESP8266 là module wifi được đánh giá rất cao cho các ứng dụng liên quan đến Internet và Wifi cũng như các ứng dụng truyền nhận sử dụng thay thế cho các module RF khác với khoảng cách truyền lên tới 100 mét (môi trường không có vật cản). Trên 400m với anten và router thích hợp.

ESP8266 cung cấp một giải pháp kết nối mạng Wi-Fi hoàn chỉnh và khép kín, cho phép nó có thể lưu trữ các ứng dụng hoặc để giảm tải tất cả các chức năng kết nối mạng Wi-Fi từ một bộ xử lý ứng dụng.

Khi ESP8266 là máy chủ các ứng dụng hay khi nó chỉ là bộ vi xử lý ứng dụng có trong thiết bị, nó có thể khởi động trực tiếp từ một flash ngoài. Nó có tích hợp bộ nhớ cache để cải thiện hiệu suất của hệ thống trong các ứng dụng này, và để giảm thiểu các yêu cầu bộ nhớ.

Luôn phiên, phục vụ như một bộ chuyển đổi Wi-Fi, truy cập internet không dây có thể được thêm vào bất kỳ thiết kế vi điều khiển nào dựa trên kết nối đơn giản qua giao diện UART hoặc giao diện cầu CPU AHB.

Khả năng lưu trữ và xử lý mạnh mẽ cho phép nó được tích hợp với các bộ cảm biến, vi điều khiển và các thiết bị ứng dụng cụ thể khác thông qua GPIOs với chi phí tối thiểu và một PCB tối thiểu. Với mức độ tích hợp cao trên chip, trong đó bao gồm các anten chuyển đổi balun, bộ chuyển đổi quản lý điện năng...



Hình 2.4: Hình ảnh thực tế của Chip ESP8266

2.2.2.2. Cấu tạo của ESP8266

Module ESP8266 có 10 chân dùng để cấp nguồn và thực hiện kết nối. Chức năng của các chân như sau:

- + VCC: 3.3V lên đến 300Ma
- + GND: Mass

- + Tx: Chân Tx của giao thức UART, kết nối đến chân Rx của vi điều khiển.
- + Rx: Chân Rx của giao thức UART, kết nối đến chân Tx của vi điều khiển.
- + RST: chân reset, kéo xuống mass để reset.
- + CH_PD: Kích hoạt chip, sử dụng cho Flash Boot và updating lại module
- + GPIO0: kéo xuống thấp cho chế độ update.
- + GPIO2: không sử dụng.



Hình 2.5. Hình ảnh sơ đồ chân kết nối ESP8266

2.2.2.3. Tính năng của ESP8266

- Hỗ trợ chuẩn 802.11 b/g/n.
- Wi-Fi 2.4 GHz, hỗ trợ WPA/WPA2.
- Chuẩn điện áp hoạt động: 3.3V.
- Chuẩn giao tiếp nối tiếp UART với tốc độ Baud lên đến 115200
- Tích hợp ngăn xếp giao thức TCP / IP.
- Tích hợp chuyển đổi TR, balun, LNA, bộ khuếch đại công suất và phù hợp với mạng.
- Tích hợp PLL, bộ quản lý, và các đơn vị quản lý điện năng.

- Công suất đầu ra +19.5dBm trong chế độ 802.11b.
- Tích hợp cảm biến nhiệt độ.
- Hỗ trợ nhiều loại anten.
- Wake up và truyền các gói dữ liệu trong <2ms.
- Chế độ chờ tiêu thụ điện năng <1.0mW (DTIM3).
- Hỗ trợ cả 2 giao tiếp TCP và UDP
- Làm việc như các máy chủ có thể kết nối với 5 máy con
- Hỗ trợ các chuẩn bảo mật như: OPEN, WEP, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK.
- Có 3 chế độ hoạt động: Client, Access Point, Both Client and Access Point.

2.2.2.4. Quản lý năng lượng ESP8266

ESP8266 được thiết kế cho điện thoại di động, điện tử lắp ráp và ứng dụng Internet of Things với mục đích đạt được mức tiêu thụ điện năng thấp nhất với sự kết hợp của nhiều kỹ thuật độc quyền. Kiến trúc tiết kiệm năng lượng hoạt động trong 3 chế độ: chế độ hoạt động, chế độ ngủ và chế độ ngủ sâu.

Bằng cách sử dụng các kỹ thuật quản lý nguồn điện và kiểm soát chuyển đổi giữa chế độ ngủ ESP8266 tiêu thụ chưa đầy 12uA ở chế độ ngủ nhỏ hơn 1.0mW so với (DTIM = 3) hoặc ít hơn 0.5mW (DTIM = 10) để giữ kết nối với các điểm truy cập.

Khi ở chế độ ngủ, chỉ có bộ phận hiệu chỉnh đồng hồ thời gian thực và cơ quan giám sát vẫn hoạt động. Đồng hồ thời gian thực có thể được lập trình để đánh thức ESP8266 ở bất kỳ khoảng thời gian cần thiết nào.

ESP8266 có thể được lập trình để thức dậy khi một điều kiện chỉ định được phát hiện. Tính năng tối thiểu thời gian báo thức này của ESP8266 có thể được sử dụng bởi Tính năng tối thiểu thời gian báo thức của ESP8266 có thể được sử dụng bởi thiết bị di động SOC. Cho phép chúng vẫn ở chế độ chờ, điện năng thấp cho đến khi Wifi là cần thiết.

Để đáp ứng nhu cầu điện năng của thiết bị di động và điện tử lắp giáp, ESP8266 có thể được lập trình để giảm công suất đầu ra của PA phù hợp với các ứng dụng khác nhau. Bằng việc tắt khoảng tiêu thụ năng lượng.

Các chip có thể được thiết lập ở các trạng thái sau:

- OFF: chân CHIP_PD ở mức thấp. Các RTC (đồng hồ thời gian) bị vô hiệu hóa và mọi thanh ghi sẽ bị xóa.
- SLEEP DEEP: Các RTC được kích hoạt, khi đó các phần còn lại của chip sẽ ở trạng thái off. RTC phục hồi bộ nhớ nội bộ để lưu trữ các thông tin kết nối WiFi cơ bản.
- SLEEP: Chỉ RTC hoạt động. Các dao động tinh thể được vô hiệu hóa. Bất kỳ sự kiện wakeup (MAC, host, RTC hẹn giờ, ngắt ngoài) sẽ đưa chip vào trạng thái wakeup.
- Wakeup: Trong trạng thái này, hệ thống đi từ trạng thái ngủ sang trạng thái PWR. Các dao động tinh thể và PLLs được kích hoạt.
- Trạng thái ON: Xung clock tốc độ cao hoạt động và gửi đến mỗi khối được kích hoạt bằng cách đăng ký kiểm soát xung clock. Mức độ thấp hơn clock gating được thực hiện ở cấp khối, bao gồm cả CPU, có thể đạt được bằng cách sử dụng lệnh WAIT, trong khi hệ thống trên off.

2.3. Nghiên cứu webserver cho nhà thông minh

2.3.1. Khái niệm

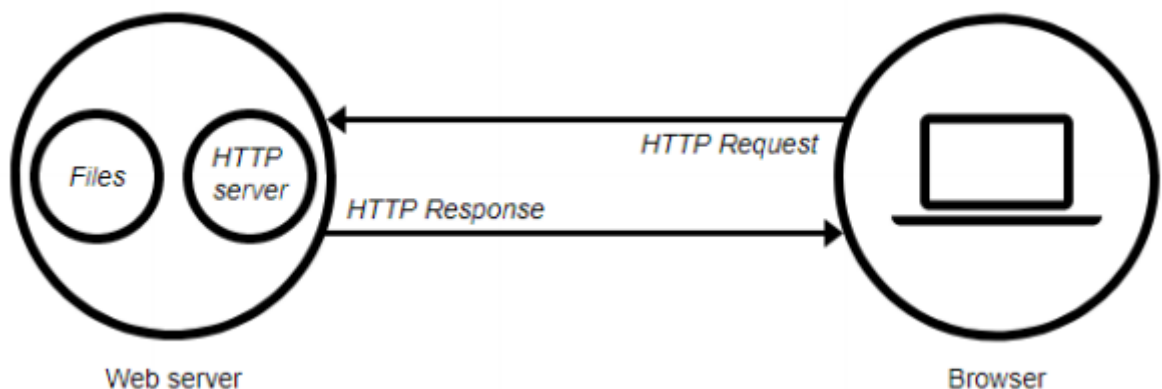
Web server dịch ra tiếng Việt nghĩa là máy chủ. Web server là máy tính lớn được kết nối với tập hợp mạng máy tính mở rộng. Đây là một dạng máy chủ trên internet mỗi máy chủ là một IP khác nhau và có thể đọc các ngôn ngữ như file *.htm và *.html... Tóm lại máy chủ là kho để chứa toàn bộ dữ liệu hoạt động trên internet mà nó được giao quyền quản lý.

Web server phải là một máy tính có dung lượng lớn, tốc độ rất cao để có thể lưu trữ vận hành tốt một kho dữ liệu trên internet. Nó sẽ điều hành trọn chu cho một

hệ thống máy tính hoạt động trên internet, thông qua các cổng giao tiếp riêng biệt của mỗi máy chủ. Các web server này phải đảm bảo hoạt động liên tục không ngừng nghỉ để duy trì cung cấp dữ liệu cho mạng lưới máy tính của mình.

2.3.2. Nguyên lý hoạt động của webserver:

Giao thức HTTP hoạt động dựa trên Client-Server. Nó vận hành theo cơ chế yêu cầu - trả lời (request-responses), khi client kết nối đến server nó sẽ gửi một yêu cầu đến server bao gồm các thông tin header, server sẽ dựa vào header này để xác định sẽ gửi data gì về cho client. Dựa trên tìm hiểu HTTP server. Ta sẽ làm một ứng dụng điều khiển LED thông qua webserver trong mạng local. Khi một trình duyệt cần một file lưu trữ trên một web server, trình duyệt sẽ yêu cầu (request) file đó thông qua HTTP. Khi một yêu cầu gửi tới đúng web server (phần cứng), HTTP server (phần mềm) sẽ gửi file được yêu cầu cũng thông qua.



Hình 2.6. Cách thức giao tiếp với web server

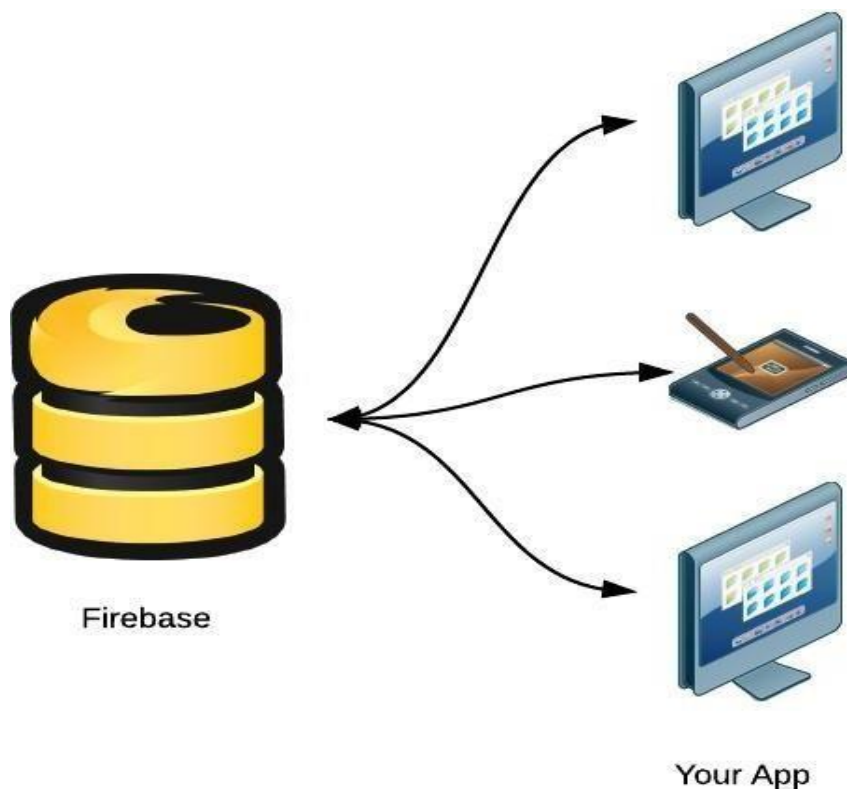
Web server hỗ trợ giao thức HTTP (Giao thức truyền phát siêu văn bản). HTTP là cách truyền các siêu văn bản giữa hai máy tính. HTTP cung cấp các quy tắc rõ ràng, về cách client và server giao tiếp với nhau: Chỉ client có thể tạo ra các HTTP request tới các server. Các server chỉ có thể phản hồi HTTP request của client. Khi yêu cầu một file thông qua HTTP, client phải cung cấp URL của file đó. Web server phải trả lời mọi HTTP request. Trên web server, HTTP server chịu trách nhiệm xử lý và trả lời các request đã được client gửi đến:

- Khi nhận một request, HTTP server sẽ kiểm tra xem URL được yêu cầu có khớp với một file hiện có không.
- Nếu có, web server gửi nội dung file trả lại client. Nếu không, một application server sẽ tạo ra file cần thiết.
- Nếu không thể xử lý, web server trả lại một thông điệp lỗi cho client.

2.4. Nghiên cứu cơ sở dữ liệu firebase

2.4.1. Khái niệm

Firebase là một dịch vụ API (giao diện lập trình ứng dụng) để lưu trữ và đồng bộ dữ liệu giữa hai hay nhiều thiết bị với nhau. Firebase hoạt động dựa trên nền tảng đám mây được cung cấp bởi Google nhằm giúp đỡ các lập trình viên phát triển nhanh ứng dụng bằng cách đơn giản hóa các thao tác ứng dụng với cơ sở dữ liệu



Hình 2.7. Trao đổi dữ liệu giữa FIREBASE với các thiết bị

2.4.2. Lịch sử phát triển

Firebase được thành lập bởi Tamplin và Lee. Hai nhà sáng lập này đã dựa vào dịch vụ API chat trực tuyến vào trang web được cung cấp bởi Envelope, phát triển sử

dụng Envolv để đồng bộ hóa dữ liệu các trạng thái trò chơi trong thời gian thực lên trang web. Dựa vào yếu tố này Tamplin và Lee đã quyết định tách riêng hệ thống chat và kiến trúc thời gian thực để thành lập một cơ sở dữ liệu firebase riêng biệt vào tháng 4 năm 2012. Vào ngày 21 tháng 10 năm 2014 Google đã mua lại Firebase.

2.4.3. Các chức năng chính của Firebase

- *Realtime Database – Cơ sở dữ liệu thời gian thực*

Firebase lưu trữ dữ liệu database dưới dạng JSON và thực hiện đồng bộ database tới tất cả các client theo thời gian thực. Chúng ta có thể xây dựng được client đa nền tảng (cross-platform client) và tất cả các client này sẽ cùng sử dụng chung 1 database đến từ Firebase và có thể tự động cập nhật mỗi khi dữ liệu trong database được thêm mới hoặc sửa đổi.

- *Firebase Authentication – Hệ thống xác thực của Firebase*

Với Firebase chúng ta có thể dễ dàng tích hợp các công nghệ xác thực của Google, Facebook, Twitter, ... hoặc một hệ thống xác thực mà chúng ta mình tạo ra từ trong ứng dụng ở bất kỳ nền tảng nào như Android, iOS hoặc Web.

- *Firebase Hosting*

Chúng ta có thể triển khai một ứng dụng nền web chỉ với vài giây với hệ thống Firebase, và các dữ liệu sẽ được lưu trữ đám mây đồng thời được bảo mật thông qua giao thức SSL.

2.4.4. Ưu nhược điểm của Firebase

❖ Ưu điểm

- Triển khai ứng dụng cực nhanh.
- Tính bảo mật cao.
- Linh hoạt và mở rộng ứng dụng dễ dàng.
- Tình ổn định cao, ít khi gặp trường hợp sập server.

- Người đăng ký được sử dụng miễn phí 1GB dung lượng lưu trữ.
- ❖ Nhược điểm
- Đăng ký tài khoản miễn phí thì chỉ được tối đa 100 thiết bị hoặc người truy cập trong khi có tính phí thì không giới hạn thiết bị hoặc người truy cập.

2.5. Nghiên cứu và cài đặt các dịch vụ nền tảng trên smartphone

2.5.1. Trợ lý ảo Google Assistant

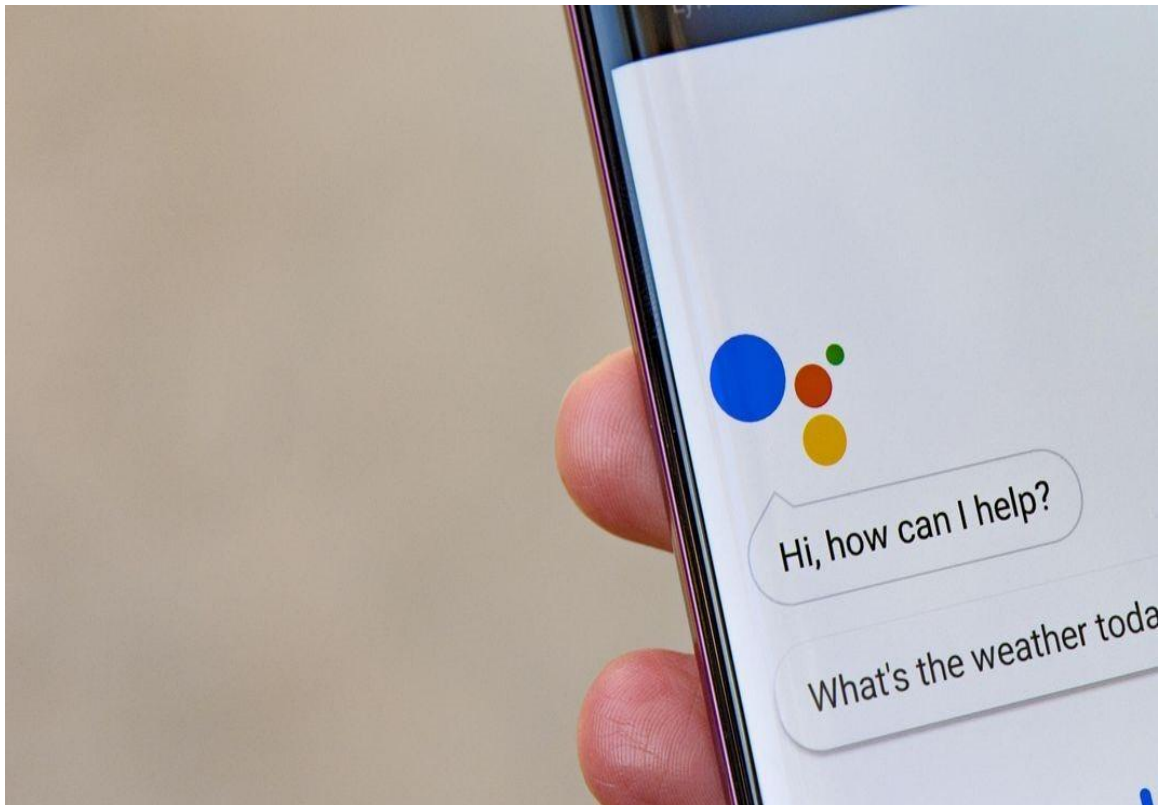
a) Khái niệm

Google Assistant là một trợ lý ảo thông minh tương tự như Siri của Apple, Bixby của Samsung hay Cortana của Microsoft trên Windows. Google Assistant hỗ trợ nhiều tính năng thông minh như tìm kiếm thông tin hay thực hiện yêu cầu mà người dùng đưa ra như mở danh bạ, gọi điện cho người có tên trong danh bạ, đọc tin nhắn, mở nhạc...

Là một trợ lý cá nhân ảo được phát triển bởi Google và có thể tham gia các cuộc trò chuyện hai chiều. Assistant ban đầu được đưa vào ứng dụng nhắn tin Google Allo, và loa thông minh Google Home. Google Assistant đã và đang được tiếp tục mở rộng hỗ trợ cho một lượng lớn thiết bị, bao gồm cả xe hơi và các thiết bị nhà thông minh. Các chức năng của Assistant cũng có thể được bổ sung bởi các nhà phát triển bên thứ ba.

Hiểu một cách đơn giản thì Google Assistant chính là một trợ lý ảo của Google giống như Siri của Apple hay Bixby của Samsung. Assistant được coi như là phiên bản nâng cấp của Google Now, cho phép mở rộng khả năng điều khiển bằng giọng nói câu lệnh "OK Google".

Hiện nay Google Assistant đã hỗ trợ ứng dụng tải riêng trên CH Play hay App Store, do đó bạn hoàn toàn có thể tải về cho mọi dòng máy và trải nghiệm nó.



Hình 2.8: Sử dụng Google Assistant trên điện thoại (nguồn Google API).

b) Ưu điểm và nhược điểm

- *Ưu điểm*

- Khả năng nhận dạng Tiếng Việt trên Google Assistant rất tốt.
- Người dùng thao tác mà không cần chạm vào điện thoại nhờ yêu cầu gọi điện hoặc nhắn tin cho một ai đó, hẹn giờ, đặt nhắc nhở, lên lịch hẹn...
- Trợ lý của Google có thể làm tính, chuyển đổi đại lượng, tính tỷ giá... trả lời các câu hỏi liên quan đến thể thao, du lịch, tìm nhà hàng, trạm xăng...
- Google Assistant cho phép người dùng điều khiển các thiết bị thông minh trong nhà bằng Tiếng Việt.

- *Nhược điểm*

- Nhiều câu hỏi trợ lý ảo không đưa ra câu trả lời trực tiếp mà chỉ gợi ý các kết quả từ Google Search.
- Mặc dù đã hỗ trợ Tiếng Việt, nhưng với các câu hỏi cho nhà thông minh, Assistant lại trả lời bằng Tiếng Anh.

c) Trợ lý ảo hỗ trợ cho ngôi nhà thông minh

Nền tảng Google Smart Home cho phép người dùng kiểm soát các thiết bị được kết nối thương mại của bạn thông qua ứng dụng Google Home và Trợ lý Google, hiện có trên hơn 1 tỷ thiết bị, như loa thông minh, điện thoại, ô tô, TV, tai nghe, đồng hồ, v.v.



Hình 2.9 Trợ lý google hỗ trợ bạn kiểm soát nhiều thiết bị trong ngôi nhà

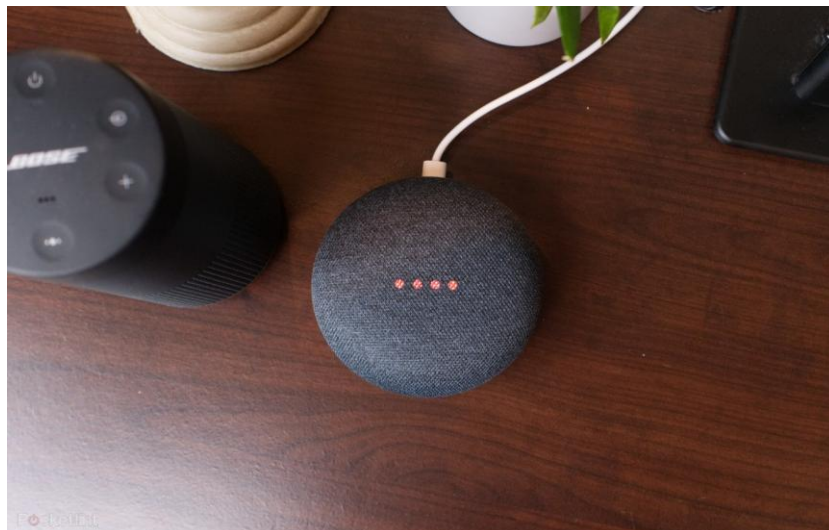
2.5.2. Loa thông minh Google Home

Google Home là một thương hiệu loa thông minh được phát triển bởi Google. Các thiết bị cho phép người dùng nói lệnh thoại để tương tác với các dịch vụ thông qua Google Assistant, trợ lý ảo của công ty. Cả hai dịch vụ nội bộ và bên thứ ba đều được tích hợp, cho phép người dùng nghe nhạc, kiểm soát phát lại video hoặc ảnh hoặc nhận cập nhật tin tức hoàn toàn bằng giọng nói. Các thiết bị Google Home cũng có hỗ trợ tích hợp cho tự động hóa gia đình, cho phép người dùng điều khiển các thiết bị gia đình thông minh bằng giọng nói của họ. Thiết bị Google Home đầu tiên được phát hành tại Hoa Kỳ vào tháng 11 năm 2016, với các sản phẩm tiếp theo được phát hành trên toàn cầu từ 2017.



Hình 2.10: Google Home (nguồn Google API)

Thông qua các bản cập nhật phần mềm cho các thiết bị Google Home và Google Assistant, chức năng bổ sung đã được thêm vào theo thời gian. Nhiều loa có thể được đặt trong các phòng khác nhau trong một ngôi nhà để phát lại nhạc đồng bộ. Vào tháng 5 năm 2017, Google đã công bố nhiều bản cập nhật, bao gồm: gọi điện thoại rảnh tay miễn phí tại Hoa Kỳ và Canada; cập nhật chủ động trước các sự kiện theo lịch trình; phản hồi trực quan trên thiết bị di động hoặc TV có hỗ trợ Chromecast; Truyền phát âm thanh Bluetooth; và khả năng thêm lời nhắc và lịch hẹn.



Hình 2.11: Google Home Mini (nguồn Google API)

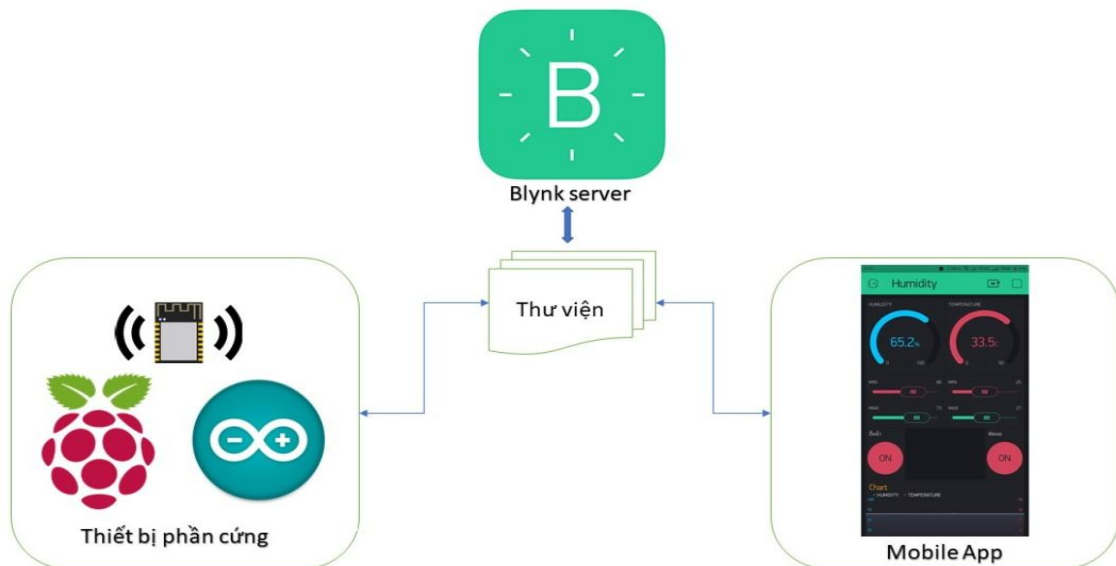
2.5.3. Nghiên cứu hệ sinh thái Blynk

a) Khái niệm

Blynk là một ứng dụng chạy trên nền tảng iOS và Android để điều khiển và giám sát thiết bị thông qua internet. Blynk không bị ràng buộc với những phần cứng cụ thể nào cả, thay vào đó, nó hỗ trợ phần cứng cho bạn lựa chọn như Arduino, Raspberry Pi, ESP8266 và nhiều module phần cứng phổ biến khác.

b) Có ba thành phần chính trong nền tảng Blynk:

- Blynk App - cho phép tạo giao diện cho sản phẩm của bạn bằng cách kéo thả các widget khác nhau mà nhà cung cấp đã thiết kế sẵn.
- Blynk Server - chịu trách nhiệm xử lý dữ liệu trung tâm giữa điện thoại, máy tính bảng và phần cứng. Chúng ta có thể sử dụng Blynk Cloud của Blynk cung cấp hoặc tự tạo máy chủ Blynk riêng cho mình. Vì đây là mã nguồn mở, nên bạn có thể dễ dàng integrate vào các thiết bị và thậm chí có thể sử dụng Raspberry Pi làm server của hệ thống.
- Blynk Libraries – support cho hầu hết tất cả các nền tảng phần cứng phổ biến - cho phép giao tiếp với máy chủ và xử lý tất cả các lệnh đến và đi.



Hình 2.12: Sơ đồ hệ sinh thái Blynk

Tính năng của Blynk khá là thân thiện và tiện ích, cung cấp API & giao diện người dùng tương tự cho tất cả các thiết bị và phần cứng được hỗ trợ:

- Kết nối với server bằng cách sử dụng: Wifi, Bluetooth và BLE, Ethernet, USB (Serial), GSM, ...
- Các tiện ích trên giao diện được nhà cung cấp dễ sử dụng.
- Thao tác kéo thả trực tiếp giao diện mà không cần viết mã.
- Dễ dàng tích hợp và thêm chức năng mới bằng cách sử dụng các cổng kết nối ảo được tích hợp trên blynk app.
- Theo dõi lịch sử dữ liệu.
- Thông tin liên lạc từ thiết bị đến thiết bị bằng Widget.

2.6. Kết luận

Chương 2 đã tìm hiểu tổng quan về nền tảng Arduino, cũng như nghiên cứu về các dịch vụ firebase, webserver để lưu trữ dữ liệu người dùng và các ứng dụng điện thoại như google assistant hay blynk để giúp điều khiển ngôi nhà thông minh.

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG NHÀ THÔNG MINH

Trong chương này sẽ trình bày về các bước tính toán và xây dựng hệ thống; thiết kế sơ đồ nguyên lý; phần mềm chạy trên hệ điều hành Blynk;

3.1. Mô tả hoạt động của hệ thống

Hình dưới trình bày sơ đồ của hệ thống giám sát nhiệt độ, độ ẩm, thiết bị ngoại vi từ xa qua mạng internet ứng dụng trong tòa nhà



Hình 3.1: Sơ đồ hoạt động của hệ thống giám sát

- Hệ thống hoạt động dựa trên sự kết hợp của hai module là module webserver và module đo điều khiển. Module webserver sẽ lưu dữ liệu mà module điều khiển gửi lên. Module webserver ngoài chức năng lưu trữ dữ liệu từ mạch đo gửi lên còn có chức năng hiển thị giao diện điều khiển thiết bị, dữ liệu về nhiệt độ, độ ẩm ra giao diện người dùng.

- Khi nhận được tín hiệu nhiệt độ báo về từ cảm biến thì bộ vi xử lý của hệ thống Arduino sẽ xử lý tín hiệu để chuyển sang module Wifi ESP8266 và sau đó truyền lên server thông qua môi trường ko dây wifi.

- Khi module wifi ESP8266 gửi tín hiệu lên server người dùng có thể truy cập vào hệ thống để điều khiển thiết bị trong gia đình và giám sát nhiệt độ, độ ẩm tại bất kỳ đâu có mạng internet.

- Tại web server, thông qua mạng LAN (nội bộ) hoặc thông qua internet, nhà quản lý có thể sử dụng tính năng của phần mềm nền tảng web (*web-based software*) để:

- + Giám sát nhiệt độ liên tục tại các thời điểm trong ngày. Có thể xem nhiệt độ tại các điểm đo trong cùng một thời điểm
- + Hiện thị nhiệt độ cao nhất và thời điểm nhiệt độ cao nhất, thuận tiện cho việc theo dõi nhiệt độ vượt ngưỡng

3.2. Thiết kế sơ đồ nguyên lý hệ thống

3.2.1. Thiết kế sơ đồ khối điều khiển bật tắt thiết bị

1. Yêu cầu

- Đóng mở qua App Android, IOS hoặc Google Assistant thông qua Wifi hoặc 3G.
- Điều chỉnh được độ sáng dùng App Android, IOS hoặc Google Assistant thông qua Wifi hoặc 3G.
- Hẹn giờ đóng mở đèn bằng App Android thông qua Wifi hoặc 3G.
- Lắp đặt dựa vào cách đi dây hiện tại của ngôi nhà.

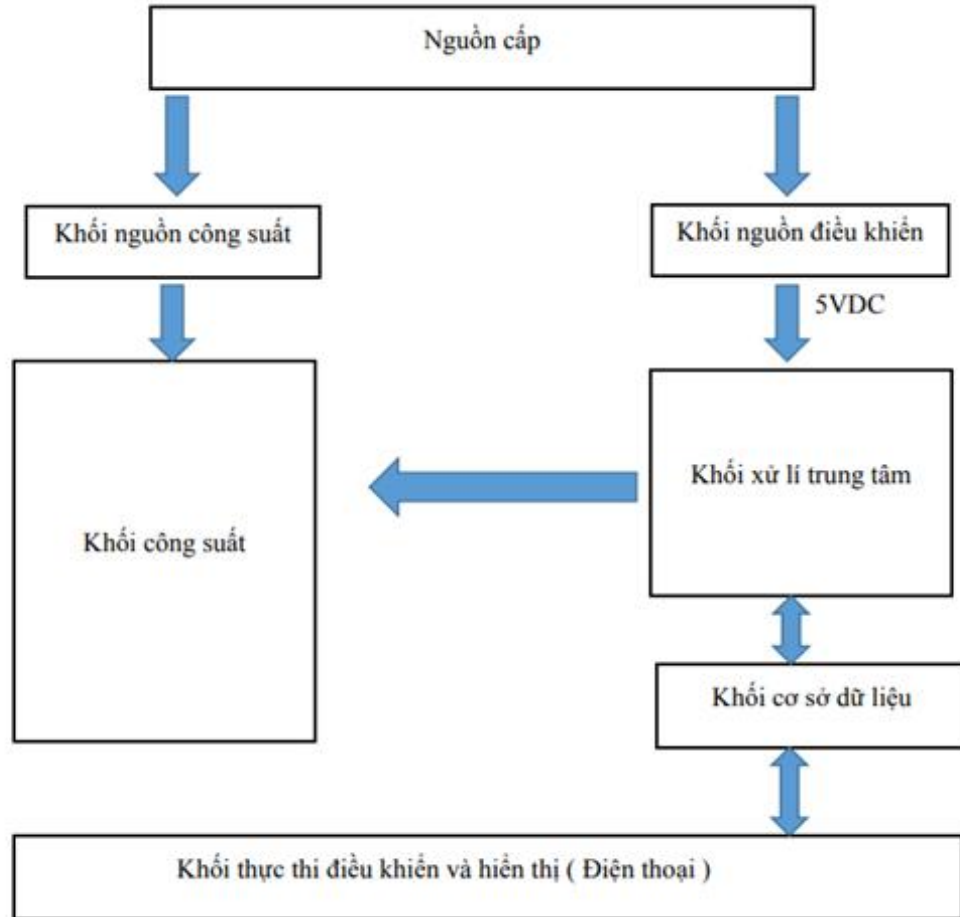
2. Phương án thiết kế

- Chọn các bóng đèn led, quạt để đảm bảo tính thẩm mỹ.
- Dùng ESP8266 ESP-12E làm vi điều khiển trung tâm để tạo xung PWM, và kết nối Wifi để nhận lệnh điều khiển từ điện thoại.

3. Sơ đồ khối mạch điều khiển hệ thống nhà thông minh

- Khối nguồn công suất
- Khối nguồn điều khiển
- Khối xử lý trung tâm
- Khối công suất

- Khối cơ sở dữ liệu
- Khối thực thi điều khiển và hiển thị



Hình 3.2 Sơ đồ khối hệ thống điều khiển đèn bulb chiếu sáng

4. Chức năng từng khối

- Khối nguồn công suất

Đây là nguồn có sẵn cung cấp điện áp cho khối công suất và cho đèn, quạt hoạt động.

- Khối nguồn điều khiển

Đây là nguồn xung nhỏ gọn điện áp ngõ ra 5VDC dòng khoảng 700mA cung cấp nguồn cho mạch xử lý trung tâm (ESP8266 ESP-12E NodeMCU)

- Khối xử lý trung tâm (ESP8266 ESP-12E NodeMCU)

Khối này cập nhật dữ liệu từ cơ sở dữ liệu sau đó xử lý điều khiển khối công suất.

- Khối cơ sở dữ liệu

Đây là nơi lưu trữ dữ liệu giữa khối xử lý trung tâm và khối thực thi điều khiển.

- Khối công suất

Nhiệm vụ khối này là điều khiển độ sáng của đèn hoặc tắt mở đèn thông qua lệnh điều khiển của khối xử lý trung tâm. Cũng như điều khiển quạt và tốc độ quạt thông qua lệnh điều khiển

- Khối thực thi điều khiển và hiển thị

Đây là ứng dụng được viết trên điện thoại hệ điều hành dùng để gửi dữ liệu lên cơ sở dữ liệu và lấy dữ liệu về để hiển thị.

3.2.2. Thiết kế sơ đồ khối nhiệt độ, độ ẩm và chuyển động của ngôi nhà.

1. Yêu cầu

- Đo chính xác nhiệt độ, độ ẩm với sai số nhỏ.
- Bật tắt được cảm biến chuyển động.
- Báo động khi nhiệt độ vượt ngưỡng cho phép hoặc khi phát hiện chuyển động.
- Đưa các dữ liệu thu thập lên cơ sở dữ liệu.
- Board mạch nhỏ gọn.

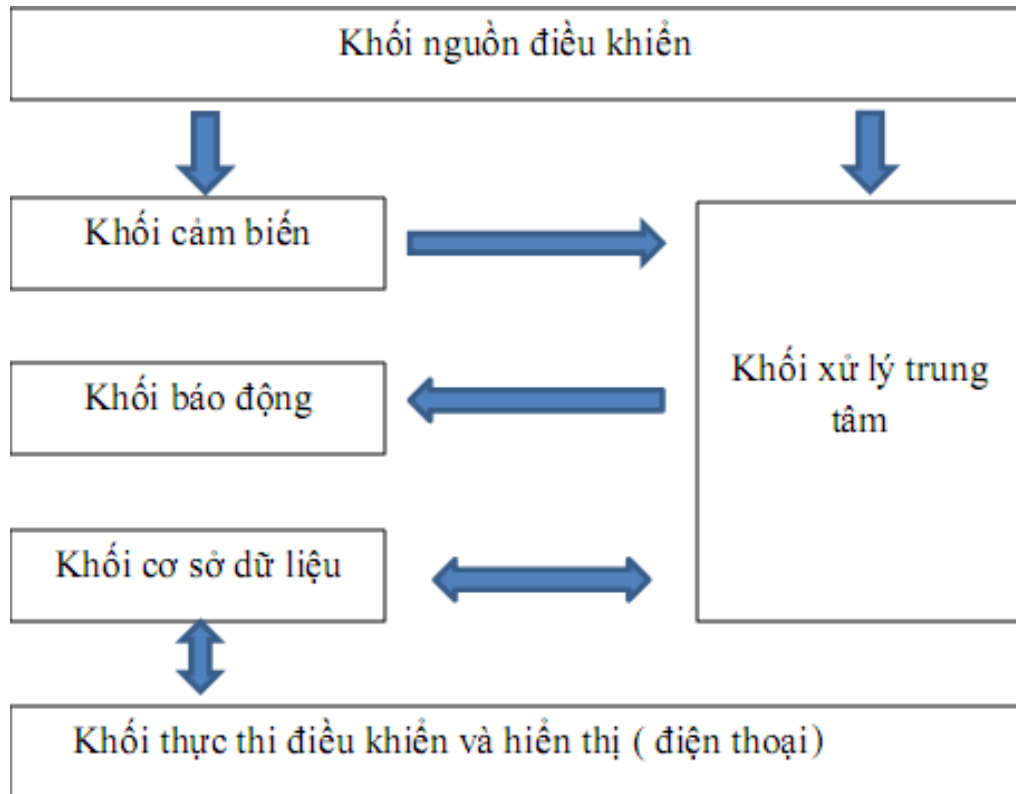
2. Phương án thiết kế

- Chọn ESP8266 ESP-12E làm vi điều khiển trung tâm để giao tiếp với các cảm biến và xử lý tín hiệu. Kết nối Wifi để đưa dữ liệu lên cơ sở dữ liệu.
- Dùng cảm biến DHT11 để đo nhiệt độ, độ ẩm và cảm biến PIR AM312 để phát hiện chuyển động.
- Dùng Buzzer để báo động.

3. Sơ đồ khối mạch giám sát nhiệt độ, độ ẩm và chuyển động.

- Khối nguồn điều khiển
- Khối xử lý trung tâm

- Khối cảm biến
- Khối cơ sở dữ liệu
- Khối thực thi điều khiển và hiển thị
- Khối báo động



Hình 3.3 Sơ đồ khối mạch giám sát nhiệt độ, độ ẩm và chuyển động

4. Chức năng từng khối

- Khối nguồn điều khiển

Đây là nguồn xung nhỏ gọn điện áp ngõ ra 5VDC dòng khoảng 700mA cung cấp nguồn cho toàn mạch.

- Khối xử lý trung tâm (ESP8266 ESP-12E NodeMCU)

Khối này đọc giá trị từ khối cảm biến, xử lý để đưa ra khối báo động và cập nhật dữ liệu lên cơ sở dữ liệu.

- Khối cơ sở dữ liệu

Đây là nơi lưu trữ dữ liệu giữa khối xử lý trung tâm và khối thực thi điều khiển và hiển thị.

- Khối cảm biến

Dùng cảm biến DHT11 đo nhiệt độ, độ ẩm và cảm biến PIR AM312 giám sát chuyển động, sau đó gửi dữ liệu đến khối xử lý trung tâm.

- Khối báo động

Dùng buzzer báo động khi phát hiện chuyển động hoặc nhiệt độ tăng cao vượt ngưỡng cho phép.

- Khối thực thi điều khiển và hiển thị

Đây là ứng dụng được viết trên điện thoại dùng để gửi dữ liệu lên cơ sở dữ liệu và lấy dữ liệu về để hiển thị lên điện thoại và LCD trên mạch hệ thống.

3.2.3. Tính toán và thiết kế toàn mạch

a) Thiết kế khối xử lý trung tâm

Do khối xử lý trung tâm của 2 mạch: mạch điều khiển độ sáng đèn và mạch giám sát nhiệt độ, độ ẩm và chuyển động tương tự nhau nên các linh kiện lựa chọn sẽ giống nhau. Nên em tính toán và thiết kế như sau:

Bảng 3.1. Bảng tiêu thụ dòng ở các chế độ khác nhau của ESP8266 NodeMCU

Parameter	Typical	Unit
Tx 802.11b, CCK 11Mbps, $P_{OUT}=+17\text{dBm}$	170	mA
Tx 802.11g, OFDM 54Mbps, $P_{OUT}=+15\text{dBm}$	140	mA
Tx 802.11n, MCS7, $P_{OUT}=+13\text{dBm}$	120	mA
Rx 802.11b, 1024 bytes packet length, -80dBm	50	mA
Rx 802.11g, 1024 bytes packet length, -70dBm	56	mA
Rx 802.11n, 1024 bytes packet length, -65dBm	56	mA
Modem-Sleep	15	mA
Light-Sleep	0.5	mA
Power save mode DTIM 1	1.2	mA
Power save mode DTIM 3	0.9	mA
Deep-Sleep	10	μA
Power OFF	0.5	μA

Dựa vào bảng tiêu thụ dòng của ESP8266 ở các chế độ khác nhau, để ESP8266 hoạt động tốt trong những chế độ đó thì dòng tiêu thụ phải lớn hơn 170mA cho nên

em chọn dòng tiêu thụ ESP8266 là 300mA để đảm bảo ESP8266 hoạt động tốt ở mọi chế độ tránh trường hợp thiếu dòng hay sụt áp sẽ làm ESP8266 khởi động lại chương trình.

Nguồn cung cấp cho ESP8266 hoạt động là từ 3~3.6VDC xét về mức dòng và áp thì IC ổn áp tuyến tính AMS1117 phù hợp với ESP8266. Vì AMS1117 tạo ra điện áp 3.3VDC và dòng có thể tạo ra gần 800mA.

Để ngõ ra nguồn 3.3 VDC ổn định và phẳng hơn em sử dụng tụ phân cực 100uF lọc nguồn bởi vì để ESP8266 hoạt động theo chế độ pwm khi không truyền nhận dữ liệu nhưng vẫn duy trì kết nối để tiết kiệm năng lượng thì ESP8266 sẽ chạy ở chế độ Modem-sleep với dòng tiêu thụ 15mA. Ta có thông số sau:

$$\text{Độ gợn sóng nguồn vào } r\% = 6\% = 0.06 \quad (3.1)$$

$$\text{Tần số } f = 2 \times f_0 = 2 \times 50 = 100 \quad (3.2)$$

$$R_t = \frac{U}{I_t} = \frac{3,3}{0,015} = 220\Omega \quad (3.3)$$

Công thức tính điện dung của tụ sau:

$$C = \frac{1}{4\sqrt{3.f.R_t.r\%}} = \frac{1}{4\sqrt{3.100.220.0,06}} = 109 \text{ uF, vậy ta chọn } 100\text{uF} \quad (3.4)$$

Tụ không phân cực 104 để lọc cao tần.

Do các đường truyền dữ liệu ở mạch nạp CP2102 có ngõ ra điện áp 5 VDC vì vậy để an toàn cho ESP8266 phải chắn diode zener 3.3VDC. Để hạn chế dòng cao chạy thẳng vào ESP8266 ở các đường truyền dữ liệu phải được nối tiếp với trở 1.8K.

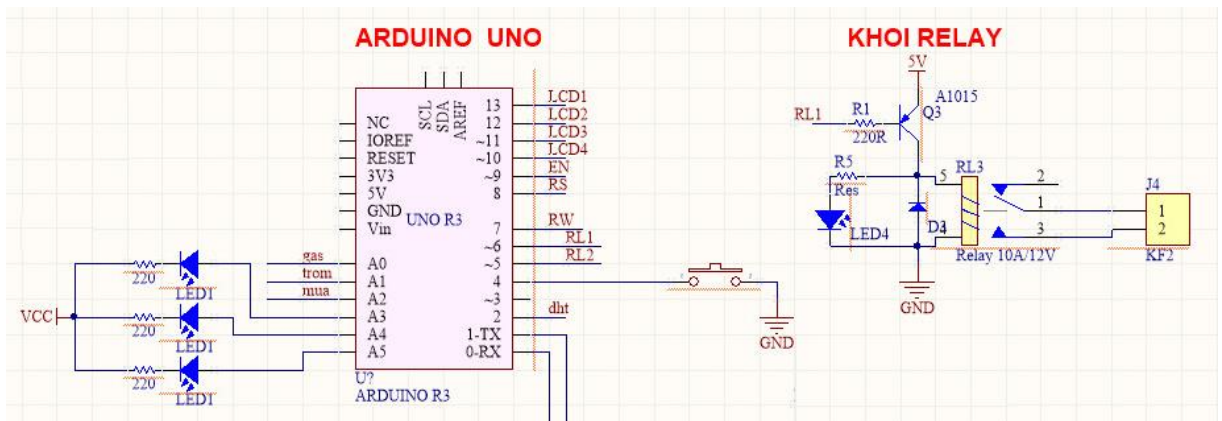
Sử dụng trở 1.8k bởi vì theo datasheet ở chế Standby dòng tiêu thụ khoảng 0.9mA để đảm bảo an toàn và tiết kiệm năng lượng trong quá trình nạp chương trình.

$$\text{Ta có: } R = \frac{V_{in} - V_{out}}{I} = \frac{5 - 3,5}{0,9} = 1.88k, \text{ chọn } 1k8 \Omega \quad (3.5)$$

Theo datasheet mức điện áp để ESP8266 hiểu là mức cao là từ $0.75V_{CC}$ $\Rightarrow V_{CC} + 0.3$. để tiết kiệm năng lượng và ESP8266 hoạt động an toàn dòng bé hơn 0.9mA và điện áp mức cao là 2V vậy ta có:

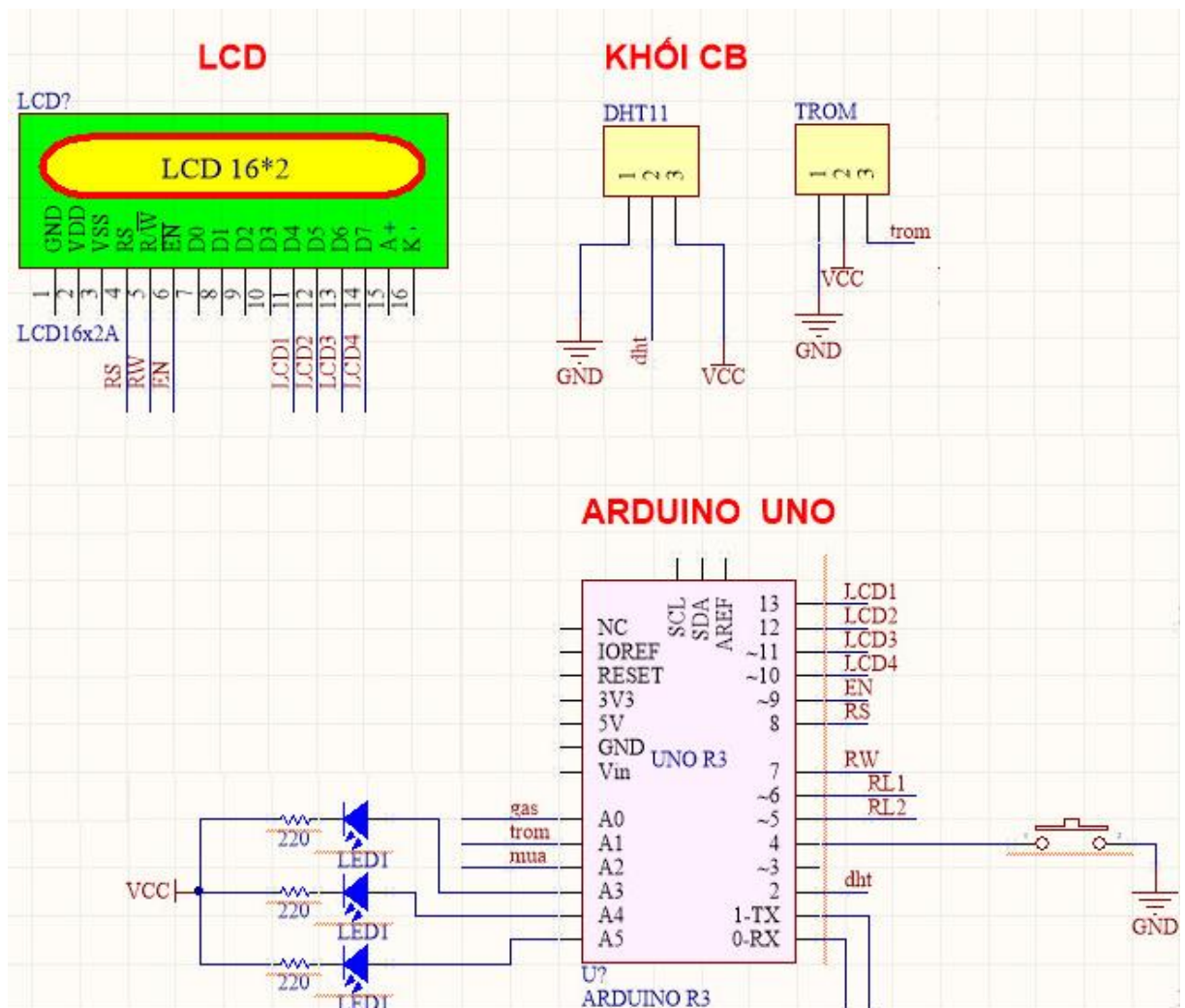
$$R = \frac{V_{cc} - V_{logic}}{I} = \frac{3,3 - 2}{0,3mA} = 4.333k, \text{ chọn } R = 4k7\Omega \quad (3.6)$$

Sơ đồ nguyên lý mạch xử lý trung tâm điều khiển đèn hoặc quạt (ESP8266 ESP-12E NodeMCU)



Hình 3.4 Sơ đồ nguyên lý mạch xử lý trung tâm điều khiển bật tắt thiết bị

Tương tự, ta có sơ đồ nguyên lý mạch xử lý trung tâm của mạch giám sát nhiệt độ, độ ẩm và chuyển động

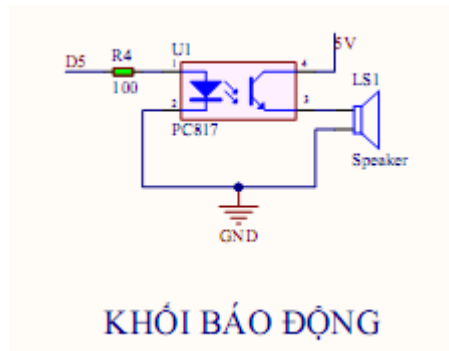


Hình 3.5 Sơ đồ nguyên lý mạch xử lý trung tâm giám sát nhiệt độ, độ ẩm và chuyển động

b) Thiết kế mạch báo động

Điện áp hoạt động của Buzzer là 5V, trong khi điện áp ra của ESP8266 ESP-12E NodeMCU chỉ được 3.3V, do đó ta cần dùng thêm opto PC817 để cấp nguồn 5V cho Buzzer. Chân xuất tín hiệu ra là GPIO14. Em chọn opto PC817 vì để đảm bảo cách ly về điện giữa khối nguồn và khối xử lý trung tâm và opto vẫn hoạt động tốt với áp và dòng điện trong mạch. Ta có: dòng lớn nhất mà ESP xuất ra là $I_{\max} = 12\text{mA}$, điện áp mức cao do ESP xuất ra là $0.8 \times V_{\text{dd}} = 0.8 \times 3.3 = 2.64\text{V}$, điện áp cho phép đặt vào diode quang của opto VF từ 1.2-1.4V. Do đó ta chọn điện trở:

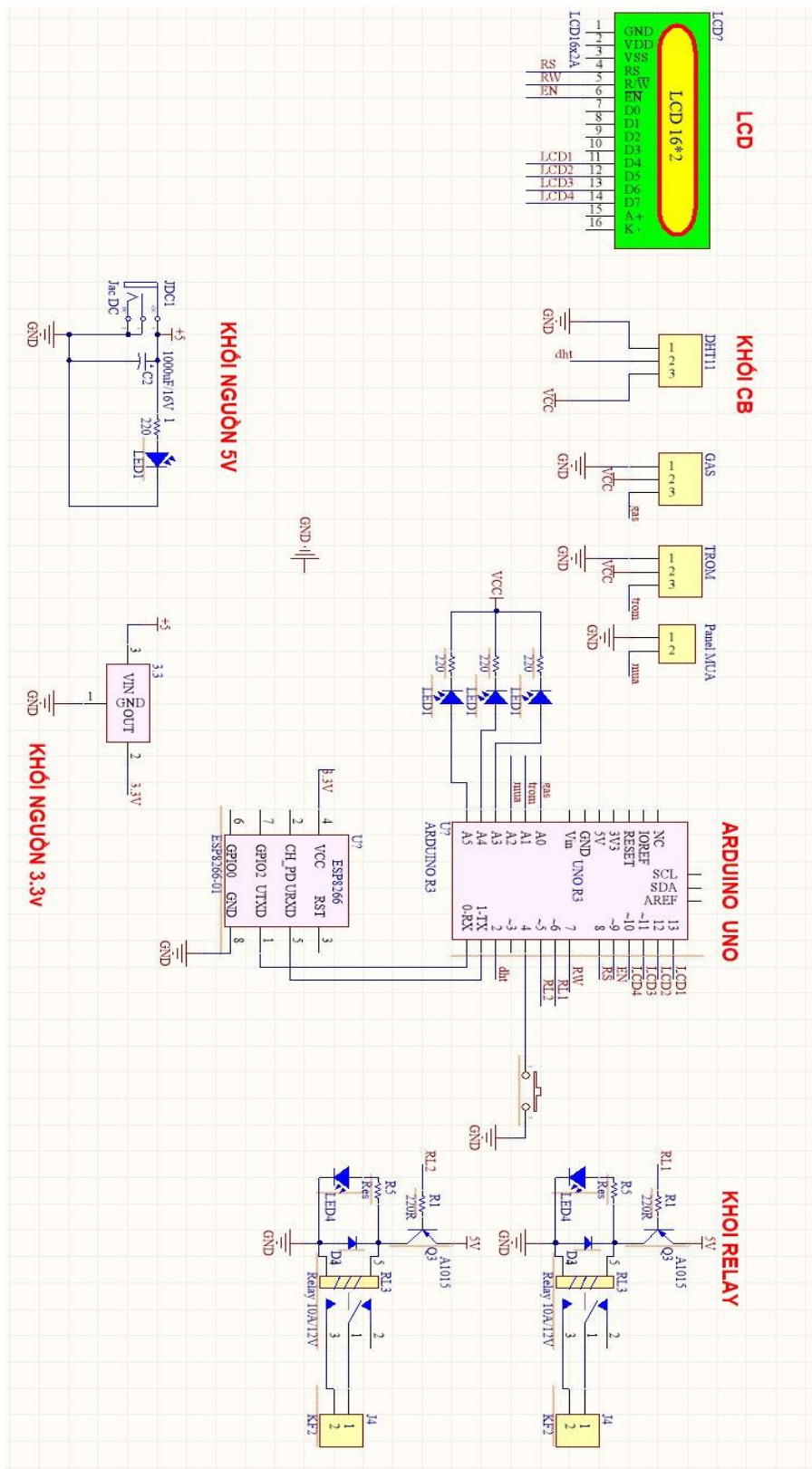
$$R = \frac{V_{0\max} - V_f}{I_{\max}} = \frac{2,64 - 1,4}{12\text{mA}} = 103,33 \, \Omega, \text{ chọn } R = 100\Omega \quad (3.7)$$



Hình 3.6 Sơ đồ kết nối mạch báo động chống trộm

c) Sơ đồ nguyên lý toàn mạch

Đối với mạch điều khiển độ sáng đèn bulb sơ đồ nguyên lý toàn mạch bao gồm hai khối chính: Khối xử lý trung tâm (ESP ESP8266 ESP-12E NodeMCU), khối CB. Như sơ đồ bên dưới IC AMS1117 tạo nguồn 3.3VDC cấp cho ESP8266 hoạt động.



Hình 3.7 Sơ đồ kết nối toàn mạch**3.3. Phần mềm chạy trên hệ điều hành Blynk****a) Yêu cầu thiết kế**

- Hiển thị được trạng thái bật/tắt các thiết bị như đèn quạt.
- Hiển thị được nhiệt độ, độ ẩm hiện tại.
- Cảnh báo cháy, trộm thông qua App.
- Điều khiển bật/ tắt thiết bị thông qua các switch.
- Điều khiển bật tắt thiết bị qua giọng nói.
- Bật chế độ an toàn khi đi ngủ.

b) Thiết kế phần mềm

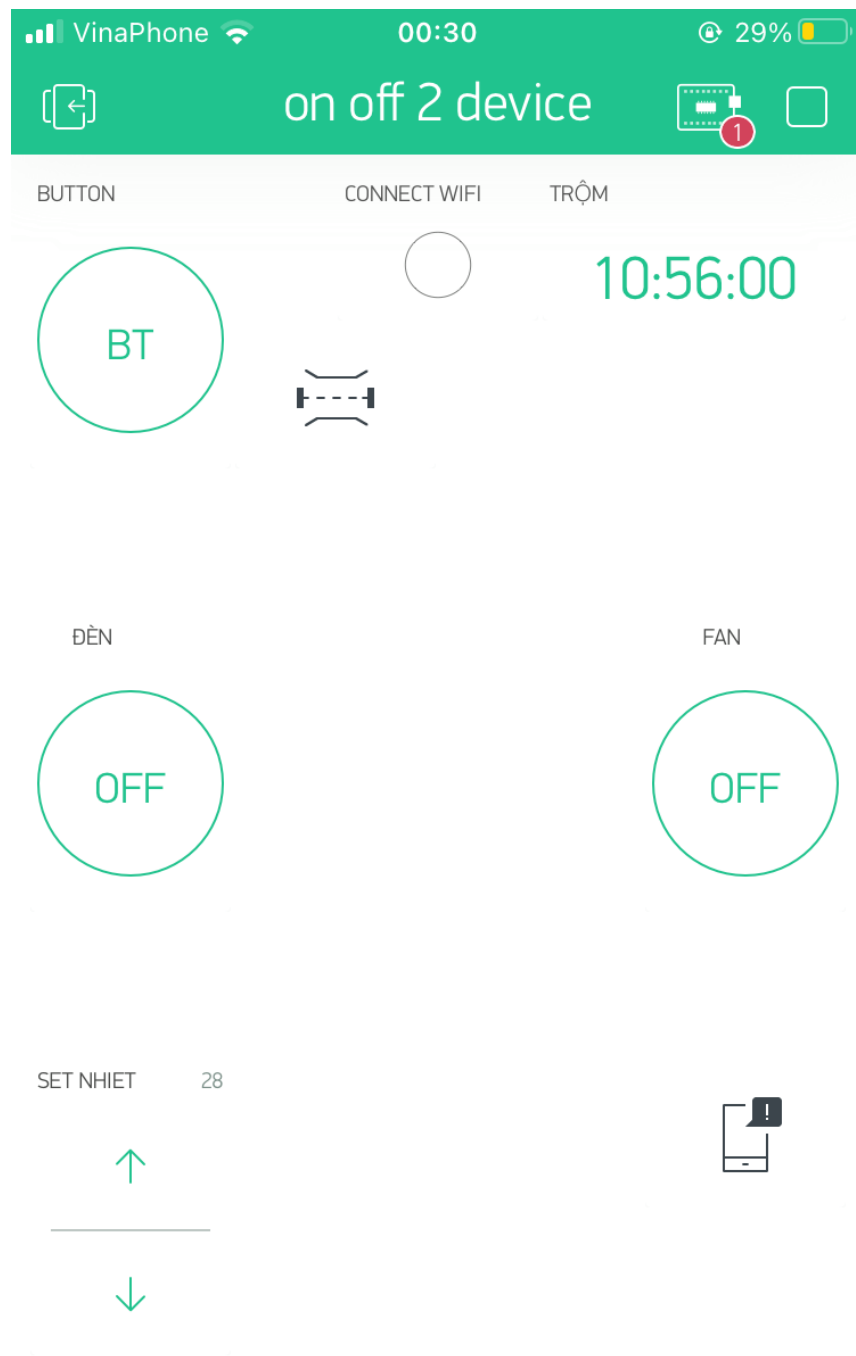
- Tổng quan về phần mềm

Phần mềm thiết kế có giao diện gồm giao diện chức năng 1 sẽ hiển thị trạng thái của đèn, quạt. Ở giao diện này có thể điều khiển được đóng mở đèn bằng các Switch.

Giao diện chức năng thứ 2 hiển thị giá trị nhiệt độ, độ ẩm, hiển thị cảnh báo cháy, trộm thông qua các hình ảnh giúp người dùng dễ dàng quan sát.

Giao diện chức năng thứ 3 chế độ an toàn khi ngủ dùng cài đặt thời gian bật và tắt các thiết bị.

Thiết kế phần mềm Ban đầu khi khởi động phần mềm sẽ có giao diện chính như sau:



Hình 3.8 Giao diện hiển thị và điều khiển độ sáng đèn, quạt, nhiệt độ, chuyển động

3.4. Kết luận

Chương 3 đã đưa ra được sơ đồ nguyên lý hoạt động của các khối cũng như hệ thống toàn mạch của ngôi nhà. Tạo ra những ứng dụng trên phần mềm để điều khiển ngôi nhà bằng mạng LAN cũng như 4G/LTE.

CHƯƠNG 4: TỐI ƯU KỊCH BẢN VÀ XÂY DỰNG MÔ HÌNH NHÀ THÔNG MINH

Trong chương này sẽ trình bày về các bước kịch bản tối ưu trong nhà; thiết kế mạch thực tế; xây dựng mô hình hoàn chỉnh

4.1. Kịch bản tối ưu cho một ngôi nhà thông minh

Sống trong nhà thông minh có gì khác sống trong nhà bình thường? Sau đây em xin giới thiệu kịch bản được thiết lập cho cả ngày đi làm của một ngôi nhà thông minh.

➤ 5:30 SÁNG

- Đèn hành lang nhà bạn tự động tắt ngay khi ánh mặt trời đầu tiên xuất hiện ở chân trời.
- Màn cửa tự động kéo lên cho nắng sớm chiếu sáng phòng ngủ nhà bạn.
- Máy lạnh phòng gym và máy nước nóng trong phòng tắm tự động bật.

➤ 6:30 SÁNG

- Sau khi tập thể dục và tắm xong, bạn xuống nhà bếp và rót cafe đã được máy cafe tự pha chế theo giờ đã thiết lập riêng biệt cho bạn và vợ.

➤ 7:30 SÁNG

- Sau khi ăn sáng, bạn vào phòng làm việc. Chỉ cần chạm vào 1 nút nhỏ trên Ipad hoặc điện thoại thông minh để chọn chế độ “LÀM VIỆC”:
 - Đèn tự bật.
 - Máy lạnh tự khởi động.
 - Tivi tự bật và bắt các kênh tin tức.
 - Máy laptop của bạn cũng sẵn sàng làm việc.

➤ 9:00 SÁNG

- Bạn vào garage và lên xe. Khi bạn khởi động xe, cửa garage mở và phòng làm việc chuyển sang trạng thái “NGỪNG LÀM VIỆC”:
 - Đèn tắt.
 - Máy lạnh ngưng hoạt động.
 - TV tắt.
 - Máy tính chuyển sang trạng thái nghỉ.
 - Hệ thống an ninh ngôi nhà được bật lên
 - Hệ thống tưới tiêu vườn cỏ chạy tự động theo thiết lập

➤ 5:00 CHIỀU

- Sau một ngày làm việc mệt mỏi và căng thẳng, bạn chỉ muốn về nhà nghỉ ngơi thư giãn. Ngôi nhà sẽ chuyển trạng thái “CHÀO ĐÓN VỀ NHÀ”:
 - khi đó cửa sẽ tự động mở.
 - các vị trí đèn cần thiết (đèn vườn, đèn lan can) trong nhà sẽ tự bật sang.
 - màn cửa được kéo lên.
 - máy điều hòa.
 - máy nóng lạnh sẽ tự khởi động.
 - nhạc sẽ tự động phát ra những bản nhạc bạn yêu thích.

➤ 7:00 TỐI

- Bữa tối đã hoàn thành, bạn không còn tất bật chuẩn bị nữa. Chỉ cần chạm nhẹ vào kịch bản “BỮA TỐI”, ngay lập tức toàn bộ không gian của phòng ăn sẽ chuyển sang chế độ phù hợp:
 - Hệ thống đèn sáng lên,
 - Hệ thống âm thanh đa vùng sẽ phát lên những bản nhạc du dương, màn cửa tự động điều chỉnh...tạo nên bữa ăn ấm cúng, thoải mái, dễ chịu cho gia đình bạn.

➤ 8:00 TỐI

- Sau bữa cơm gia đình vui vẻ, sẽ chính là thời gian vui vẻ sum họp của cả gia đình sau một ngày làm việc và học tập mệt mỏi, ngôi nhà sẽ chuyển sang trạng thái “GIA ĐÌNH SUM HỌP”:
 - Hệ thống đèn phòng khách (phòng chung) sáng lên.
 - Điều hòa nhiệt độ tăng dần nhiệt độ do trời đêm thường lạnh
 - Bật máy lọc không khí
 - Tivi bật chương trình hoạt hình hoặc những bộ phim yêu thích

➤ 11:30 TỐI

- Trước khi lên giường, bạn chọn chế độ “ĐI NGỦ”:
 - Hệ thống đèn trong phòng tắt, còn đèn ngủ sẽ được bật
 - Nhiệt độ điều hòa tự điều chỉnh ở mức thích hợp đã thiết lập.
 - Hệ thống âm thanh tự động bật nhạc du dương dễ ngủ
 - Hệ thống an ninh được bật lên và
 - Màn cửa kéo xuống.

4.2. Thi công hệ thống

Sau khi hoàn thành công đoạn thiết kế nhóm chúng em tiến hành việc thi công lắp ráp đèn led, quạt, lắp ráp mạch giám sát nhiệt độ, độ ẩm và chuyển động và phần mềm ứng dụng như sau:

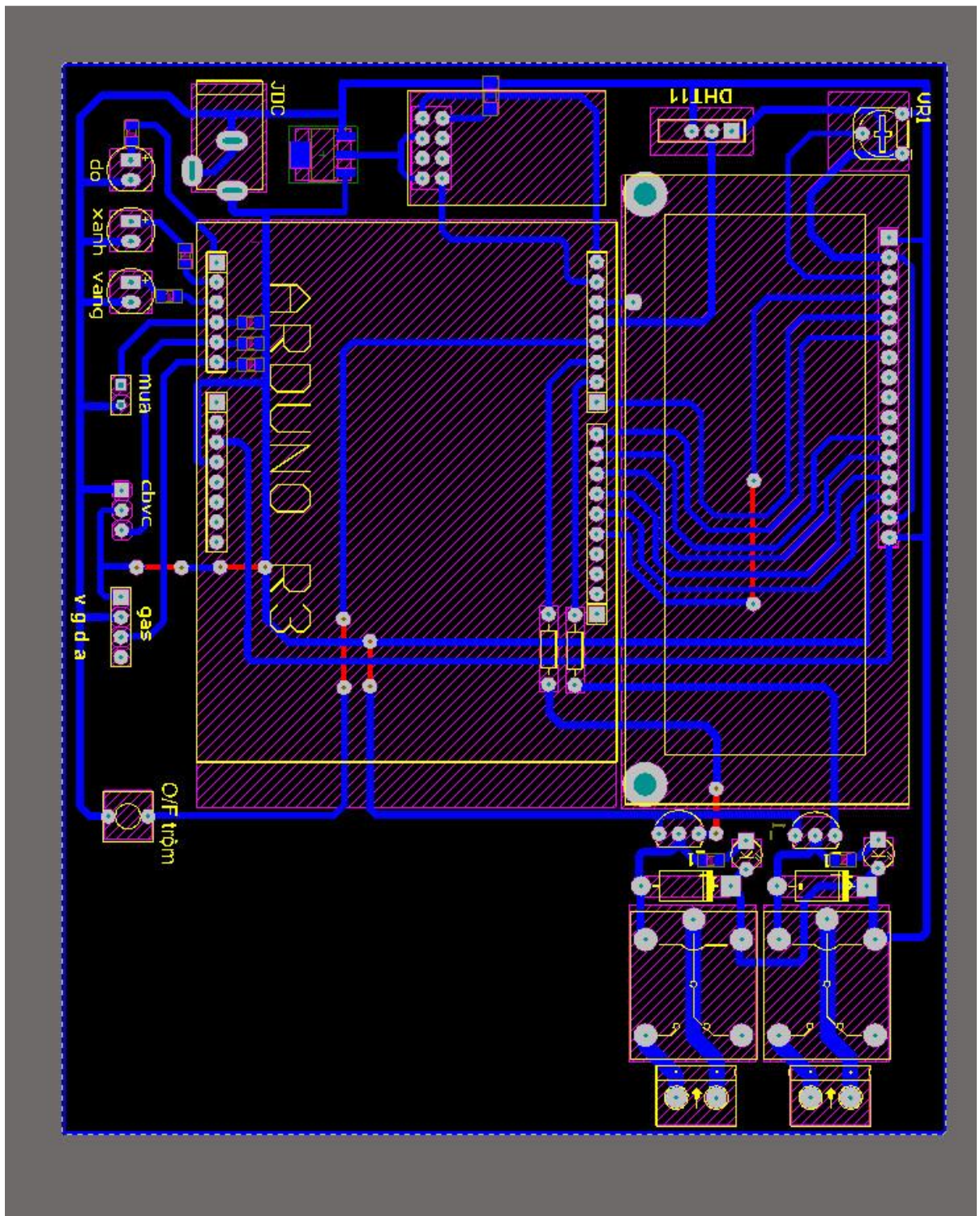
- Thi công hệ thống:
 - Thi công phần cứng.
 - Lắp ráp và kiểm tra.
- Lập trình mạch điều khiển bật tắt thiết bị.
- Lập trình mạch giám sát nhiệt độ, độ ẩm và chuyển động.
- Lập trình điều khiển bằng giọng nói.
- Viết tài liệu hướng dẫn, thao tác.

4.2.1. Thi công phần cứng:

Bảng 4.1 Danh sách các linh kiện sử dụng

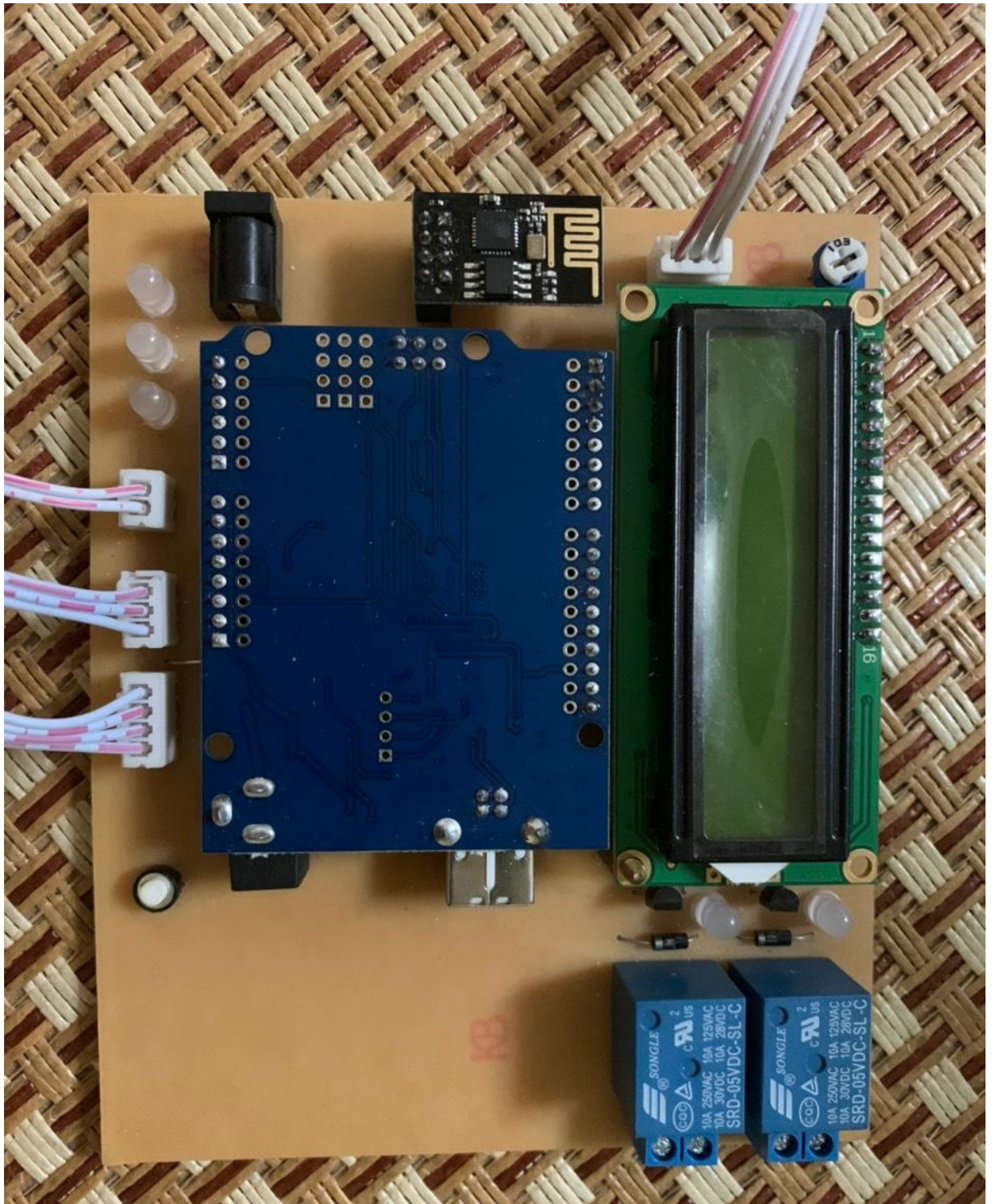
STT	Tên linh kiện	Giá trị	Số lượng	Chú thích
1	Mạch Arduino R3	Điện áp hoạt động: 5V	1	Linh kiện dán
2	Tụ 100uF	Điện dung 100uF, điện áp 25V	2	Linh kiện dán
3	Điện trở 10k Ω	Giá trị điện trở 10k Ω , công suất 1/8W	2	Linh kiện xuyên lỗ
4	ESP8266	Điện áp 3V-3,6V. Dòng tiêu thụ 80mA	1	Linh kiện dán
5	Cảm biến DHT11	Điện áp hoạt động 3V-5V. Dòng tiêu thụ 2,5mA	1	Linh kiện xuyên lỗ
6	Led xanh lá, đỏ, vàng	Điện áp hoạt động 3V-3,4V. Dòng tiêu thụ 10mA	2	Linh kiện xuyên lỗ
7	Cảm biến mưa	Điện áp hoạt động: 5V	1	Linh kiện dán
8	Module Relay	Điện áp hoạt động 3V-5V. Dòng tiêu thụ 2,5mA	2	Linh kiện xuyên lỗ

1) Sơ đồ mạch in của mạch ngôi nhà thông minh



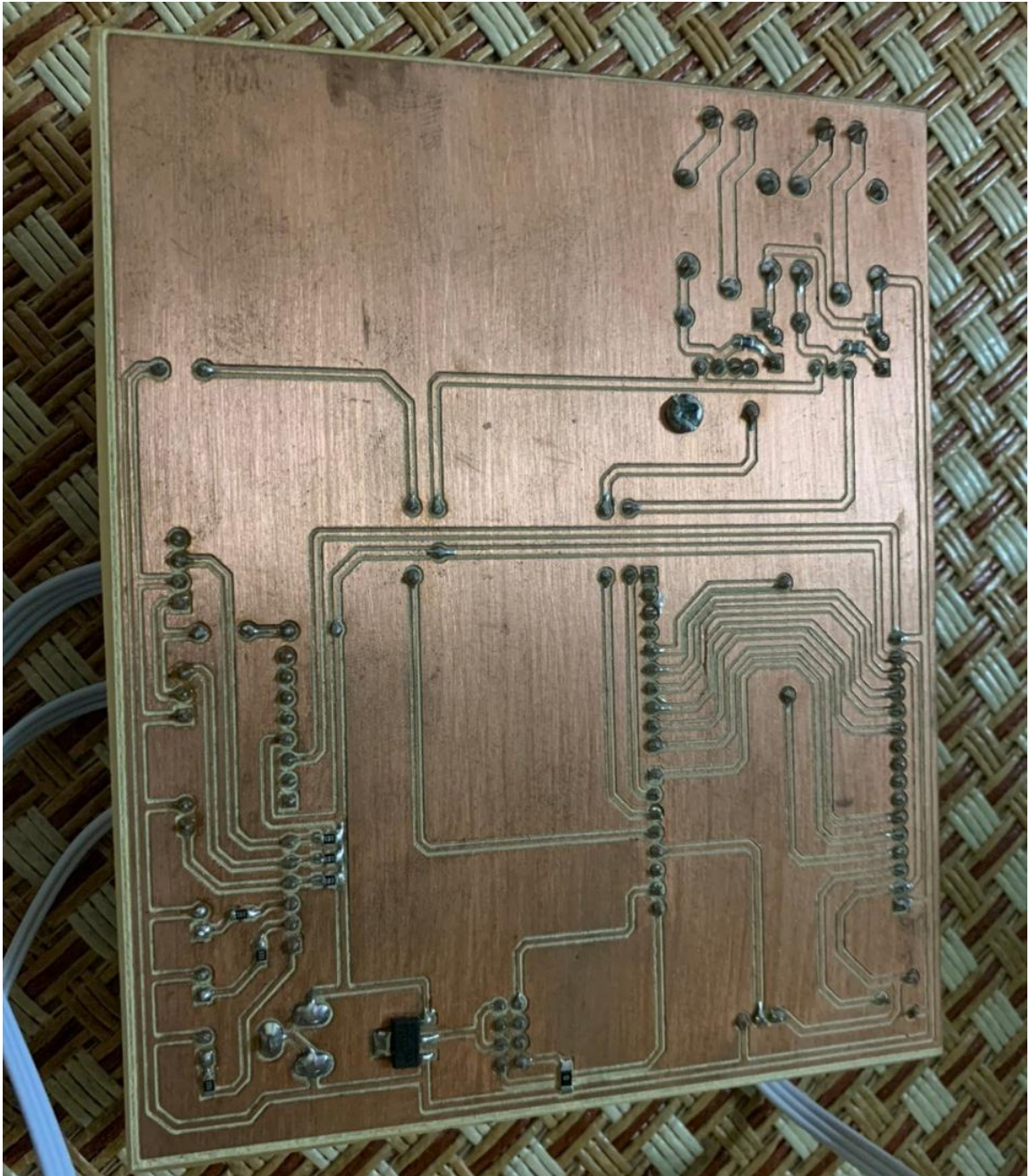
Hình 4.3 Sơ đồ mạch in

2) Mặt trước của mạch in



Hình 4.3 Mặt trước mạch in thực tế

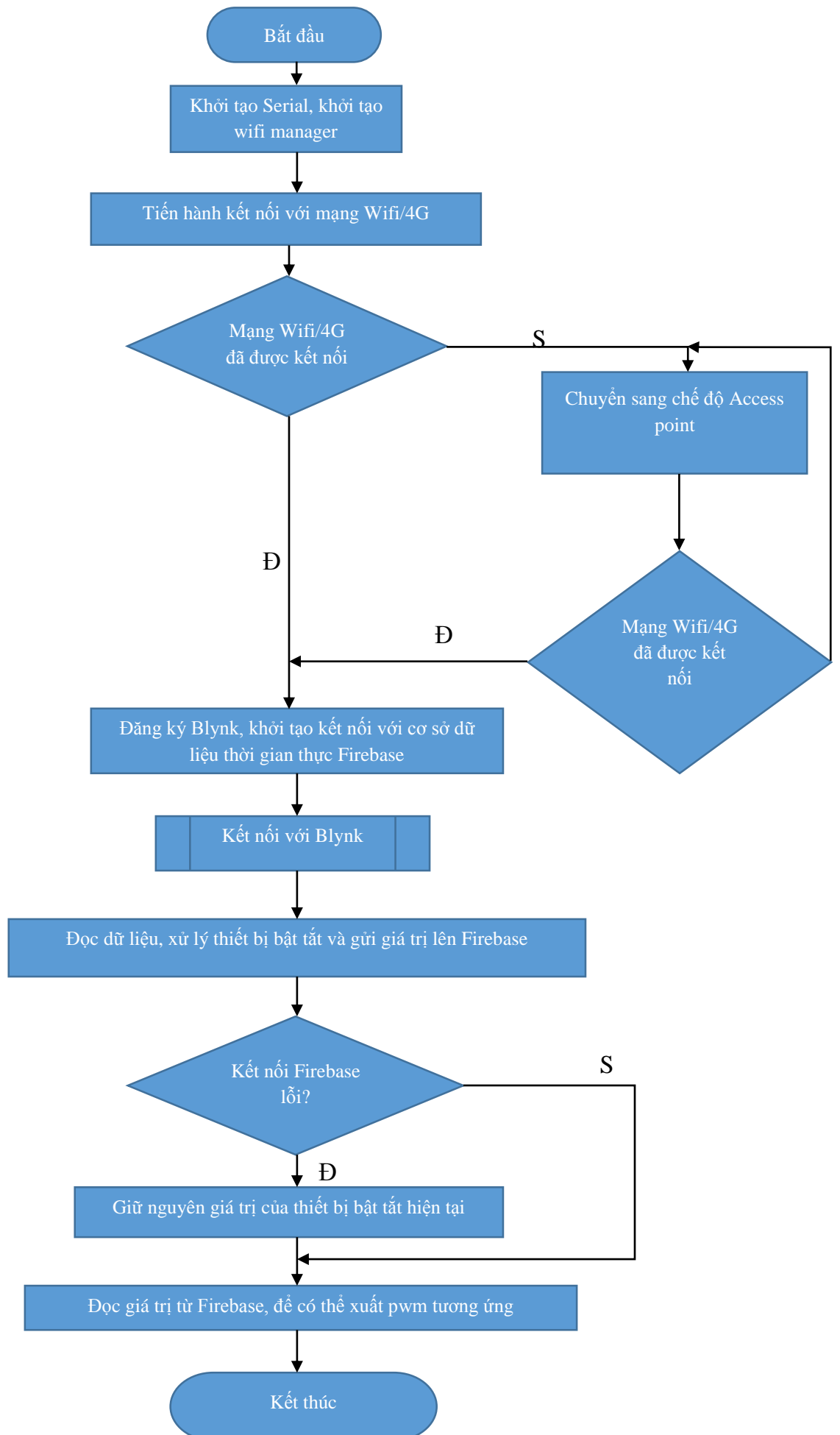
3) Mặt sau của mạch in



Hình 4.4 Mặt sau mạch in thực tế

4.2.2. Lưu đồ giải thuật mạch điều khiển bật tắt thiết bị

Hình 4.5: Lưu đồ giải thuật bật tắt thiết bị



- Nguyên lý hoạt động của hệ thống:

- Điều khiển bằng giọng nói với Google Assistant và Google Home mini

Khi ta nói một câu lệnh trên Google Assistant đã được cài đặt sẵn để bật tắt một thiết bị thì Google Assistant sẽ nhận lệnh sau đó thông qua IFTTT truyền xuống Blynk sau đó từ Blynk truyền lệnh xuống NodeMCU Lua ESP8266. Tại đây NodeMCU Lua ESP8266 sẽ xử lý dữ liệu để kích relay tương ứng với các mức logic để bật tắt thiết bị theo câu lệnh đã cài đặt trên Google Assistant

- Điều khiển bằng Blynk:

Khi có tín hiệu điều khiển từ app Blynk về khối trung tâm thì NodeMCU Lua ESP8266 sẽ xử lý dữ liệu nhận được để kích các relay tương ứng để điều khiển thiết bị điện cũng như đồng bộ trạng thái lên công tắc cảm ứng.

- Chương trình cho mạch điều khiển thiết bị bật tắt

Chương trình nhóm chúng em viết cho mạch điều khiển thiết bị bật tắt gồm các tính năng như sau:

- Thứ 1: Các thiết bị đèn hoạt động độc lập với nhau, người dùng có thể tùy chỉnh riêng biệt từng đèn trên app hoặc thông qua Google assistant.
- Thứ 2: Các thiết bị như đèn, quạt có thể kết nối với mạng wifi bất kì, khi chưa kết nối được thì thiết bị đèn chuyển sang chế độ Access point (với tên wifi không trùng lặp) để người dùng kết nối với đèn và truy cập địa chỉ "192.168.4.1" để thiết lập wifi cho nó, sau khi thiết lập xong đèn sẽ chuyển sang chế độ station để kết nối wifi mà người đã dùng thiết lập.
- Thứ 3: Khi đèn đang hoạt động mà wifi bị lỗi hoặc mất kết nối thì đèn vẫn giữ nguyên (sáng hoặc không sáng), thiết bị đèn sẽ chờ kết nối, khi kết nối được sẽ hoạt động lại bình thường.
- Thứ 4: Các trạng thái của đèn (sáng hoặc không sáng) sẽ được cập nhật lên cơ sở dữ liệu và phần mềm điện thoại sẽ cập nhật được trạng thái thực tế để giúp người dùng nhận biết.
- Thứ 5: Khi đèn hoạt động thì dữ liệu sẽ được cập nhật lên cơ sở dữ liệu và khi không hoạt động thì sẽ không cập nhật dữ liệu lên cơ sở dữ liệu nhờ dữ liệu này

mà ứng dụng trên điện thoại sẽ phân tích và báo cho người dùng biết thiết bị thực tế có đang hoạt động hay không.

- Thứ 6: Đèn có thể bật tắt và thay đổi độ sáng bằng giọng nói thông qua trợ lý ảo Google assistant mà không cần bật ứng dụng.

Do phần chương trình bao gồm một số tính năng nên chương trình nhiều dòng lệnh và em sẽ giải thích chương trình chính của chương trình lập trình cho vi điều khiển ESP8266 ESP-12E NodeMCU trong mạch điều khiển.

Để tiến hành lập trình thì cần thêm các bộ thư viện để sử dụng các tính năng. Thêm thư viện để thực hiện được các tính năng wifi của mạch vi điều khiển ESP8266 ESP-12E NodeMCU.

```
#include <ESP8266WiFi.h>
```

Thêm thư viện để thực hiện việc đọc ghi dữ liệu lên cơ sở dữ liệu thời gian thực FIREBASE.

```
#include <FirebaseArduino.h>
```

Thêm thư viện để thực hiện việc kết nối Adafruit_Mqtt để nhận dữ liệu điều khiển bằng Google assistant.

```
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
```

Thêm thư viện để thực hiện việc kết nối wifi tự động.

```
#include <WiFiManager.h>
```

Để tạo kết nối với cơ sở dữ liệu thời gian thực cần có địa chỉ và mã cho người phát triển, 2 thông tin này sẽ được gán vào 2 biến tương ứng là FIREBASE_HOST, FIREBASE_AUTH thông qua định nghĩa:


```
#define FIREBASE_HOST "testdatn-12cd4.firebaseio.com"

#define                                     FIREBASE_AUTH
"HszRBsnMPCl1Yc05DeHtErXk7lhNoVk8KJ10swB1"
```

Để kết nối với Adfruit thì cần có các thông số sau: địa chỉ sever kết nối, port kết nối, tên tài khoản, mật khẩu với định nghĩa:

```
#define MQTT_SERV "io.adafruit.com"

#define MQTT_PORT 1883

#define MQTT_NAME "masterpoke"

#define MQTT_PASS "7b5c65d02b14449f945c55acfd4d20fe"
```

Để thực hiện việc điều khiển ta cần liên kết với các feed được tạo từ trước trên Adfruit.

```
WifiClient client;

Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT,
MQTT_NAME, MQTT_PASS);

Adafruit_MQTT_Subscribe led1 = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/led1");

Adafruit_MQTT_Subscribe led2 = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/led2");

Adafruit_MQTT_Subscribe fan = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/fan");

Adafruit_MQTT_Subscribe fanmax = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/fanmax");
```

Định nghĩa chân tạo PWM và khởi tạo các biến lưu giá trị

```
#define LEDpin D1
```

```
uint16_t a=0;

int8_t led1;
```

Trong chương trình khởi tạo:

```
WiFiManager w;

w.autoConnect("led 1");

Serial.println("connected...");
```

Khởi tạo wifimanager, thực hiện tự động kết nối wifi, khi chưa kết nối được wifi thì tự chuyển sang chế Access point với tên wifi “led 1”, sau khi kết nối được thì in ra “connected”. Với các câu lệnh:

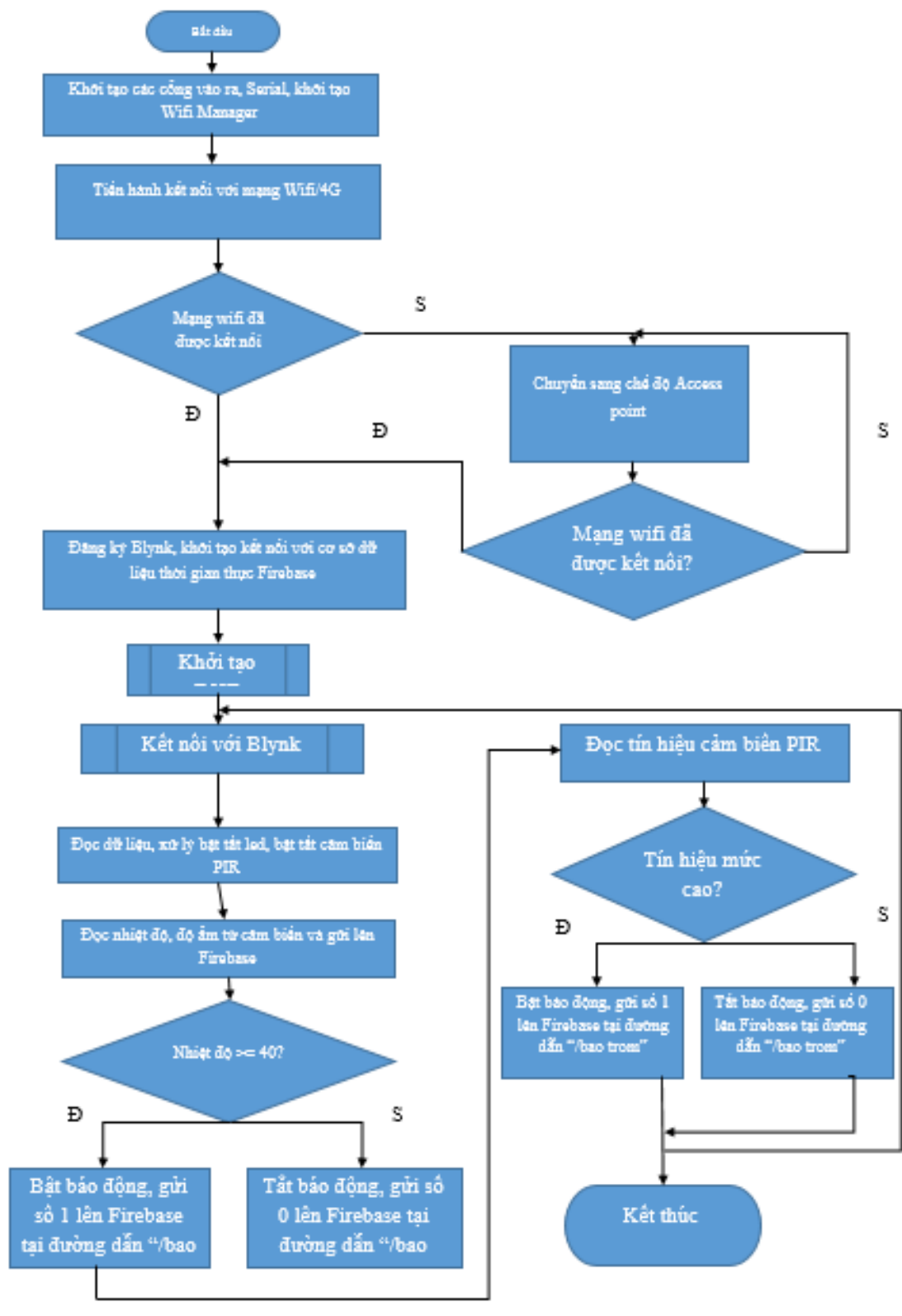
```
mqtt.subscribe(&ds1);

    mqtt.subscribe(&led1);

    mqtt.subscribe(&led2);

    mqtt.subscribe(&fanmax);
```

4.2.3. Lưu đồ giải thuật mạch giám sát nhiệt độ, độ ẩm và chuyển động.



Hình 4.6 Lưu đồ giám sát nhiệt độ, độ ẩm và chuyển động.

Khi có nguồn cung cấp cho mạch vi điều khiển ESP8266 ESP-12E NodeMCU sẽ hoạt động khởi tạo Serial, Wifi Manager, các cổng vào ra. Tiếp theo nó sẽ tiến hành kết nối với Wifi, nếu kết nối được thì tiến hành đăng kí Blynk và khởi tạo kết nối Firebase, trường hợp không kết nối được thì nó sẽ chuyển sang chế độ Access point (điểm truy cập) và đợi người dùng kết nối với Wifi do nó phát ra và thiết lập Wifi cho nó tại địa chỉ 192.168.4.1. Sau khi người dùng thiết lập xong ESP sẽ chuyển sang chế độ Station (thu Wifi) và kết nối Wifi, nếu vẫn chưa kết nối được thì nó vẫn ở chế độ Access point. Tiếp theo ESP khởi tạo DHT.

Trong vòng lặp tuần hoàn:

Chương trình Blynk sẽ nhận được thông báo xử lý dữ liệu để bật tắt cảm biến PIR hoặc đèn led. Sau đó đọc giá trị nhiệt độ, độ ẩm từ cảm biến và gửi các giá trị đó lên Firebase. Nếu nhiệt độ lớn hơn hoặc bằng 40 thì bật Buzzer báo động và đèn led đỏ sẽ hiện lên và thông báo ngôi nhà có hỏa hoạn, ngược lại tắt Buzzer và đèn led xanh hiện lên. Tiếp theo đọc tín hiệu PIR từ cảm biến chuyển động, nếu tín hiệu mức cao thì bật Buzzer báo động và gửi số 1 lên Firebase sẽ thông báo lên Blynk hệ thống có trộm, ngược lại tắt Buzzer báo động và gửi số 0. Chương trình cứ thế lặp lại.

- Chương trình cho mạch giám sát nhiệt độ, độ ẩm và chuyển động.

Chương trình em viết cho mạch giám sát nhiệt độ, độ ẩm và chuyển động gồm các tính năng như sau:

- Thứ 1: Thiết bị có thể kết nối với mạng wifi bất kì, khi chưa kết nối được thì thiết bị chuyển sang chế độ Access point (với tên wifi không trùng lặp) để người dùng dùng điện thoại kết nối với đèn và truy cập địa chỉ “192.168.4.1” để thiết lập wifi cho nó, sau khi thiết lập xong đèn sẽ chuyển sang chế độ station để kết nối wifi mà người đã dùng thiết lập.
- Thứ 2: Thiết bị có thể đo nhiệt độ, độ ẩm và kích hoạt cảm biến PIR khi được yêu cầu, cảnh báo khi vượt ngưỡng nhiệt độ đặt trước hoặc phát hiện chuyển động.

- Thứ 3: Các dữ liệu được đưa lên Firebase để điện thoại đọc về và báo lên app người dùng
- Thứ 4: Có thể bật tắt led trên vỏ hộp bằng phần mềm Blynk như báo nhiệt độ quá lớn so với cảnh báo

Sau đây nhóm em xin trình bày chương trình của hệ thống:

Thêm thư viện để thực hiện được các tính năng wifi của mạch vi điều khiển ESP8266 ESP-12E NodeMCU.

```
#include <ESP8266WiFi.h>
```

Thêm thư viện để thực hiện việc đọc ghi dữ liệu lên cơ sở dữ liệu thời gian thực FIREBASE.

```
#include <FirebaseArduino.h>
```

Thêm thư viện để thực hiện việc kết nối Adafruit_Mqtt để nhận dữ liệu điều khiển bằng Google assistant.

```
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
```

Thêm thư viện để thực hiện việc kết nối wifi tự động.

```
#include<WiFiManager.h>
```

Thêm thư viện của cảm biến DHT11: #include<DHT.h>

Để tạo kết nối với cơ sở dữ liệu thời gian thực cần có địa chỉ và mã cho người phát triển, 2 thông tin này sẽ được gán vào 2 biến tương ứng là FIREBASE_HOST, FIREBASE_AUTH thông qua định nghĩa:

```
#define FIREBASE_HOST "testdatn-12cd4.firebaseio.com"

#define FIREBASE_AUTH
"HszRBsnMPCl1Yc05DeHtErXk7IhNoVk8KJ10swB1"
```

Để kết nối với Adfruit thì cần có các thông số sau: địa chỉ sever kết nối, port kết nối, tên tài khoản, mật khẩu với định nghĩa:

```
#define MQTT_SERV "io.adafruit.com"

#define MQTT_PORT 1883

#define MQTT_NAME "masterpoke"

#define MQTT_PASS "7b5c65d02b14449f945c55acfd4d20fe"
```

Để thực hiện việc điều khiển ta cần liên kết với các feed được tạo từ trước trên Adfruit

```
WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, MQTT_SERV, MQTT_PORT,
MQTT_NAME, MQTT_PASS);

Adafruit_MQTT_Subscribe goout = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/goout ");

Adafruit_MQTT_Subscribe denngu = Adafruit_MQTT_Subscribe(&mqtt,
MQTT_NAME "/f/denngu");
```

Định nghĩa loại cảm biến và chân giao tiếp với cảm biến:

```
#define DHTTYPE DHT11

#define DHTPIN D4
```

Gán các chân giao tiếp và gán val = 0, pirState = LOW.

```
int powerPin = D7;    // chọn chân 3 NGUON
int inputPin = D2;    // chọn ngõ tín hiệu vào cho PIR
int pirState = LOW;   // Bắt đầu với không có báo động
int pinSpeaker = D5;  //chọn chân cho chuông khi có đột nhập
int val = 0;
```

Gán các tham số vào hàm: DHT dht(DHTPIN, DHTTYPE);

- Trong chương trình khởi tạo

Khai báo vào ra cho các chân:

```
pinMode(powerPin, OUTPUT);
pinMode(inputPin, INPUT);

pinMode(pinSpeaker, OUTPUT);
pinMode(D0,OUTPUT);
```

Khởi tạo wifimanager, thực hiện tự động kết nối wifi, khi chưa kết nối được wifi thì tự chuyển sang chế Access point với tên wifi “light 1”, sau khi kết nối được thì in ra “connected”.

```
WiFiManager w;

w.autoConnect("light 1");

Serial.println("connected...");
```

Thực hiện đăng kí Mqtt cho các địa chỉ.

```
mqtt.subscribe(&goout );

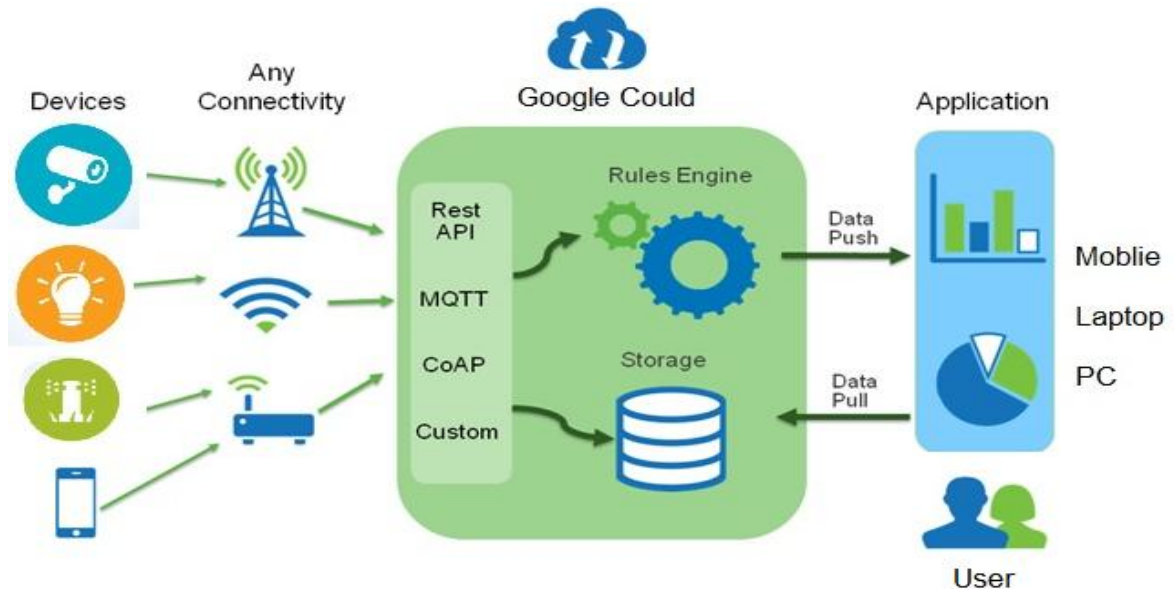
mqtt.subscribe(&denngu);
```

Thực hiện kết nối Firebase

```
firebase.initializeApp({
  apiKey: 'AIzaSyB...',
  authDomain: '...',
  databaseURL: '...',
  storageBucket: '...',
  messagingSenderId: '...'
});
```

Khởi tạo chương trình DHT: dht.begin();

4.2.4. Sơ đồ hệ thống nhà thông minh



Hình 4.7 Sơ đồ nguyên lý mạch thực tế

- Nguyên lý hoạt động

Để hệ thống hoạt động được trước tiên các thiết bị phải kết nối được với sóng Wifi. Riêng Google assistant, phần mềm điện thoại là các ứng dụng trên điện thoại nên yêu cầu điện thoại kết nối Wifi hoặc 3G.

Sau khi các thiết bị đã kết nối thành công với Wifi, việc điều khiển thiết bị sẽ thông qua điện thoại bằng 2 cách.

Cách 1: Ta sẽ sử dụng trợ lý ảo Google (Google Assistant) để ra lệnh điều khiển thiết bị bằng giọng nói. Dữ liệu đó sẽ được các thiết bị đèn nhận xử lý để điều khiển độ sáng đèn sau đó đưa dữ liệu đó lên Firebase, riêng mạch giám sát nhiệt độ, độ ẩm và chuyển động ta chỉ điều khiển bật tắt cảm biến chuyển động và đèn led ngủ.

Cách 2: Ta sẽ sử dụng đã App Android để điều khiển bằng cách gửi dữ liệu thông qua Firebase, bộ phận xử lý của đèn sẽ nhận dữ liệu đó và điều khiển đèn.

Ngoài tính năng điều khiển thì phần mềm còn có chức năng hiển thị các trạng thái của đèn, giá trị nhiệt độ, độ ẩm, cảnh báo cháy trộm thông qua việc đọc dữ liệu trên Firebase do bộ phận xử lý của đèn và bộ phận xử lý nhiệt độ, độ ẩm gửi lên.

4.3. Ứng dụng ngôi nhà hướng đến hệ sinh thái thành phố thông minh



4.3.1. Bảo mật

Bảo mật của ngôi nhà là vấn đề cấp thiết được đặt lên hàng đầu, nếu cơ sở hạ tầng có bất kỳ sai lệch nào trong hoạt động sẽ mang lại sự bất tiện và gây nguy hiểm đến cho người dùng. Vì vậy, an ninh là một mối quan tâm lớn trong ngôi nhà thông minh.

Cho nên em đã mã hóa dữ liệu truyền qua mạng, tăng độ bảo mật cũng như tính xác thực của mật khẩu người dùng, sử dụng các cảm biến có độ uy tín và liên kết các cảm biến đồng nhất để bảo vệ được ngôi nhà một cách tốt nhất. Hệ sinh thái Blynk đã hỗ trợ em rất nhiều trong vấn đề bảo mật này.

4.3.2. Cảm biến thông minh

Cảm biến thông minh là thành phần cứng thu thập dữ liệu của ngôi nhà. Việc triển khai ngôi nhà thông minh sẽ yêu cầu tất cả các thiết bị này trao đổi dữ liệu, thực hiện lịch lập trình cho các nhiệm vụ giữa chúng và có thể tổng hợp dữ liệu lại với nhau để đưa ra các suy luận. Giải pháp là phát triển và sử dụng các giao thức mở và định dạng dữ liệu sẽ cho phép các thiết bị giao tiếp giữa nhau và thúc đẩy hơn nữa việc triển khai hệ thống IoT. Tất cả đều được vi xử lý Arduino R3 và ESP8266 cải thiện.

Hơn nữa, cần đo lường và các phương án truyền dữ liệu công suất thấp, chi phí thấp mà còn phải phát triển tối ưu bộ nhớ và lưu trữ mới cũng như các thiết bị tiêu thụ điện thấp để kéo dài tuổi thọ của pin. Việc lưu trữ lượng lớn dữ liệu sẽ yêu cầu phát triển các thuật toán nén sẽ được sử dụng và các sơ đồ cơ sở dữ liệu trong tương lai. Các giải pháp cho các vấn đề điện năng đòi hỏi sự phát triển của các công nghệ pin mới.

4.3.3. Mạng kết nối

IoT phụ thuộc vào khả năng cảm biến và các thiết bị khác để có thể gửi và nhận thông tin cho nhau và dữ liệu đám mây. Với các phương pháp kết nối mạng hiện

tại như 4G/Wifi rất khả dụng trong thành phố thông minh. Nhiều thiết bị trong thành phố thông minh có các yêu cầu về tính di động và thông lượng dữ liệu ổn định để đáp ứng cung cấp chất lượng dịch vụ. Như vậy, mạng 5G chính là xu thế mà IoT cần hướng tới.

4.3.4. Phân tích dữ liệu thông minh

Các thiết bị kết nối IoT trong ngôi nhà để tăng cường sử dụng dữ liệu này và liên tục cải thiện các dịch vụ được cung cấp, các thuật toán phân tích dữ liệu mới cần được phát triển. Với vô số các tham số khác nhau được đo lường trong các thành phố thông minh, các thuật toán này cần được áp dụng cho dữ liệu có tính chất khác nhau, các kỹ thuật tổng hợp dữ liệu tốt hơn cần được phát triển cũng như kết hợp chúng theo những cách có ý nghĩa và có thể trích xuất các suy luận và nhận ra các mẫu. ‘Deep learning’ đã được quan tâm trong lĩnh vực này vì có thể tận dụng lượng lớn dữ liệu này để cung cấp kết quả tốt hơn cho các ứng dụng khác nhau. Nhờ vậy, mà các thiết bị và phần mềm có thể hỗ trợ một cách tốt nhất cho chính ngôi nhà cũng như cho thành phố thông minh.

KẾT LUẬN VÀ KIẾN NGHỊ

Việc thiết kế hệ thống điều khiển thiết bị và giám sát nhiệt độ, độ ẩm từ xa qua internet có ý nghĩa rất to lớn, có thể ứng dụng trong nhiều lĩnh vực của đời sống xã hội và trong công nghiệp. Ngoài ra, module wifi kết hợp với Arduino cũng mở ra rất nhiều hướng ứng dụng khác, phục vụ tốt cho việc học tập và nghiên cứu của sinh viên. Việc xây dựng hệ thống giám sát nhiệt độ, độ ẩm từ xa qua internet này cũng liên quan đến nhiều nền tảng kiến thức từ những kiến thức lý thuyết cho đến những kiến thức thực tiễn.

Những kết quả đạt được

- Tìm hiểu nguyên lý chung chuẩn truyền thông wifi do IEEE 802.11 quy định, tìm hiểu kiến thức về họ giao thức TCP/IP.
- Kết nối thành công mạch với server, demo điều khiển 4 thiết bị thông qua wifi.
- Thiết kế thành công mạch đo nhiệt độ, độ ẩm sử dụng cảm biến DHT11, và truyền lên server sau mỗi một phút.
- Lập trình giao diện web để đo lường, hiển thị thống kê các thông số nhiệt độ, độ ẩm và điều khiển cơ cấu chấp hành.

Đánh giá những kết quả đã đạt được

- Đo và hiển thị chính xác nhiệt độ, độ ẩm môi trường lên LCD và upload dữ liệu lên server ổn định. Kết quả đo là khá chính xác với sai số nằm trong phạm vi cho phép. Mạch hoạt động ổn định.
- Tuy nhiên, một số mặt hạn chế đó là trang web chưa tự động cập nhật lên giá trị mới phải refresh lại trang để cập nhật dữ liệu mới.
- Với kết quả bước đầu là kết nối truyền dữ liệu thành công qua mạng wifi và thực hiện đo lường và điều khiển một số thông số môi trường, điều khiển cơ cấu chấp hành để thay đổi như là quạt, điều hòa... với những kết quả demo như vậy đã khẳng định khả năng phát triển của đề tài.

Hướng phát triển của đề tài

Trong thời gian tới, em sẽ tiếp tục nghiên cứu phát triển đề tài theo hướng sau đây:

- Tăng tính chính xác và ổn định hơn nữa.
- Tích hợp nhiều cảm biến hơn nữa phù hợp với nhiều đối tượng người dùng để phục vụ cho cuộc sống và phục vụ trong công nghiệp thay thế các chuẩn ethernet, RS485...

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Phạm Duy Hưng, Luận văn thạc sĩ, *Điều khiển các thiết bị điện tử xa qua mạng Internet*, Trường Đại học Công nghệ, 2012.
- [2] Trần Quang Vinh, đề tài “*Nghiên cứu thiết kế, chế tạo các cấu kiện và hệ thống tự động hóa phục vụ giám sát, điều khiển, điều hành cho các tòa nhà cao tầng (nhà công ích và dân dụng)*”, mã số: KC.03.12/06-10
- [3] Tran Quang Vinh, Pham Manh Thang, Phung Manh Duong, “*Controlling Communication Network in the Building Automation System*,” Journal of Science, Vietnam National University, pp.129-140, Vol.26, 2010.
- [4] Trần Thị Hà, Trương Thị Bích Nga, Nguyễn Thị Lương,...*Giáo trình: Điện tử cơ bản*, Xuất bản: Đại học Quốc gia Tp. HCM, 2013.
- [5] Nguyễn Văn Hiệp – Đinh Quang Hiệp, “*Lập trình android cơ bản*”, Nhà xuất bản ĐH Quốc Gia, Tp.HCM, 2015

Tiếng Anh

- [6] Luigi Atzori, Antonio Iera, Giacomo Morabito, *Internet of Things: A survey*, Computer Networks 54 (2010) 2787–2805.
- [7] Ovidiu Vermesan, Peter Friess, Internet of Things – Converging Technologies for Smart Environments and Integrated Ecosystems, River Publishers Series in Communications.
- [8] Kiran Maraiya, Kamal Kant, Nitin Gupta, Application based Study on Wireless Sensor Network, International Journal of Computer Application (0975- 8887), Volume 21, No.8, May 2011.
- [9] Datasheet ESP8266 12EX
- [10] Datasheet OptoPC817
- [11] Datasheet IRF830
- [12] Datasheet DHT11
- [13] Datasheet PIR AM312