

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
Khoa Điện tử Viễn thông



ĐỒ ÁN 3
BỘ MÔN KỸ THUẬT MÁY TÍNH

Đề tài:

Hệ thống giám sát không khí, điều khiển thiết bị điện sử dụng công nghệ Internet of Things
áp dụng trong phòng học

GVHD: TS. Trần Thị Minh Hạnh

SVTH: Đào Minh Tâm

MSSV: 106170260

Gia Lai, tháng 08 năm 2021

LỜI CẢM ƠN

Thật vui và hạnh phúc vì em đã hoàn thành xong đề tài của đồ án này, em vẫn nhớ trước thời điểm nghỉ Tết Nguyên Đán 2021. Khi gặp cô để trao đổi đề tài, trong lòng em thật sự rất lo và thiếu ý tưởng. Em là sinh viên duy nhất chưa thực hiện một đồ án nào trước đây, chính vì vậy tâm trạng lại càng lo lắng hơn. Thế rồi dưới sự động viên, hướng dẫn của cô, cuối cùng một đề tài bắt đầu được em hoạch ra và bước vào thực hiện. Có thể nói, em thật sự may mắn khi có thêm ba tháng quý giá trong thời gian nghỉ dịch để nghiên cứu và tìm hiểu thêm về dự án này. Ba tháng đó đã giúp em đúc kết ra được rất nhiều thứ.

Em xin chân thành cảm ơn cô Hạnh thời gian qua đã giúp em hoàn thiện đồ án. Em cũng xin cảm ơn Nhà trường và thầy cô trong hội đồng bảo vệ đã tạo điều kiện để em được bảo vệ đồ án trong thời điểm dịch bệnh vẫn còn đang diễn biến rất phức tạp. Lời cuối em xin chúc thầy và cô luôn mạnh khỏe, bình an!

Gia Lai, tháng 8 năm 2021

Sinh viên



Đào Minh Tâm

MỤC LỤC

LỜI CẢM ƠN	2
TÓM TẮT ĐỒ ÁN.....	4
CHƯƠNG 1: GIỚI THIỆU	5
1.1 Tổng quan	5
1.2 Tính cấp thiết	6
1.3 Nhiệm vụ của đề tài	7
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	8
2.1 Chi tiết về Internet of Things.....	8
2.2 Cơ sở lý thuyết của hệ thống	9
CHƯƠNG 3: THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG.....	18
3.1 Yêu cầu thiết kế	18
3.2 Phân tích thiết kế.....	18
3.3 Sơ đồ khối tổng quát và giải thích	19
3.4 Tính toán và vẽ sơ đồ mạch chi tiết	20
Chương 4: THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM.....	21
4.1 Yêu cầu thiết kế	21
4.2 Phân tích thiết kế.....	21
4.3 Phân tích phần mềm.....	21
4.4 Lưu đồ thuật toán và giải thích	26
CHƯƠNG 5: KẾT QUẢ THỰC HIỆN	27
5.1 Kết quả thi công phần cứng	27
5.2 Kết quả thi công phần mềm	29
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	31
1. Kết luận	31
2. Hướng phát triển.....	31
TÀI LIỆU THAM KHẢO.....	33
PHỤ LỤC	34
1. Code.....	34
2. Phụ lục hình ảnh	43

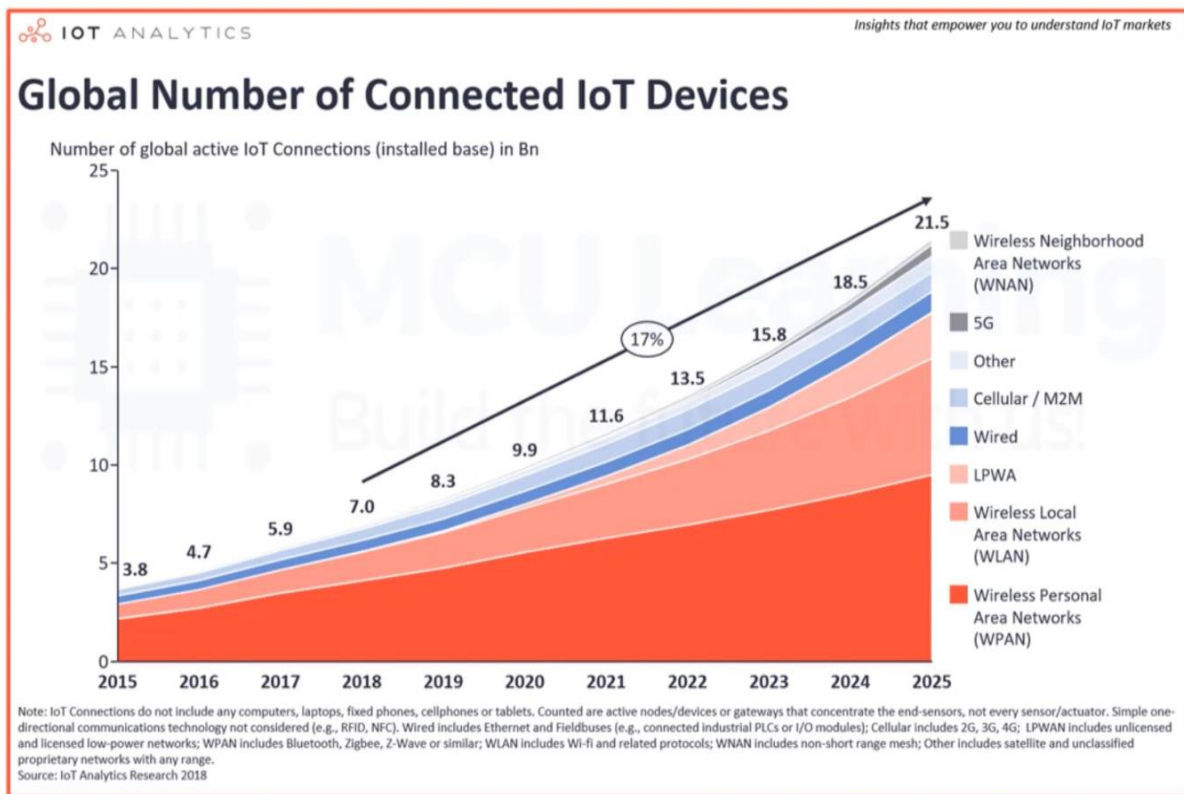
TÓM TẮT ĐỒ ÁN

Đồ án trình bày về giải pháp Internet of Things với ứng dụng điều khiển thiết bị điện và giám sát không khí áp dụng trong phòng học. Ý tưởng đặt ra là mọi thao tác điều khiển thiết bị điện cũng như các thông số về không khí sẽ được kiểm soát ngay trên chính chiếc điện thoại thông minh của bạn bởi một web server được thiết kế riêng. Người dùng có thể thao tác trên web server hoặc từ một ứng dụng mobile.

CHƯƠNG 1: GIỚI THIỆU

1.1. Tổng quan

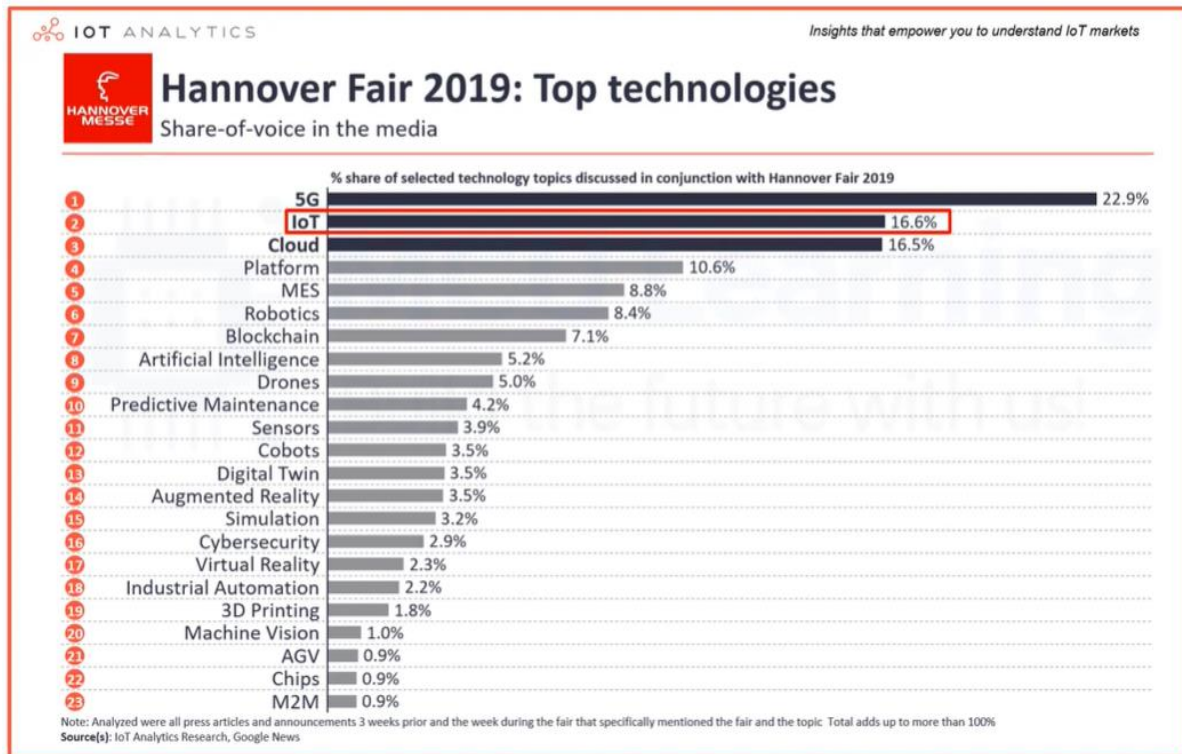
Theo thống kê của tổ chức IoT Analytics năm 2018 có khoảng 7 tỷ thiết bị IoT đã được sử dụng trên toàn cầu, tăng 18% so với năm 2017 và con số này dự kiến sẽ đạt 21.5 tỷ vào năm 2025. Đây là một tốc độ tăng trưởng cực kỳ nhanh.



Hình 1.1 Thống kê số thiết bị IoT từ năm 2015

Trong 7 tỷ thiết bị đó, hơn một nửa sẽ là các sản phẩm tiêu dùng như TV thông minh và loa thông minh. Các thiết bị IoT dành cho doanh nghiệp được sử dụng nhiều nhất là đồng hồ điện thông minh và camera an ninh thương mại.

Dưới đây là một thống kê nữa từ sự kiện Hannover Fair 2019 (*Sự kiện công nghệ hàng đầu thế giới*) nói về xếp hạng những xu thế công nghệ được triển lãm nhiều nhất tại sự kiện diễn ra trong năm 2019. IoT chiếm 16.6% cao thứ hai chỉ sau 5G, cho thấy xu thế IoT cũng đang là một xu thế hấp dẫn cho công nghệ toàn cầu.



Hình 1.2 Xếp hạng xu thế công nghệ năm 2019

Một nghiên cứu khác đến từ Safeatlat về thị trường IoT toàn cầu thì thị trường chiếm lĩnh thị phần cao nhất trong việc ứng dụng internet of things là smart cities (28.6%), industrial (26.4%), connected health (22.0%), smart homes (15.4%), connected cars (7.7%). Ta có thể thấy các ngành công nghiệp mũi nhọn hiện nay đều đang tích cực đổi mới ứng dụng internet of things vào trong lĩnh vực của mình.

1.2. Tính cấp thiết

Việc áp dụng giải pháp IoT giúp cuộc sống thông minh, tiện lợi và kết nối tốt hơn. Trong các doanh nghiệp, hệ thống internet vạn vật này cung cấp cái nhìn chi tiết về mọi thứ từ thời gian, hiệu suất máy móc đến chuỗi cung ứng. Ngoài ra IoT giúp công ty tự động hóa các quy trình, giảm chi phí lao động, cải thiện dịch vụ. Do đó, IoT là công nghệ quan trọng của cuộc sống hàng ngày và sẽ ngày càng phát triển mạnh mẽ.

Đề tài của đồ án này sử dụng IoT ứng dụng trong phòng học sẽ đem đến trải nghiệm tiện ích cho người dùng. Đây sẽ là một giải pháp ngày càng phổ biến trong tương lai, biến mọi căn phòng trong ngôi nhà của chúng ta trở nên thông minh, độc đáo.

Hệ thống giám sát không khí, điều khiển thiết bị điện sử dụng công nghệ IoT phù hợp với người dùng nhằm đem lại sự tiện nghi và tiết kiệm thời gian để người sử dụng tập trung tối đa cho công việc. Với hệ thống này, ta sẽ không cần phải di chuyển nhiều để bật tắt thiết bị trong khi làm việc. Điều kiện cần thiết để sử dụng rất đơn giản, ta chỉ cần 1 chiếc điện thoại thông minh hoặc 1 chiếc máy tính bất kỳ cùng 1 đường truyền mạng internet.

1.3. Nhiệm vụ của đề tài

Đề tài “*Hệ thống giám sát không khí, điều khiển thiết bị điện sử dụng công nghệ Internet of Things áp dụng trong phòng học*” bao gồm 4 nội dung:

- Nội dung 1: Tìm hiểu như thế nào là một hệ thống Internet of Things, nắm rõ được nguyên lý hoạt động.
- Nội dung 2: Tìm hiểu về các cảm biến, vi xử lý, bo mạch được sử dụng trong đề tài.
- Nội dung 3: Tìm hiểu giao thức HTTP và các ngôn ngữ lập trình được sử dụng trong đề tài (HTML, CSS, JavaScript).
- Nội dung 4: Thiết kế hệ thống sử dụng công nghệ Internet of Things.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Chi tiết về Internet of Things

2.1.1. Định nghĩa

Internet of Things (IoTs) thực tế hiện tại chưa có định nghĩa nào gọi là tiêu chuẩn chung, do đó để có cái nhìn đơn giản và tổng quan nhất em đã tham khảo các định nghĩa của các hãng công nghệ hàng đầu thế giới về IoT như Google, Cisco, AWS, ... Mỗi tổ chức kể trên đều đưa ra một định nghĩa riêng nhưng nhìn chung thì chúng ta có thể định nghĩa đơn giản như sau:

- Internet of Things là một thuật ngữ trong giới công nghệ thông tin với ý nghĩa về việc mở rộng sức mạnh của internet vượt ra ngoài khỏi phạm vi máy tính và điện thoại thông minh, kết nối với vạn vật. IoT đề cập đến hàng tỷ thiết bị vật lý trên khắp thế giới hiện được kết nối internet, thu thập và chia sẻ dữ liệu. Nhờ bộ xử lý bên trong cùng mạng không dây nó có khả năng truyền dữ liệu qua mạng mà không cần sự tương tác giữa người và máy tính, ta có thể biến mọi thứ trở nên chủ động và thông minh hơn.

- Internet of Things mô tả một mạng lưới các thực thể vật lý, thường tích hợp các cảm biến, phần mềm và các công nghệ khác nhau với mục đích kết nối và trao đổi dữ liệu với nhau qua môi trường internet.

2.1.2. Cách thức hoạt động

Hệ thống IoT bao gồm các thiết bị thông minh hỗ trợ web sử dụng các hệ thống nhúng, chẳng hạn như bộ xử lý, cảm biến và phần cứng. Các thiết bị IoT chia sẻ các dữ liệu cảm biến mà chúng thu thập được bằng cách kết nối với cổng IoT hoặc thiết bị biên khác, nơi dữ liệu được gửi đến đám mây để phân tích hoặc phân tích cục bộ. Đôi khi các thiết bị này giao tiếp với các thiết bị liên quan khác và hoạt động dựa trên thông tin chúng nhận được từ nhau. Các thiết bị thực hiện hầu hết công việc mà không có sự can thiệp của con người, mặc dù mọi người có thể tương tác với các thiết bị.

2.1.3. Lợi ích

Sử dụng hiệu quả dữ liệu: Dữ liệu là một phần không thể thiếu của bất kỳ lĩnh vực nào, bằng chứng là sự phát triển rất nhanh của các công nghệ về xử lý và phân tích dữ liệu. Một trong số đó là công nghệ dữ liệu lớn Big Data. Hiện nay các cảm biến được sử dụng trong các ứng dụng IoT rất đa dạng và tốc độ truyền nhận dữ liệu trong môi trường internet cũng ngày càng nhanh. Nên IoT có thể giúp chúng ta thu thập một lượng lớn dữ liệu một cách tự động. Dựa vào đó doanh nghiệp có thể giám sát các thông số chi tiết của hệ thống từ xa đưa ra dự đoán hoặc kịp thời đưa ra các điều chỉnh cần thiết để cải thiện hiệu quả vận hành.

Giảm thiểu sự tác động của con người: Việc tích hợp internet of things vào hệ thống sẽ cung cấp khả năng tự động hóa trong rất nhiều lĩnh vực từ smart homes cho tới các nhà máy trong công nghiệp có thể giúp tối thiểu hóa thời gian tác động của con người và giúp cắt giảm chi phí vận hành.

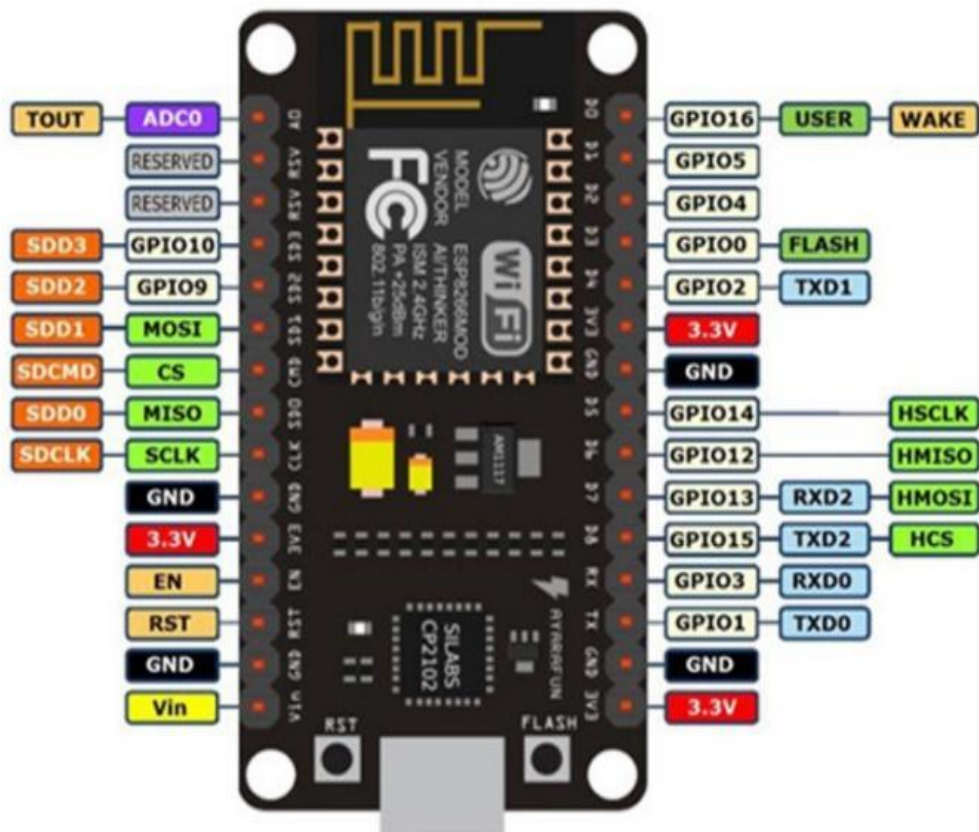
Cải thiện độ an toàn: Các ứng dụng Internet of Things giúp ta có thể tương tác và giám sát các chi tiết từ xa bằng nhiều phương tiện ví dụ như ứng dụng qua website, mobile hay sử dụng giọng nói trong các trợ lý ảo rất phổ biến trong các ứng dụng smart home. Đặc biệt trong các nhà máy khi các hệ thống vận hành liên tục thì việc giám sát hệ thống từ xa là rất cần thiết để khi có sự cố hoặc bất thường thì sẽ xử lý kịp thời. Vì vậy có thể nói IoT giúp tăng tính an toàn rất nhiều lĩnh vực.

Nâng cao chất lượng cuộc sống: đây là lợi ích hàng đầu và cũng là lợi ích hiệu quả mà các lợi ích ở phía trên mang lại. Hiện nay các thiết bị IoT xuất hiện ở khắp nơi

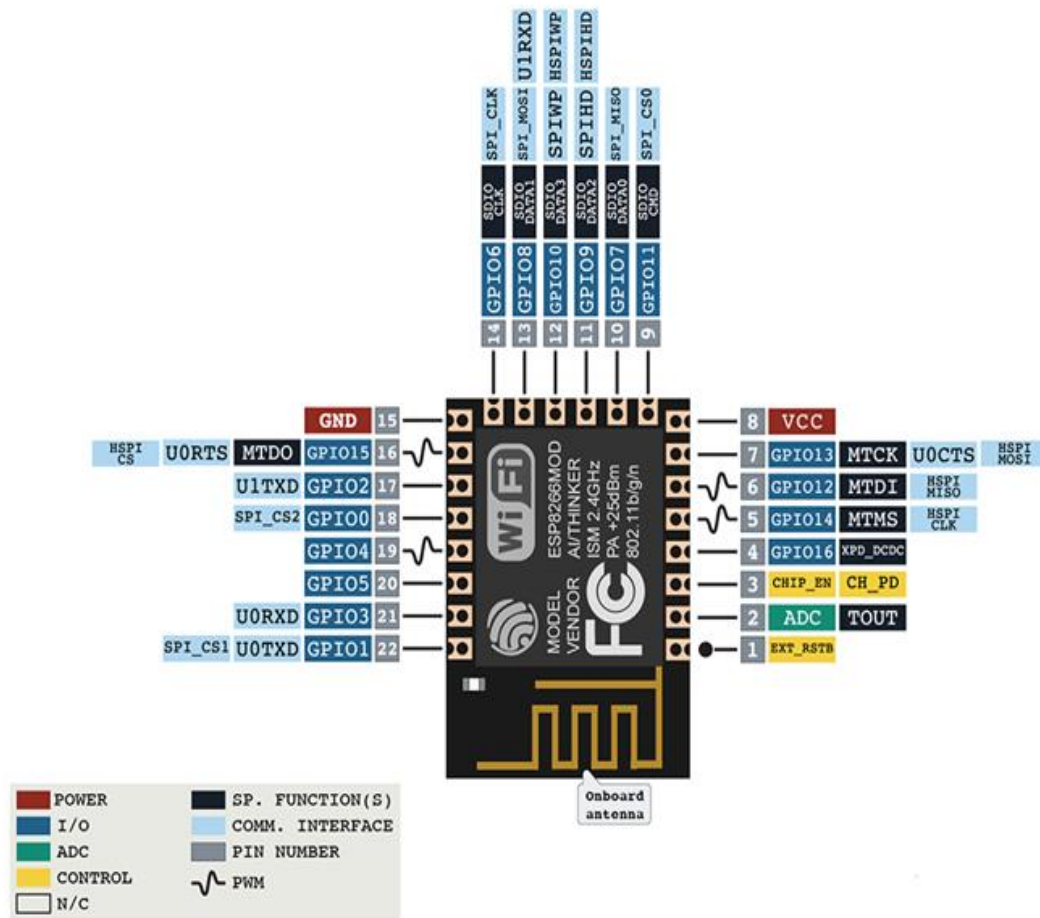
2.2. Cơ sở lý thuyết của hệ thống

2.2.1. Tổng quan về NodeMCU

2.2.1.1. Giới thiệu Kit NodeMCU và Module ESP8266



Hình 2.1 Kit NodeMCU



Hình 2.2 Module ESP8266

NodeMCU là firmware dựa trên mã nguồn mở LUA được phát triển cho chip wifi ESP8266. Firmware NodeMCU đi kèm với bo ESP8266, tức Kit NodeMCU. Vì NodeMCU là một nền tảng mã nguồn mở, thiết kế phần cứng của nó có thể mở để chỉnh sửa hoặc sửa đổi hoặc xây dựng thêm.

Module ESP8266-01 hỗ trợ chuẩn giao tiếp UART, thích hợp giao tiếp với vi điều khiển để truyền tải dữ liệu hay điều khiển các thiết bị thông qua WiFi. Nó có khả năng hoạt động độc lập với 2 chân I/O (cho phép module kết nối trực tiếp với cảm biến, thiết bị ngoại vi hoặc điều khiển máy chủ thông qua Wifi) và có khả năng lưu trữ với bộ nhớ Flash 1MB. 2 chân I/O cho mức điện áp ra tối đa 3.6V nên cần một bộ chuyển đổi mức điện áp đối với các thiết bị điện áp cao hơn như Arduino (5V).

Module ESP8266 sử dụng nguồn áp 3.3V nên cần dùng 1 mạch chuyển đổi điện áp 3.3V để module hoạt động. Là một sự lựa chọn khá tốt khi làm dự án IoT với giá thành tương đối thấp.

2.2.1.2. Thông số kỹ thuật và tính năng

Thông số:

Microcontroller	32-bit Soc
Operating Voltage	3.3V
Input Voltage	7-12V
Digital I/O Pins (DIO)	16
Analog Input Pins (ADC)	1
UART	1
SPI	1
I2C	1
Flash Memory	4 MB
SRAM	64 KB
Clock Speed	84 MHz

Tính năng:

- USB-TTL dựa trên CP2102 được tích hợp sẵn trên bo mạch, cho phép xây dựng thiết bị USB một cách dễ dàng.
- PCB Antenna.
- Mô-đun có kích thước nhỏ, phù hợp cho các dự án IoT.

2.2.1.3. GPIO (General purpose input, output)

PinMode(PIN_NUMBER, MODE):

- Chức năng: Để khai báo sử dụng một chân là Input hoặc Output.
- *PIN_NUMBER*: Số thứ tự chân GPIO.
- *MODE*: INPUT, OUTPUT, INPUT_PULLUP.

DigitalRead(PIN_NUMBER):

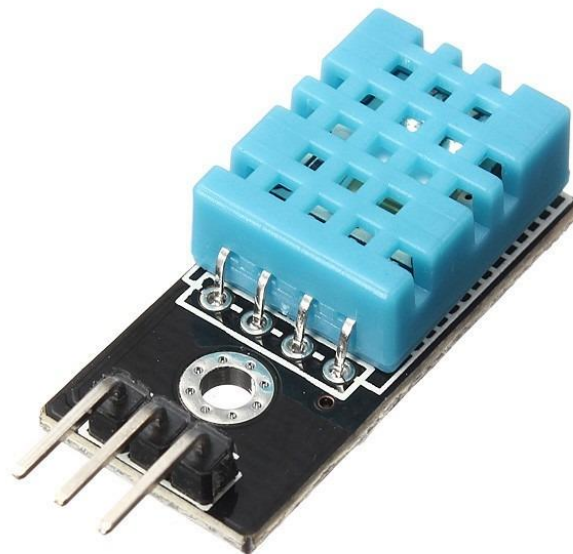
- Chức năng: Đọc giá trị digital tại chân *PIN_NUMBER*.
- *PIN_NUMBER*: Số thứ tự chân GPIO.
- Giá trị trả về: 0 (Tương ứng LOW) hoặc 1 (Tương ứng HIGH).

DigitalWrite(PIN_NUMBER, VALUE):

- Chức năng: Ghi mức điện áp *VALUE* ra chân *PIN_NUMBER*.
- *PIN_NUMBER*: Số thứ tự chân GPIO.
- *VALUE*: LOW hoặc HIGH.

2.2.2. Cảm biến nhiệt độ độ ẩm DHT11

2.2.2.1. Giới thiệu



Hình 2.3 Cảm biến nhiệt độ, độ ẩm DHT11

DHT11 là cảm biến đo nhiệt độ độ ẩm phổ biến do giá thành rẻ và tính ứng dụng cao, tích hợp sẵn điện trở 5.1k giúp người dùng dễ dàng kết nối và sử dụng. Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp có được dữ liệu chính xác mà không cần phải qua bất kỳ tính toán nào. DHT11 sử dụng chuẩn giao tiếp one-wire (Sử dụng 1 dây data để truyền dữ liệu) vì vậy khi giao tiếp với vi điều khiển chú ý cài đặt thời gian timer để dữ liệu đọc không bị lỗi.

2.2.2.2. Thông số kỹ thuật

- Điện áp hoạt động: 3,5V đến 5,5V
- Dòng điện hoạt động: 0,3mA (đo) 60uA (chờ)
- Đầu ra: Dữ liệu nối tiếp
- Phạm vi nhiệt độ: 0°C đến 50°C
- Phạm vi độ ẩm: 20% đến 90%
- Độ phân giải: Nhiệt độ và độ ẩm đều là 16 bit
- Độ chính xác: $\pm 1^\circ\text{C}$ và $\pm 1\%$

2.2.2.3. Nguyên lý hoạt động

Để đo độ ẩm, họ sử dụng thành phần cảm biến độ ẩm có hai điện cực với chất giữ ẩm giữa chúng. Vì vậy, khi độ ẩm thay đổi, độ dẫn của chất nền thay đổi hoặc điện trở giữa các điện cực này thay đổi. Sự thay đổi điện trở này được đo và xử lý bởi IC khiến cho vi điều khiển luôn sẵn sàng để đọc.

Mặt khác, để đo nhiệt độ, các cảm biến này sử dụng cảm biến nhiệt độ NTC hoặc điện trở. Một nhiệt điện trở thực sự là một điện trở thay đổi điện trở của nó với sự thay đổi của nhiệt độ. Các cảm biến này được chế tạo bằng cách thiêu kết các vật liệu bán dẫn như gốm hoặc polyme để cung cấp những thay đổi lớn hơn trong điện trở chỉ với những thay đổi nhỏ về nhiệt độ.

2.2.2.4. Ứng dụng

Ứng dụng trong việc xác định nhiệt độ, độ ẩm không khí. Sử dụng trong lĩnh vực khí tượng, nông nghiệp thông minh.

2.2.3. Cảm biến khí gas MQ2

2.1.3.1. Giới thiệu



Hình 2.4 Cảm biến khí gas MQ2

MQ2 là cảm biến khí gas. Nó được cấu tạo từ chất bán dẫn SnO_2 . Chất này có độ nhạy cảm thấp với không khí sạch. Nhưng khi trong môi trường có chất gây cháy, độ dẫn của nó thay đổi ngay. Chính nhờ đặc điểm này người ta thêm vào mạch đơn giản để biến đổi từ độ nhạy này sang điện áp.

Khi môi trường sạch điện áp đầu ra của cảm biến thấp, giá trị điện áp đầu ra càng tăng khi nồng độ khí gây cháy xung quanh MQ2 càng cao.

MQ2 hoạt động rất tốt trong môi trường khí hóa lỏng LPG, H_2 , và các chất khí gây cháy khác. Nó được sử dụng rộng rãi trong công nghiệp và dân dụng do mạch đơn giản và chi phí thấp.

2.1.3.2. Thông số kỹ thuật

- Nguồn hoạt động: 5V
- Tín hiệu đầu ra: Analog (AO) và Digital (DO)
- Phạm vi phát hiện rộng
- Tốc độ phản hồi nhanh và độ nhạy cao
- Mạch đơn giản
- Ổn định khi sử dụng trong thời gian dài

2.1.3.3. Nguyên lý hoạt động

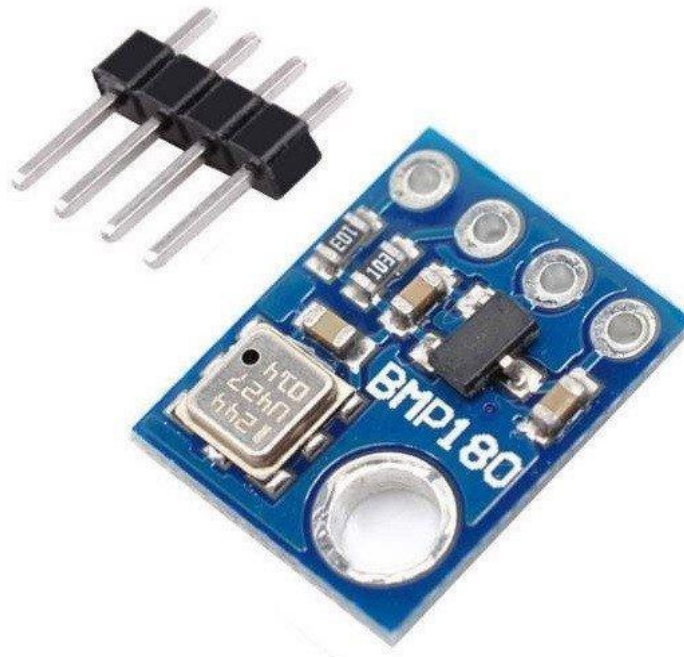
Đo hàm lượng chất gây cháy trong không khí và đưa ra điện áp ở chân AO tương ứng.

2.1.3.4. Ứng dụng

Ứng dụng để phát hiện khí gas, các khí gây cháy. Áp dụng cho việc phòng chống cháy nổ.

2.1.4. Cảm biến áp suất không khí BMP180

2.1.4.1. Giới thiệu



Hình 2.5 Cảm biến áp suất BMP180

Cảm biến áp suất BMP180 đo áp suất của môi trường sử dụng áp kế số, bằng cách chuyển đổi áp suất thành độ cao tương ứng, có thể dùng cho việc xác định độ cao của robot, máy bay hay một vật thể được phóng lên cao, cảm biến có thể đo được áp suất trong dải 300 ~ 1100hPa với dòng điện nhỏ khoảng 0.3uA thích hợp cho các thiết bị sử dụng Pin.

Cảm biến BMP180 đi kèm bộ hiệu chỉnh và sẵn sàng cho việc sử dụng, cảm biến có giao tiếp chuẩn I2C và có trở kéo sẵn trên board.

2.1.4.2. Thông số kỹ thuật

- Điện áp cung cấp: 1.8 ~ 3.6V
- Công suất tiêu thụ thấp: 0.5uA tại 1Hz
- Giao tiếp chuẩn I2C
- Tốc độ I2C max: 3.5MHz
- Độ nhiễu rất thấp: Lên đến 0.02hPa (17cm)
- Có sẵn bộ hiệu chỉnh bên trong
- Dải đo áp suất: 300 hPa ~ 1100 hPa (+9000m đến -500m)
- Trọng lượng: 1.18g
- Kích thước: 21mm x 18mm

2.1.4.3. Nguyên lý hoạt động

Cảm biến áp suất BMP180 được thiết kế để kết nối trực tiếp với vi điều khiển theo chuẩn I2C, dữ liệu áp suất và nhiệt độ sẽ được bù vào bởi hàm hiệu chuẩn dữ liệu của E2PROM của BMP180.

BMP180 bao gồm cảm biến điện trở, qua bộ chuyển đổi tín hiệu tương tự thành tín hiệu số và bộ điều khiển E2PROM, giao tiếp I2C, BMP180 chuyển dữ liệu áp suất và nhiệt độ chưa được hiệu chỉnh, E2PROM chứa 176bit hiệu chỉnh độc lập để hiệu chỉnh lại dữ liệu.

UP = dữ liệu áp suất (16 to 19 bit).

UT = dữ liệu nhiệt độ (16 bit).

2.1.4.4. Ứng dụng

Ứng dụng trong việc xác định độ cao máy bay, độ sâu, và cũng áp dụng trong việc đo nhiệt độ.

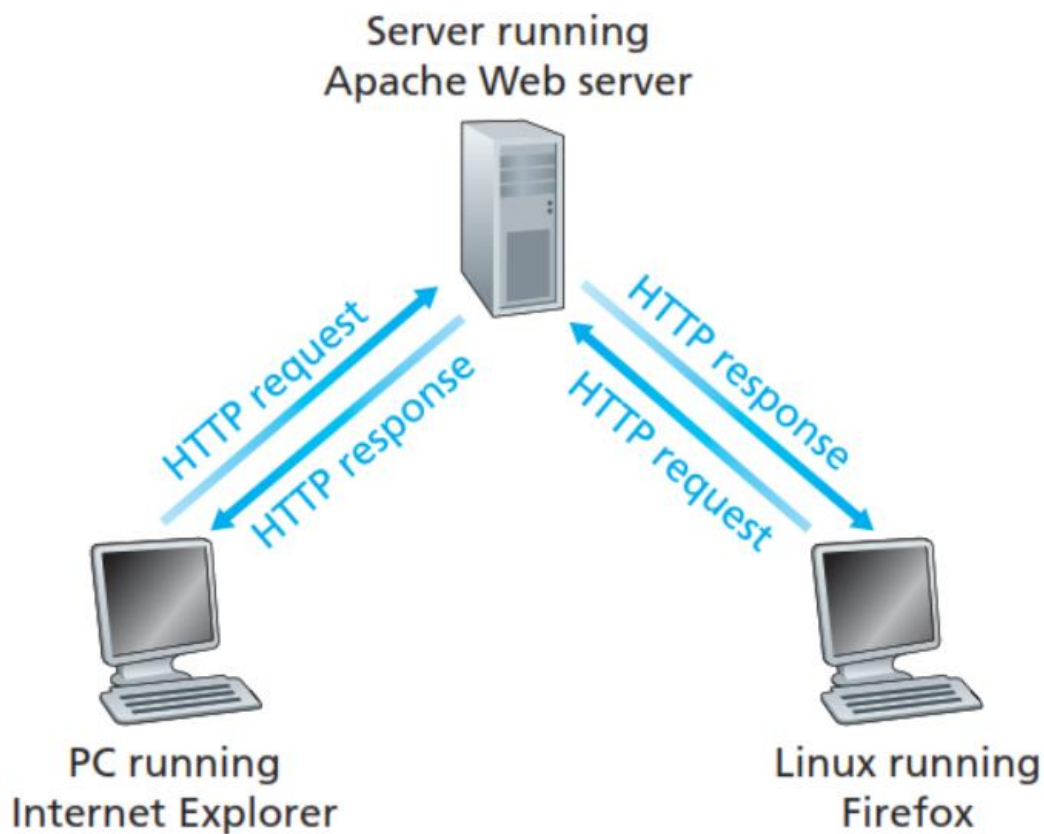
2.1.5. Giao thức HTTP

2.1.5.1. Giới thiệu

HTTP (Hyper Text Transfer Protocol) là một giao thức nằm ở tầng ứng dụng (Application layer) của tập giao thức TCP/IP, sử dụng để truyền nhận dữ liệu giữa các hệ thống phân tán thông qua internet, cụ thể giao thức hoạt động theo mô hình Client-Server bằng cách thực hiện các quá trình request-response giữa các hệ thống máy tính khác nhau.

Giao thức HTTP quy định cấu trúc của các gói tin và cách thức truyền nhận dữ liệu giữa client và server thông qua môi trường internet. Với khả năng truyền dẫn siêu văn bản (text, hình ảnh, âm thanh, video, ...), HTTP hiện là nền tảng truyền dẫn dữ liệu của ứng dụng duyệt web hiện nay và được ứng dụng rất nhiều trong các hệ thống Internet of Things.

2.1.5.2. Mô hình giao thức



Hình 2.6 Cấu trúc giao tiếp HTTP

Khi người dùng sử dụng trình duyệt và truy cập vào một website, một phiên làm việc HTTP (gọi là Session) sẽ được diễn ra với client là máy tính của người dùng và server là máy chủ của website. Mặc định HTTP sẽ được thực hiện thông qua port 80, đây là port chuẩn của giao thức được định nghĩa bởi tổ chức IANA quy định.

CHƯƠNG 3: THIẾT KẾ VÀ THỰC HIỆN PHẦN CỨNG

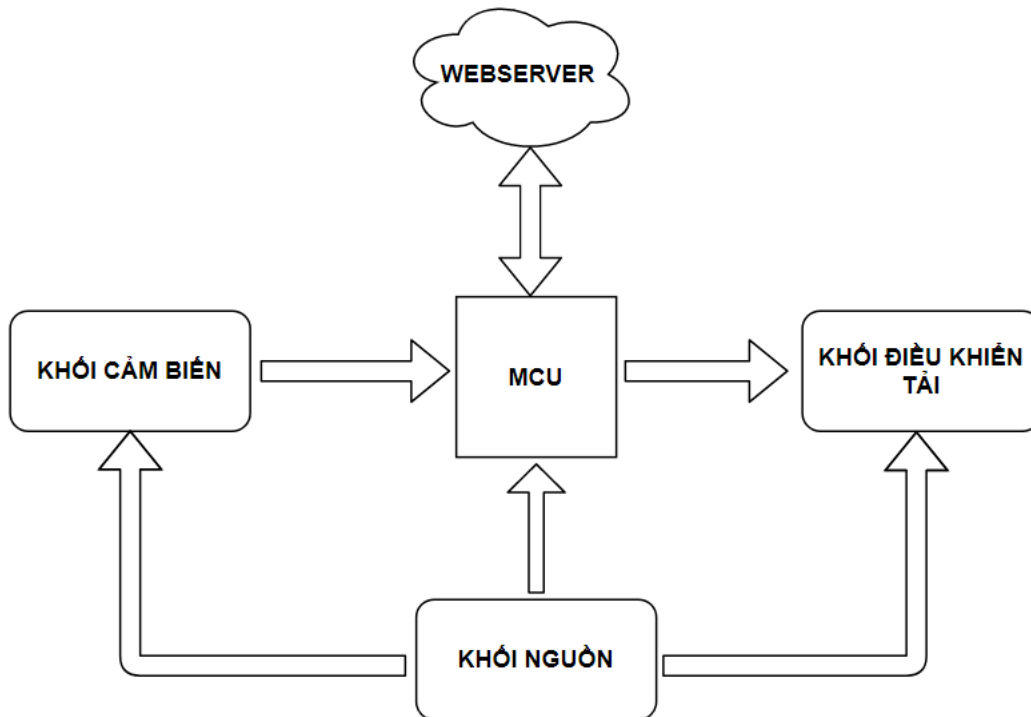
3.1. Yêu cầu thiết kế

- Mạch nhỏ gọn, đảm bảo và đáp ứng được hiệu quả của hệ thống.
- Thể hiện đúng bản chất, vai trò của hệ thống.
- Có thể đo được nhiệt độ, độ ẩm, khí gas, áp suất.
- Tủ bo mạch, điều khiển được thiết bị 220V.
- Thiết kế được khối nguồn để cấp đúng dòng điện cho các linh kiện cũng như vi điều khiển.
- Đảm bảo vi điều khiển, các linh kiện hoạt động an toàn, không bị sự cố gây cháy, hỏng hóc.

3.2. Phân tích thiết kế

- Để mạch nhỏ gọn, em thiết kế mạch từ phần mềm Proteus, gia công và in mạch lên board đồng, sau đó hàn gắn linh kiện theo đúng sơ đồ.
- Bản chất của đồ án là sử dụng công nghệ IoT, nên vi điều khiển cần có là esp8266.
- Để đo nhiệt độ, độ ẩm, khí gas, áp suất. Mạch cần có các cảm biến tương ứng để đọc được các giá trị liên quan.
- Để điều khiển các thiết bị sử dụng nguồn 220V. Mạch cần được lắp thêm Rowle phù hợp để tương tác giữa mạch và thiết bị điện.
- Mạch sử dụng nguồn 12V, bố trí 1 Adapter 12V để lấy nguồn trực tiếp từ ổ điện. Đồng thời thiết kế một khối nguồn hạ điện áp từ adapter về 5V để phù hợp điện áp mức của vi điều khiển.

3.3. Sơ đồ khối tổng quát và giải thích

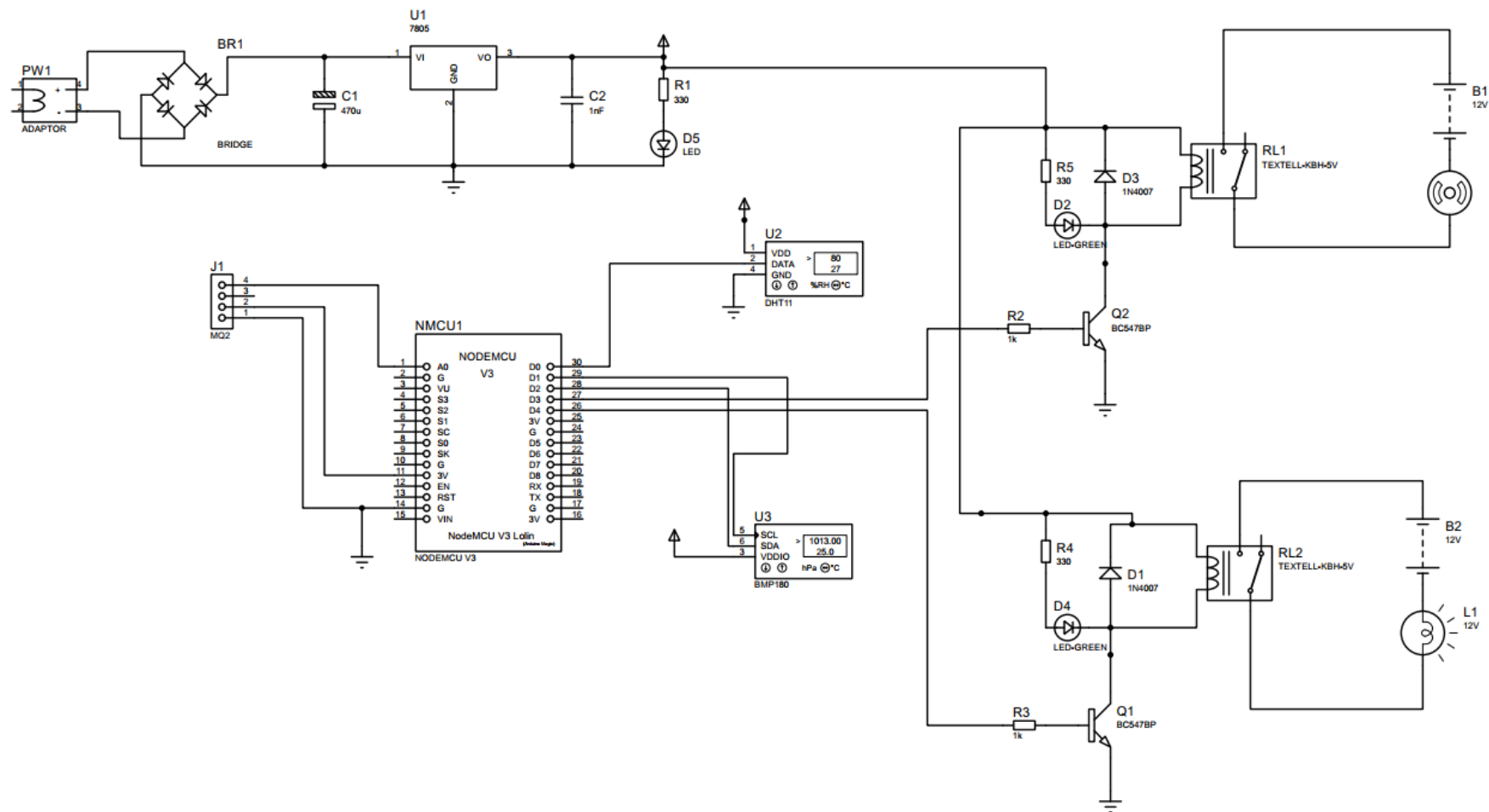


Hình 3.1 Sơ đồ khối tổng quát

Giải thích: Khối nguồn cấp nguồn cho khối cảm biến, khối điều khiển và NodeMCU hoạt động. Nhờ khối cảm biến, các giá trị nhiệt độ, độ ẩm, áp suất, khí gas được thu thập và gửi dữ liệu qua NodeMCU xử lý. NodeMCU tiếp tục gửi dữ liệu từ khối cảm biến lên Webserver để truyền thông số đo được cho người dùng. Cũng thông qua Webserver, người dùng tương tác lại với NodeMCU để truyền dữ liệu đến khối điều khiển. Từ đây, các thông tắc bật tắt thiết bị điện được thực hiện.

3.4. Tính toán và vẽ sơ đồ mạch chi tiết

3.4.1. Sơ đồ mạch chi tiết



Hình 3.2 Sơ đồ mạch chi tiết

3.4.2. Tính toán chi tiết

Chọn IC ổn áp **LM7805** để có điện áp ra 5V

Công thức tính tụ lọc (trường hợp chỉnh lưu toàn kỳ): $V_{g\phi n} = \frac{I_{max} \times \Delta t}{C}$

Tần số điện áp nguồn là 50Hz, ta tính được: $\Delta t = \frac{T_0}{2} = \frac{1}{2f} = \frac{1}{100} = 0.01s$

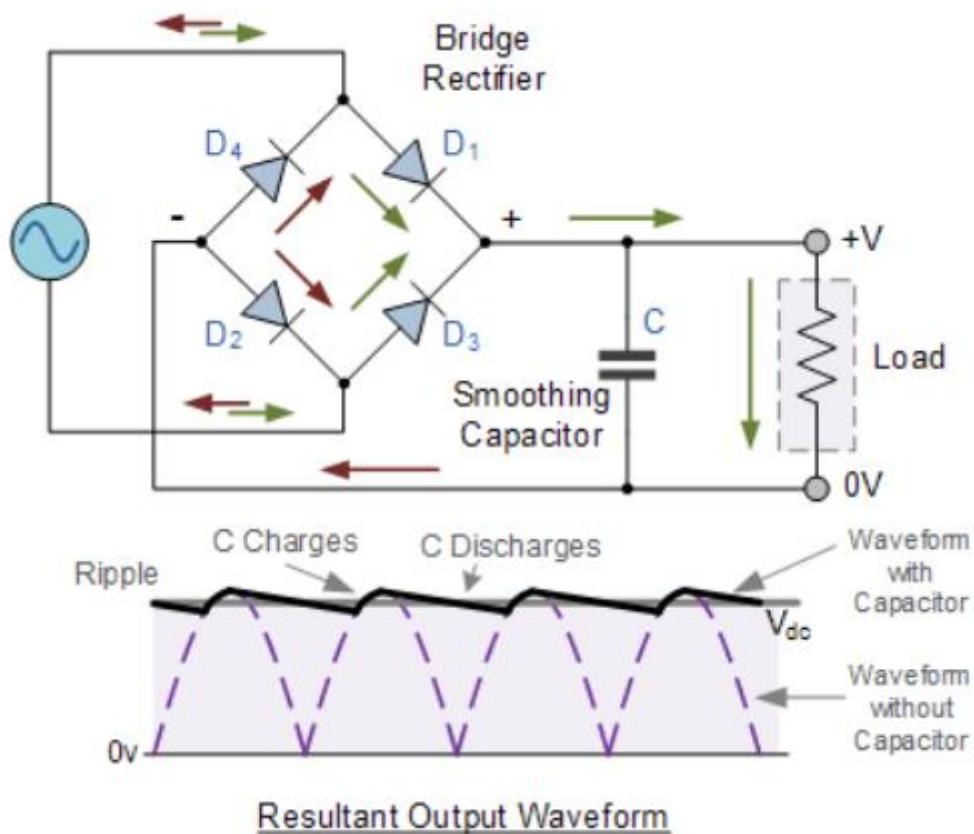
$I_{max} = 500mA$ (Datasheet)

$\Delta V = V_{g\phi n} = V_{sau\ bi\grave{e}n\ \acute{a}p} - V_{7805} = 12 - 7 = 5V$

\Rightarrow Chọn tụ C_1 sao cho $C_1 = \frac{500 \times 10^{-3} \times 0.01}{5} = 1000\mu F$

Vậy chọn tụ **1000 μF**

Giải thích: Sau khi chỉnh lưu ta thu được điện áp 1 chiều nhấp nhô, nếu không có tụ lọc thì điện áp này chưa thể dùng được vào mạch điện tử.



Hình 3.3 Chỉnh lưu toàn sóng với tụ điện cân bằng

Do tín hiệu kích từ vi điều khiển là 5V nên ta chọn Rơ le 5V để điều khiển tải. Theo datasheet của Relay, dòng hoạt động của cuộn dây khi rơ le được kích là:

$$I_{rl} = 71.4mA$$

Chọn $V_{cesat} = 0.2VDC$. Điện áp ngược đặt lên Diode khi BJT dẫn bão hòa là:

$$V_D = V_{cc} - V_{cesat} = 5 - 0.2 \approx 5VDC$$

Dòng xả qua Diode: $I_D = I_{rl} = 71.4mA$;

Ta chọn Diode có: $V_{AK} \geq 2 \sim 3V_D$; $I_d \geq 2 \sim 3I_{rl}$. Vậy ta chọn Diode D1: **1N4007**.

Ta chọn dòng $I_l = 10mA$ từ nguồn qua R4 và qua LED

Khi BJT dẫn bão hòa: $I_c = 71.4mA$; $V_{cesat} = 0.2VDC$. Công suất tiêu tán trên Q1:

$$P_c = I_c \times V_{cesat} = 71.4 \times 10^{-3} \times 0.2 = 0.014W.$$

Khi BJT tắt: $V_{cemax} = V_{cc} = 5VDC$. Ta chọn BJT sao cho:

$$V_{CE0} \geq 2V_{cc} = 10V.$$

$$I_{cmax} \geq 2I_c = 140mA.$$

$$P_{cmax} \geq 2P_c = 0.028W.$$

Vậy chọn BJT là: **NPN BC547**

P_{cmax}	I_{cmax}	V_{ce0}	f	h_{FE}	T
1.5W	100mA	45V	150 - 300Hz	110 - 800	55 - 150°C

$$\text{Ta có: } I_{Bthres} = \frac{I_{cmax}}{\beta_{min}} = \frac{71.4 \times 10^{-3}}{110} = 6.49 \times 10^{-4} A.$$

$$\text{Chọn } I_B = 2I_{Bthres} = 1.3 \text{ mA. } \Rightarrow R_1 = \frac{V_i - V_{BEsat}}{I_B} = \frac{3 - 0.7}{1.3 \times 10^{-3}} = 1538 \Omega$$

CHƯƠNG 4: THIẾT KẾ VÀ THỰC HIỆN PHẦN MỀM

4.1. Yêu cầu thiết kế

- Xây dựng được web server để lưu trữ dữ liệu, tương tác với người dùng thông qua ứng dụng cuối. Đồng thời phải thiết kế giao diện web server và ứng dụng cuối sao cho người dùng dễ dàng sử dụng.
- Web server, Blynk hiển thị giá trị đo không khí, đồng thời giúp người dùng tương tác với tải.

4.2. Phân tích thiết kế

- Sử dụng Blynk làm ứng dụng cuối để người dùng giao tiếp với hệ thống
- Sử dụng giao thức HTTP, đây hiện là nền tảng truyền dẫn dữ liệu của ứng dụng duyệt web hiện nay và được ứng dụng rất nhiều trong các hệ thống Internet of Things.
- Các ngôn ngữ lập trình HTML, CSS, JavaScript được sử dụng trong việc thiết kế giao diện Web server.

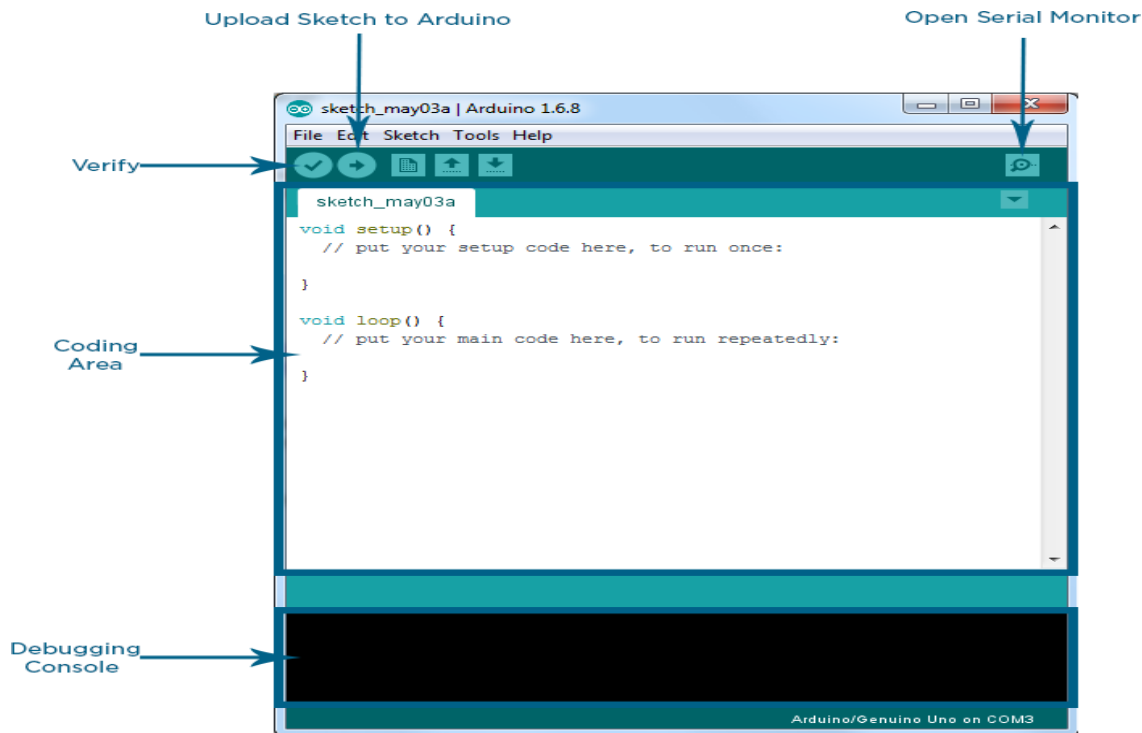
4.3. Phân tích phần mềm

4.3.1. Môi trường phát triển tích hợp Arduino (IDE)

Môi trường phát triển tích hợp Arduino (IDE) là một ứng dụng đa nền tảng (dành cho Windows, macOS, Linux) được viết bằng ngôn ngữ lập trình Java. Nó được sử dụng để viết và tải các chương trình lên các bảng tương thích Arduino.

Arduino IDE hỗ trợ các ngôn ngữ C và C++ bằng cách sử dụng các quy tắc cấu trúc mã đặc biệt. Arduino IDE cung cấp một thư viện phần mềm từ dự án Wires, cung cấp nhiều thủ tục đầu vào và đầu ra phổ biến.

Mã do người dùng viết chỉ yêu cầu hai hàm cơ bản, để bắt đầu phác thảo và vòng lặp chương trình chính, được biên dịch và liên kết với hàm main thành một chương trình điều hành theo chu kỳ thực thi với chuỗi công cụ GNU, cũng được bao gồm trong bản phân phối IDE. Arduino IDE sử dụng chương trình để chuyển đổi mã thực thi thành tệp văn bản ở dạng mã thập lục phân được nạp vào bảng Arduino bằng chương trình nạp trong phần sụn của bo mạch.



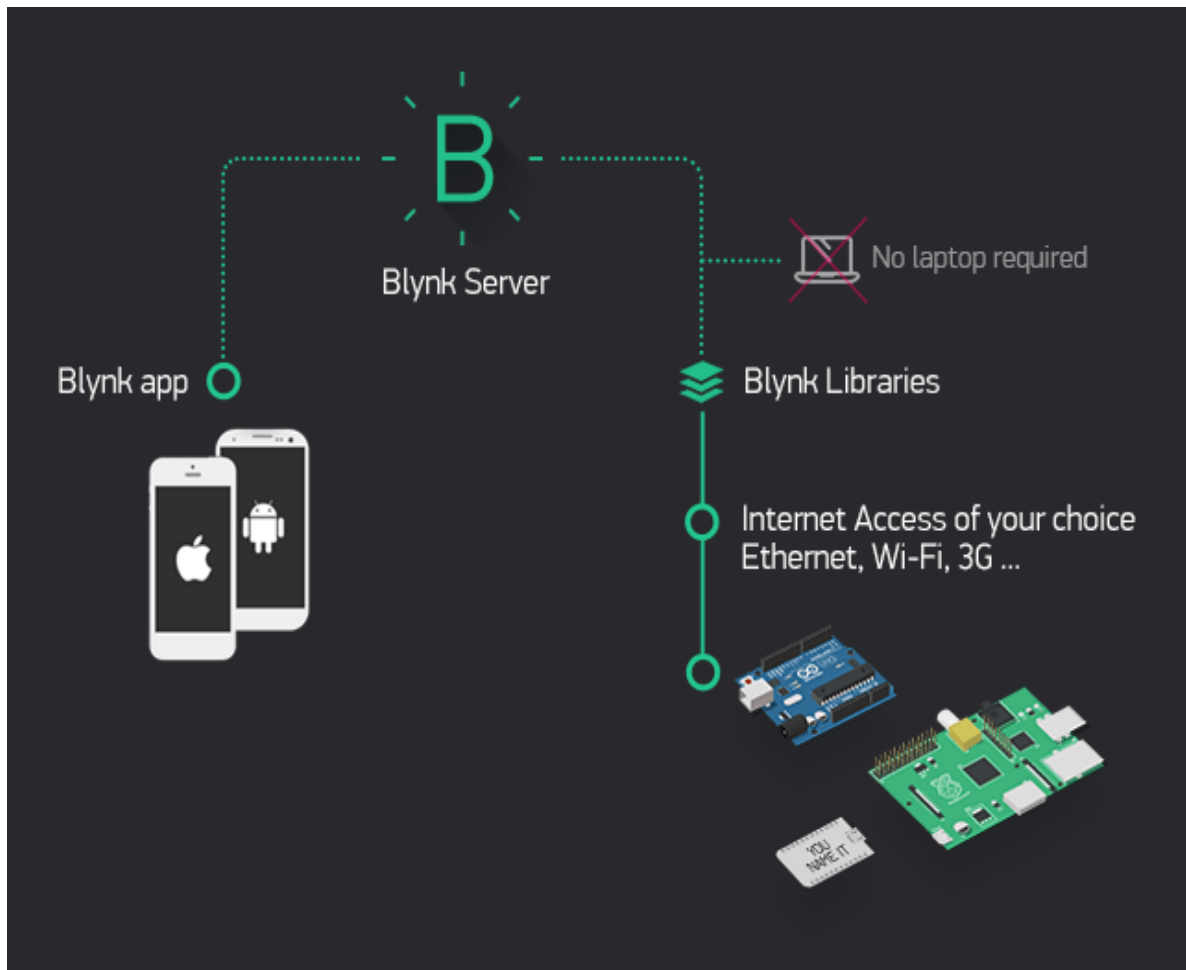
Hình 4.1 Giao diện ứng dụng Arduino IDE

4.3.2. Giới thiệu nền tảng Blynk

Blynk là một phần mềm mã nguồn mở được thiết kế cho các ứng dụng IoT. Ứng dụng giúp người dùng điều khiển phần cứng từ xa, có thể hiển thị dữ liệu cảm biến, lưu trữ dữ liệu, biến đổi dữ liệu hoặc làm nhiều việc khác. Hãy thử tưởng tượng khi ta ấn vào một nút nhấn trên phần mềm lập tức bóng đèn sáng lên và ngược lại.

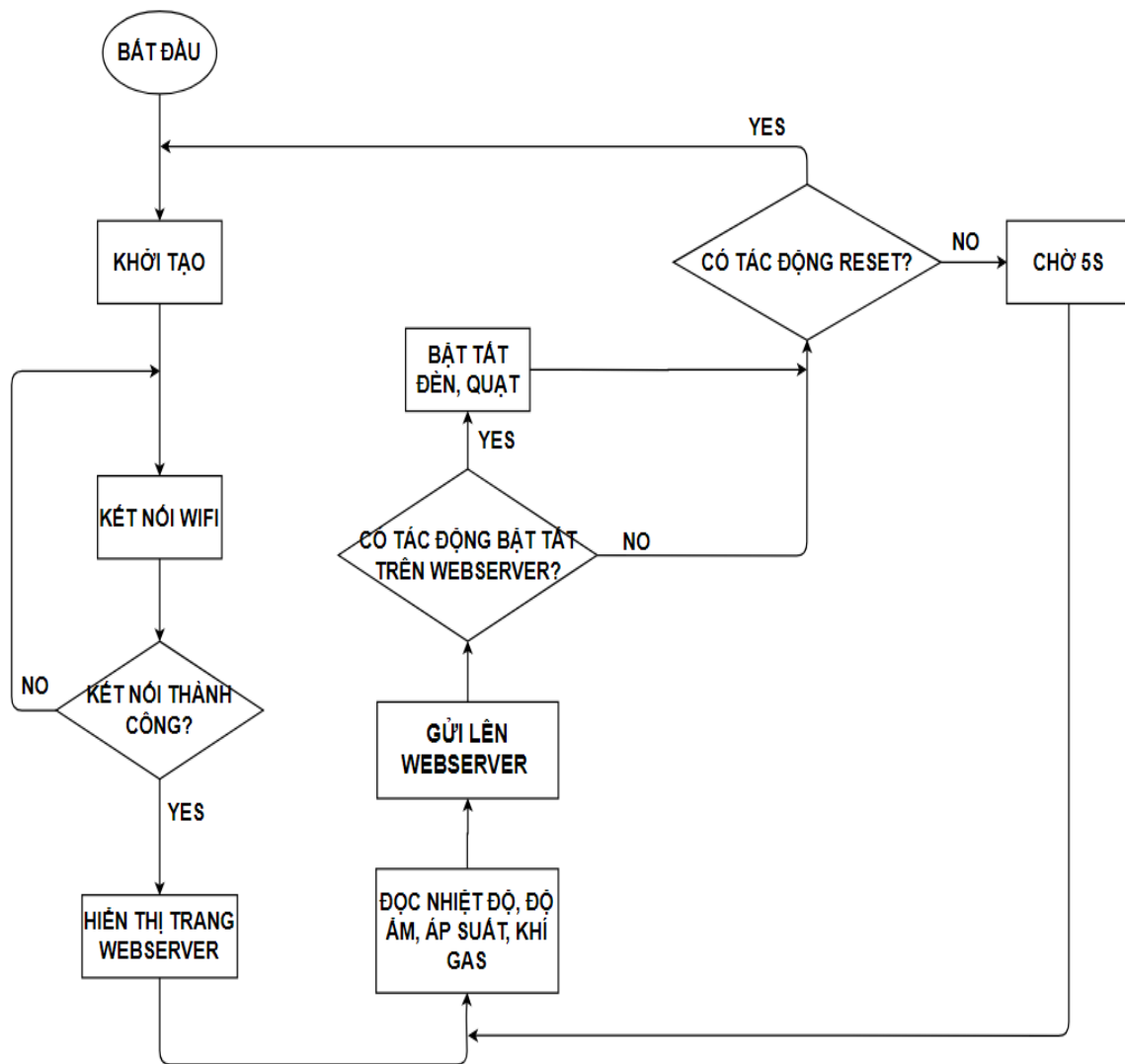
Nền tảng Blynk có ba phần chính:

- *Blynk App*: Ứng dụng Blynk cho phép khởi tạo giao diện cho các dự án của mình.
- *Blynk Server*: Chịu trách nhiệm giao tiếp qua lại hai chiều giữa điện thoại và phần cứng. Bạn có thể sử dụng server của Blynk nhưng sẽ bị giới hạn điểm Energy. Trong các hướng dẫn sau này mình sẽ sử dụng Server riêng của mình! Và bạn cũng có thể sử dụng nó.
- *Blynk Library*: Thư viện chứa các nền tảng phổ biến, giúp việc giao tiếp phần cứng với Server dễ dàng hơn.



Hình 4.2 Ba phần chính của Blynk

4.4. Lưu đồ thuật toán và giải thích



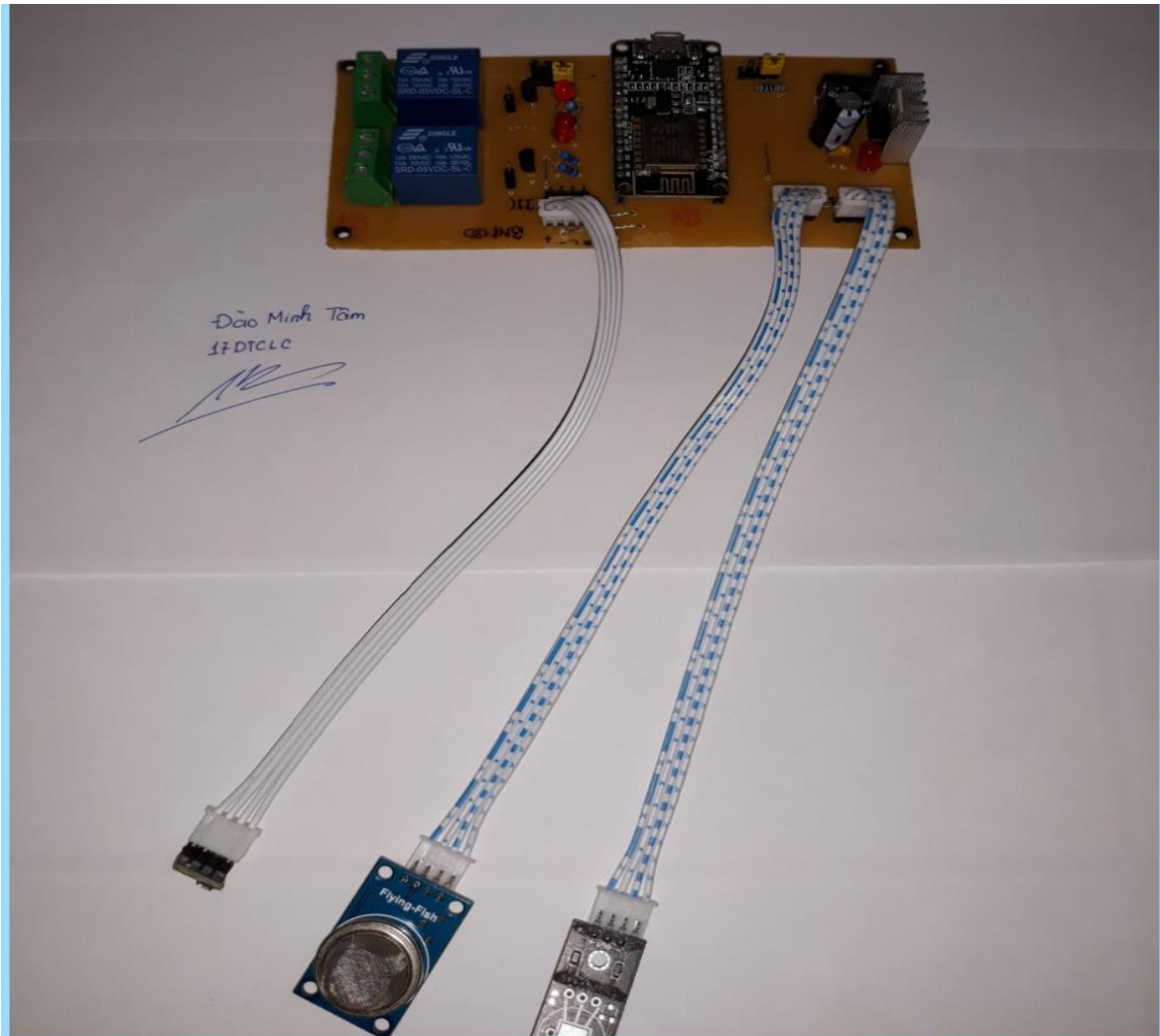
Hình 4.3 Lưu đồ thuật toán đề tài

Giải thích:

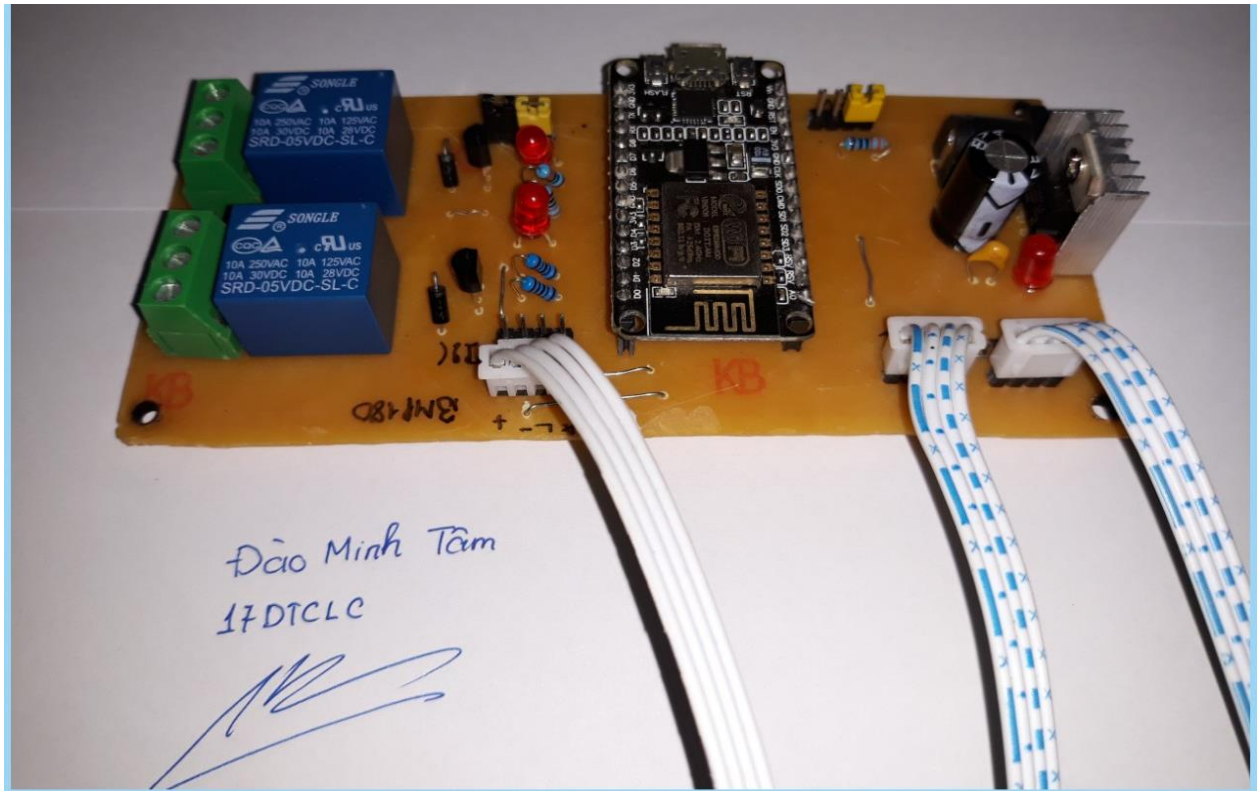
- Khi chương trình bắt đầu chạy, khởi tạo hàm.
- Thực hiện kết nối wifi, xác nhận kết nối. Nếu kết nối thành công, máy cấp cho MCU 1 địa chỉ IP để khởi tạo trang web. Nếu kết nối không thành công, lập tức kết nối lại wifi.
- Khi khởi tạo xong trang web, MCU đọc điện áp từ cảm biến và tính toán nhiệt độ, độ ẩm, khí gas, áp suất. Sau đó đẩy dữ liệu lên Webserver.
- Web server xác nhận hành động người dùng (bật/tắt). Nếu có tác động, thực hiện bật/tắt tải. Nếu không, thực hiện xác nhận tác động reset.
- Có tác động reset, chương trình quay lại khởi tạo hàm. Nếu không, tiếp tục đọc nhiệt độ, độ ẩm, áp suất, khí gas sau khi chờ 5 giây.

CHƯƠNG 5: KẾT QUẢ THỰC HIỆN

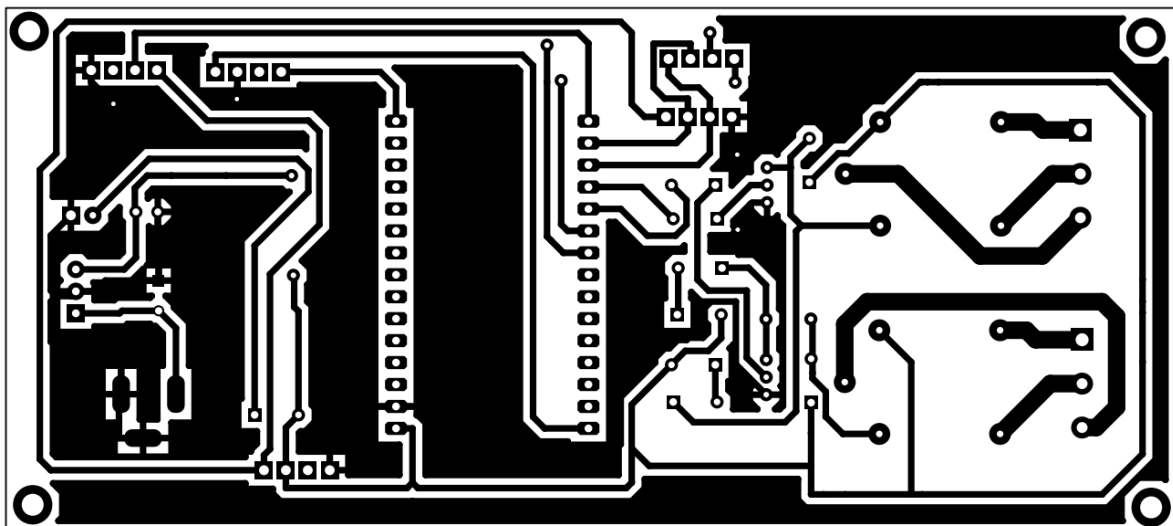
5.1. Kết quả thi công phần cứng



Hình 5.1 Kết quả thi công phần cứng tổng quát

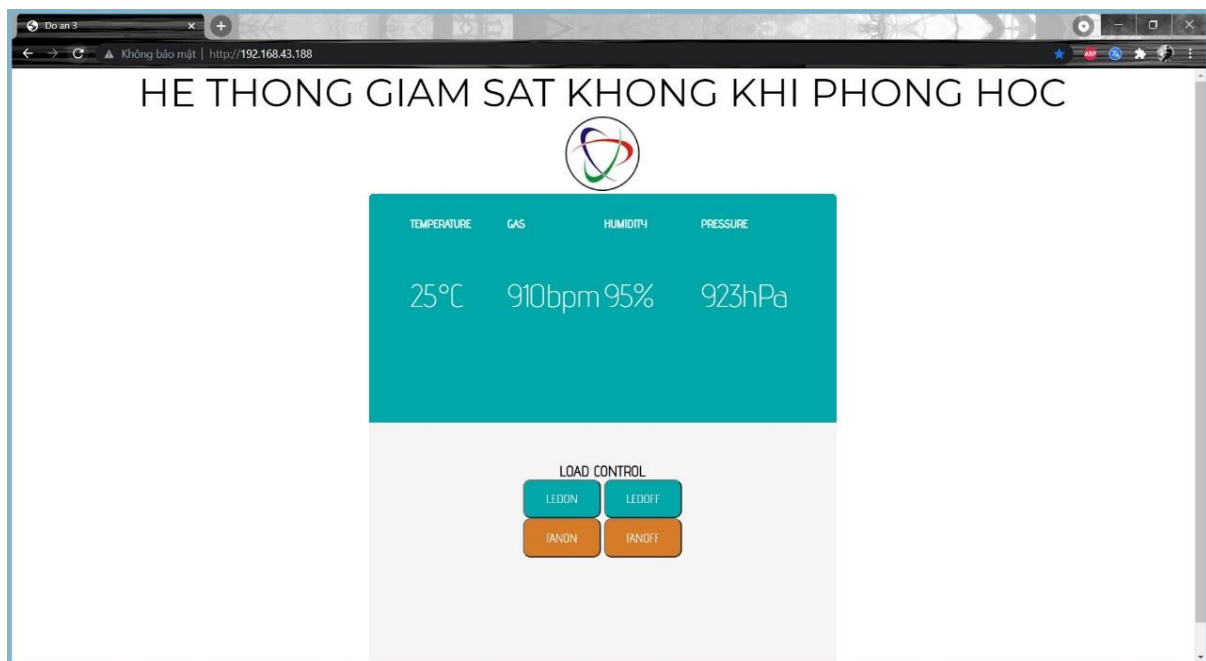


Hình 5.2 Kết quả thi công phần cứng chi tiết

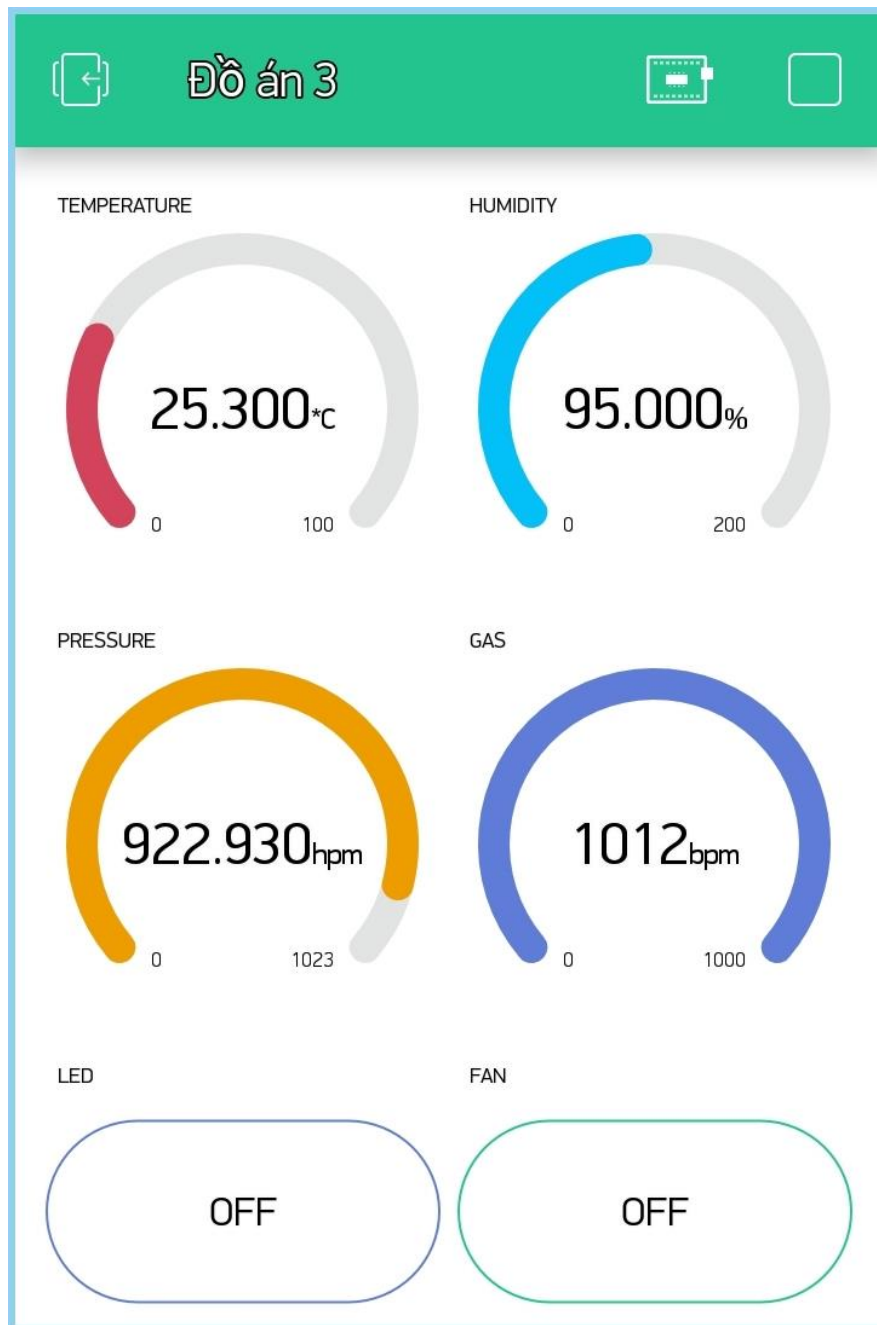


Hình 5.3 Board mạch được vẽ từ phần mềm Proteus

5.2. Kết quả thi công phần mềm



Hình 5.4 Kết quả giao diện web server



Hình 5.5 Kết quả giao diện nền tảng Blynk

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

1.1. Kinh nghiệm và những điều rút ra được từ kết quả đề tài

Từ kết quả của đề tài, em nhận thấy Internet of Things (IoTs) đúng là một xu hướng lớn của thế giới, tác động đến mọi mặt của xã hội. Các hệ thống Internet of Things rất đa dạng và muôn màu muôn vẻ. Tuy nhiên để tiếp cận với IoT được tốt hơn, bản thân em cần phải học hỏi và cố gắng thêm nhiều.

Các giao thức MQTT, HTTP, kể cả cơ sở dữ liệu MySQL, ... là những kiến thức rất cần để em khám phá nhiều hơn đến xu thế IoT.

Việc thực hiện đề tài của đồ án này, giúp em rút ra được rất nhiều kinh nghiệm:

- Có được cái nhìn tổng quan về IoTs trong thời đại Industry 4.0.
- Làm chủ được ESP8266 (KIT phát triển tích hợp WiFi), có thể tự tay tạo ra các ứng dụng IoTs cho riêng mình.
- Hiểu được cách hoạt động và làm việc được với giao thức HTTP trong các ứng dụng IoTs qua mạng WiFi.
- Làm việc được với ngôn ngữ lập trình website cơ bản.
- Có kinh nghiệm về thiết kế và gia công board mạch điện tử.

1.2. Ưu và khuyết điểm của kết quả nghiên cứu

- **Ưu điểm:** Mạch thiết kế nhỏ gọn, đáp ứng nhanh các thao tác. Web server sau khi hoàn thiện chạy ổn định, giao diện Blynk khá bắt mắt. Sau khi hoàn thiện xong em có thể lắp luôn vào phòng học của em để sử dụng, ba mẹ em cảm thấy rất thích thú với đề tài này của em.

- **Khuyết điểm:** Chưa thi công được mô hình phòng học, chính vì vậy em mong thầy và cô trong hội đồng bảo vệ thông cảm cho em về thiếu sót này. Đối với mạch, chưa tối ưu hóa được phần cứng, mới chỉ điều khiển được 2 thiết bị (đèn và quạt). Về web server chưa thiết kế được giao diện sao cho bắt mắt nhất.

2. Hướng phát triển

Hiện nay, MQTT và HTTP là hai giao thức phổ biến bậc nhất dùng cho các thiết bị Internet of Things (IoT), mỗi giao thức nên được thiết kế sử dụng phù hợp trong các ngữ cảnh khác nhau sẽ giúp hệ thống IoT trở nên linh hoạt và hiệu quả.

Đối với các ứng dụng cần lưu trữ dữ liệu lâu dài và hiển thị website, ngôn ngữ phổ biến mà em nhận thấy nên sử dụng là PHP kết hợp với cơ sở dữ liệu quan hệ MySQL. Có một số thư viện PHP hỗ trợ thực thi MQTT client bằng cách kết hợp thêm một số cơ chế giữ kết nối với broker tuy nhiên việc sử dụng cũng thường gặp nhiều khó khăn, vì vậy em thấy có một cách tốt hơn và cũng dễ thực hiện là có thể kết hợp với một ngôn ngữ khác như Node.js đóng vai trò là MQTT client kết nối giữa thiết bị với cơ sở dữ liệu MySQL và máy chủ website dùng PHP.

Đó là hướng phát triển mà em nghĩ cần áp dụng vào những đề tài IoT khác sau này của em.

TÀI LIỆU THAM KHẢO

- [1] Altera Corp, “SDRAM Controller for Altera’s DE2/ DE1 boards”, www.altera.com
- [2] Tapit Engineering CO.,LTD “Tổng hợp hướng dẫn IoT với NodeMCU ESP8266”, <https://tapit.vn/>
- [3] Tapit Engineering CO.,LTD “Khóa học cơ bản Internet of Things”, <https://tapit.vn/>
- [4] MESIDAS “Tổng quan về Web Server, nhúng Web cho thiết bị”, <https://mesidas.com/>
- [5] Codehub “Tự học thiết kế web với HTML”, <https://www.codehub.com.vn/>
- [6] Tapit Engineering CO.,LTD “Hướng dẫn nạp code cho ESP8266 sử dụng USB-TTL và Arduino IDE”, <https://tapit.vn/>
- [7] Arduino KIT “Hướng dẫn cài đặt ES8266 và kết nối với Blynk”, <https://arduinokit.vn/>

PHỤ LỤC

1. Code

1.1. Code giao diện Web Server

```
<!DOCTYPE html>
<html>
<head>
<title>Do an 3</title>
</head>
<style>
@import url(https://fonts.googleapis.com/css?family=Montserrat);
@import url(https://fonts.googleapis.com/css?family=Advent+Pro:400,200)
;
*{margin: 0;padding: 0;}

body{
    background:#544947;
    font-family:Montserrat,Arial,sans-serif;

}
h2{
    font-size:14px;
}
.widget{
    box-shadow:0 40px 10px 5px rgba(0,0,0,0.4);
    margin:100px auto;
    height: 330px;
    position: relative;
    width: 500px;
}

.upper{

background:#00A8A9;
border-radius:5px 5px 5px 5px;
font-family:'Advent Pro';
font-weight:200;
height:500px;
width:600px;
}

.date{
    font-size:40px;
}
.year{
    font-size:30px;
    color:#c1c1c1;
}
```

```
.place{
  color:#222;
  font-size:40px;
}
.lower{
  border-radius:0 0 0 0;
  background:#f5f5f5;
  height:200px;
  padding:50px;
  margin-top: 160px;

}

.infos{
  list-style:none;
}
.info{
  color:#fff;
  float:left;
  height:100%;
  padding-top:10px;
  text-align:right;
  width:25%;
}
.info span{
  display: inline-block;
  font-size:70px;
  margin-top:40px;
float:center;
}
.weather p {
  font-size:20px;padding:10px 0;
}
.anim{animation:fade .8s linear;}

@keyframes fade{
  0%{opacity:0;}
  100%{opacity:1;}
}

a{
  text-align: center;
  text-decoration: none;
  color: white;
  font-size: 15px;
  font-weight: 500;
}
```

```

        .tabcontent {
            display: none;
            border: 1px solid #2ec54f;
            border-top: none;
        }
        .b{width: 100px;height: 50px;font-size: 15px;color: #FFF;background-
        color:#00A8A9;border-radius: 10px;font-family:'Advent Pro';}
        .t{width: 100px;height: 50px;font-size: 15px;color: #FFF;background-
        color:#d67c28;border-radius: 10px;font-family:'Advent Pro';}
    </style>
    <body>
    <center>
        <div style="text-align: center;"><a style="align:center; font-
        size: 50px;">HỆ THỐNG GIÁM SÁT KHÔNG KHÍ PHÒNG HỌC</a></div>
        <div class="upper" >
            <ul class="infos">
                <li class="info temp">
                    <h2 class="title">TEMPERATURE</h2>
                    <span class='update' id="temp">21 &deg;C</span>
                </li>
                <li class="info gas">
                    <h2 class="title">GAS</h2>
                    <span class='update' id="gas">0%</span>
                </li>
                <li class="info humidity">
                    <h2 class="title">HUMIDITY</h2>
                    <span class='update' id="humidity">23%</span>
                </li>
            </ul>
            <br>
            <br>
            <div class="lower">

                <h3>LOAD CONTROL</h3>
                <p>
                    <tr>
                        <td><a href="/LEDON"><button class="b">LEDON</butto
                        n></a></td>
                        <td><a href="/LEDOFF"><button class="b">LEDOFF</but
                        ton></a></td>
                    </tr>
                    <br>
                    <tr>
                        <td><a href="/FANON"><button class="t">FANON</butto
                        n></a></td>

```

```

        <td><a href="/FANOFF"><button class="t">FANOFF</but
ton></a></td>
    </tr>

</p>

</div>

</div>
</div>
<script>

    //Get Humidity Temperature and Rain Data
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
        if (this.readyState == 4 && this.status == 200) {
            var txt = this.responseText;
            var obj = JSON.parse(txt); //Ref: https://www.w3schools.com/js/js
_json_parse.asp
            document.getElementById("Gas").innerHTML = obj.Gas + "%";
            document.getElementById("temp").innerHTML = Math.round(obj.Temp
erature) + "&deg;C";
            document.getElementById("humidity").innerHTML = Math.round(obj.
Humidity) + "%";
        }
    };
    xhttp.open("GET", "readADC", true); //Handle readADC server on ESP82
66
    xhttp.send();
}
</script>

</center>
</body>
</html>

```

1.2. Code điều khiển NodeMCU

```

#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP085_U.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>

```

```
#include <ESP8266WebServer.h>
#include "html.h"
#include <dht11.h>
#define BLYNK_PRINT Serial
#include <BlynkSimpleEsp8266.h>
#define Load1 2
#define Load2 0
#define dht_pin D0
BlynkTimer timer;
Adafruit_BMP085_Unified bmp = Adafruit_BMP085_Unified(10085);

dht11 dhtObject;

const char* ssid = "OK e";
const char* password = "tam123123";
char auth[] = "5fsYnDsdVj2Pqf4yrgBsozwqib1E0IDv";
ESP8266WebServer server(80);

void handle_OnConnect() {
    String s= MAIN_page;
    server.send(200, "text/html", s);
}

void handleADC() {
    int gas = analogRead(A0);
    gas= gas*1023/100;
    float temperature, humidity, pressure, atttitude;
    dhtObject.read(dht_pin);
    Serial.print(" Humidity in %= ");
    humidity = dhtObject.humidity;
    sensors_event_t event;
    bmp.getEvent(&event);

    if (event.pressure)
```

```

{
  Serial.print("Pressure:   ");
  Serial.print(event.pressure);
  Serial.println(" hPa");
  bmp.getTemperature(&temperature);
  Serial.print("temperature: ");

  Serial.print(temperature);
  Serial.println("°C");
  float seaLevelPressure = SENSORS_PRESSURE_SEALEVELHPA;
  Serial.print("Altitude:");
  Serial.print(bmp.pressureToAltitude(seaLevelPressure,
event.pressure));
  Serial.println(" m");
  Serial.println("");
}

String data = "{\"gas\": \""+String(gas)+"\", \"humidity\": \""+
String(humidity) +"\", \"pressure\": \""+ String(event.pressure) +"\",
\"temperature\": \""+ String(temperature) + "\"}";

server.send(200, "text/plain", data);
Blynk.virtualWrite(V5, temperature);
Blynk.virtualWrite(V6, humidity);
Blynk.virtualWrite(V7, event.pressure);
Blynk.virtualWrite(V8, gas);

}

void handleLEDOn() {
  Serial.println("LED on");
  digitalWrite(Load1, HIGH);
  server.sendHeader("Location", "/");
  server.send(303);
}

```

```

void handleLEDOff() {
    Serial.println("LED off");
    digitalWrite(Load1, LOW);
    server.sendHeader("Location", "/");
    server.send(303);
}

void handleRelayOn() {
    Serial.println("Fan on");
    digitalWrite(Load2, HIGH);
    server.sendHeader("Location", "/");
    server.send(303);
}

void handleRelayOff() {
    Serial.println("Fan off");
    digitalWrite(Load2, LOW);
    server.sendHeader("Location", "/");
    server.send(303);
}

void handle_NotFound() {
    server.send(404, "text/plain", "Not found");
}

void setup()
{
    Serial.begin(9600);
    delay(10);

    WiFi.begin(ssid, password);
    Serial.println("");

```



```
pinMode(Load1, OUTPUT);
pinMode(Load2, OUTPUT);

digitalWrite(Load1, LOW);
digitalWrite(Load2, LOW);
Blynk.begin(auth, ssid, password);

// Wait for connection
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

//If connection successful show IP address in serial monitor
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
if(!bmp.begin())
{
    Serial.print(" no BMP085 detected ... Check your wiring or I2C
ADDR!");
    while(1);
}

server.on("/", handle_OnConnect);
server.on("/readADC", handleADC);
server.onNotFound(handle_NotFound);
server.on("/LEDON", handleLEDOOn);
server.on("/LEDOFF", handleLEDOff);
server.on("/FANON", handleRelayOn);
server.on("/FANOFF", handleRelayOff);
server.begin();
```

```
Serial.println("HTTP server started");

}

void loop()
{
    server.handleClient();
    Blynk.run();
    timer.run();
}
```

2. Phụ lục hình ảnh

Hình 1.1: Thống kê số thiết bị IoT từ năm 2015.....	5
Hình 1.2: Xếp hạng xu thế công nghệ năm 2019.....	6
Hình 2.1: Kit NodeMCU.....	9
Hình 2.2: Module ESP8266.....	10
Hình 2.3: Cảm biến nhiệt độ, độ ẩm DHT11.....	12
Hình 2.4: Cảm biến khí gas MQ2.....	14
Hình 2.5: Cảm biến áp suất BMP180.....	15
Hình 2.6: Cấu trúc giao tiếp HTTP.....	17
Hình 3.1: Sơ đồ khối tổng quát.....	19
Hình 3.2: Sơ đồ mạch chi tiết.....	20
Hình 3.3: Chính lưu toàn sóng và tụ điện cân bằng.....	21
Hình 4.1: Giao diện ứng dụng Arduino IDE.....	24
Hình 4.2: Ba phần chính của Blynk.....	25
Hình 4.3: Lưu đồ thuật toán đề tài.....	26
Hình 5.1: Kết quả thi công phần cứng tổng quát.....	27
Hình 5.2: Kết quả thi công phần cứng chi tiết.....	28
Hình 5.3: Board mạch được vẽ trên phần mềm Proteus.....	28
Hình 5.4: Kết quả giao diện Web Server.....	29
Hình 5.5: Kết quả giao diện Blynk.....	30