

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
KHOA ĐIỆN – ĐIỆN TỬ**

**ĐỒ ÁN TỐT NGHIỆP
ĐẠI HỌC**



**NGÀNH : CÔNG NGHỆ KỸ THUẬT ĐIỆN – ĐIỆN TỬ
CHUYÊN NGÀNH : KỸ THUẬT ĐIỆN TỬ**

ĐỀ TÀI:

**GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ ĐIỆN
TRONG CÔNG NGHIỆP DÙNG PHƯƠNG
THỨC MODBUS RTU**

Người hướng dẫn: TS. Nguyễn Linh Nam

Sinh viên thực hiện: Hoàng Như Cương

Msv:1811505120202 Lớp: 18D4

Nghiêm Sỹ Bảo

Msv:1811505120301 Lớp: 18D4

Đà Nẵng, Tháng 02/2022

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
KHOA ĐIỆN – ĐIỆN TỬ**

**ĐỒ ÁN TỐT NGHIỆP
ĐẠI HỌC**



**NGÀNH : CÔNG NGHỆ KỸ THUẬT ĐIỆN – ĐIỆN TỬ
CHUYÊN NGÀNH : KỸ THUẬT ĐIỆN TỬ**

ĐỀ TÀI :

**TÊN ĐỀ TÀI : GIÁM SÁT VÀ ĐIỀU KHIỂN
THIẾT BỊ ĐIỆN TRONG CÔNG NGHIỆP
DÙNG PHƯƠNG THỨC TRUYỀN MODBUS
RTU**

Người hướng dẫn: TS. Nguyễn Linh Nam

Sinh viên thực hiện: Hoàng Như Cương

Msv:1811505120202 Lớp: 18D4

Nghiêm Sỹ Bảo

Msv:1811505120301 Lớp: 18D4

Đà Nẵng, tháng 02/2022

TÓM TẮT

Tên đề tài: **GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG CÔNG NGHIỆP DÙNG PHƯƠNG THỨC TRUYỀN MODBUS RTU**

Giảng viên hướng dẫn : **TS. Nguyễn Linh Nam**

Sinh viên thực hiện : **Hoàng Như Cương** Mã SV : **1811505120202** Lớp: **18D4**

Nghiêm Sỹ Bảo Mã SV : **1811505120301** Lớp: **18D4**

Nhóm em tìm hiểu và tiến hành thực hiện đề tài “ Giám sát và điều khiển thiết bị điện trong công nghiệp dùng phương thức truyền MODBUS RTU “ với phương thức MQTT để điều khiển và giám sát nhiệt độ, độ ẩm, đo dòng, điện áp,..các thiết bị điện sử dụng trong công nghiệp thông qua Web server và di động thông qua mạng Wifi. Sử dụng STM32F103C8T6 truyền qua ESP8266 dùng MODBUS RTU (RS485), sau đó ESP8266 truyền dữ liệu lên Website bằng giao thức MQTT.

Nội dung mà nhóm thực hiện gồm : Nghiên cứu 1 cách tổng quan các linh kiện liên quan và các cơ sở lý thuyết cơ bản, thiết kế và xây dựng được phần mềm, phần cứng. Tiếp theo kiểm tra từng phần một để xem tín hiệu phần cứng so với Server hay phần mềm so với Server , rồi từ liên kết và tạo thành một hệ thống, từ phần mềm sang phần cứng và ngược lại.

Ở trong đề tài này, nhóm sẽ tiến hành và đề xuất ra các phương án kỹ thuật để thiết kế và vận hành động cơ trong công nghiệp.

Đề tài nhóm em gồm có 3 chương :

Chương 1 : Tổng quan :

- Tình hình nghiên cứu khoa học kỹ thuật trong và ngoài nước
- Tìm hiểu về Smart Factory
- Lý do chọn đề tài
- Phạm vi và đối tượng nghiên cứu

Chương 2 : Cơ sở lý thuyết :

- Tìm hiểu về IoT
- Tìm hiểu về Web Server
- Tìm hiểu về phương thức Modbus RTU
- Tìm hiểu về Arduino
- Giới thiệu phần cứng

Chương 3:

- Giới thiệu sơ đồ khối hệ thống

- Tính toán và thiết kế hệ thống
- Thi công sản phẩm
- Kết quả

Mục đích mà nhóm mong muốn hướng đến sau khi sản phẩm hoàn thành là:

- Biết cách lập trình phần mềm Arduino ide, phương thức truyền Modbus RTU, phương thức MQTT, biết được phương pháp để điều khiển, giám sát thiết bị .
- Thiết kế được một Web Server để giám sát trên máy tính và di động.
- Xây dựng được mô hình mô phỏng.
- Giúp chúng ta có được kỹ năng tìm kiếm tài liệu, nghiên cứu.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Giảng viên hướng dẫn: TS. Nguyễn Linh Nam

Sinh viên thực hiện: HOÀNG NHƯ CƯỜNG

Mã SV: 1811505120202

NGHIÊM SỸ BẢO

Mã SV: 1811505120301

1. Tên đề tài : GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG CÔNG NGHIỆP DÙNG PHƯƠNG THỨC TRUYỀN MODBUS RTU

2. Các số liệu, tài liệu ban đầu:

Số liệu :

- Sử dụng các giao thức để truyền dẫn và đồng bộ hóa theo thời gian .
- Sử dụng STM32F1C8T6 để làm board điều khiển chính của khối Node.
- Sử dụng cảm biến điện áp, cảm biến đo dòng, cảm biến nhiệt độ, cảm biến độ ẩm để lấy dữ liệu.
- Sử dụng Modul MP1584 để hạ áp
- Sử dụng Board mạch Node MCU ESP32 cho phép kết nối máy tính thông qua các giao thức, cổng kết nối giao tiếp SPI..và sử dụng mạng Wifi.
- Các tín hiệu hiển thị đến người dùng là dạng số và biểu đồ cập nhật theo thời gian.

Tài liệu ban đầu:

- Tìm hiểu các thông tin qua các trang mạng,và thao khảo các tài liệu
- Tra cứu thông tin tin linh kiện
- Tài liệu về công nghệ IOT
- Cộng đồng arduino Việt Nam.

3. Những nội dung chính :

Tìm hiểu cũng như thiết kế và chế tạo hệ thống giám sát và điều khiển các thiết bị điện sử dụng trong công nghiệp thông qua Wedserver.

Chương 1 : Tổng quan

Trình bày về đặt vấn đề dẫn nhập lý do chọn đề tài , mục tiêu , nội dung nghiên cứu, các giới hạn thông số và bố cục đồ án

Chương 2 : Cơ Sở Lý Thuyết

Trình bày về các lý thuyết có liên quan đến các vấn đề mà đề tài sẽ dùng để thực hiện thiết kế, thi công cho đề tài .

Chương 3 : Thiết Kế và Thi Công

Giới thiệu tổng quan về các yêu cầu của đề tài mà mình thiết kế thiết kế gồm những phần nào. Thiết kế sơ đồ khối hệ thống , sơ đồ nguyên lý từng khối và tính toán thiết kế mạch .

Trình bày cơ sở lí thuyết , các linh kiện , mạch in , quá trình thực hiện ,lắp đặt hệ thống . Vẽ lưu đồ thuật toán , chương trình cho sản phẩm .

4. Các sản phẩm dự kiến

- Hoàn thành sản phẩm có thể giám sát đo các tham số yêu cầu đề ra.
- Báo cáo tổng kết đồ án tốt nghiệp

5. Ngày giao đồ án: 05/10/2021

6. Ngày nộp đồ án: 20/1/2022

Đà Nẵng, ngày 18 tháng 02 năm 2022

Trưởng Bộ môn

Người hướng dẫn

ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

Giảng viên hướng dẫn: TS. Nguyễn Linh Nam

Sinh viên thực hiện: HOÀNG NHƯ CƯỜNG

Mã SV: 1811505120202

NGHIÊM SỸ BẢO

Mã SV: 1811505120301

1. Tên đề tài: GIÁM SÁT VÀ ĐIỀU KHIỂN THIẾT BỊ ĐIỆN TRONG CÔNG NGHIỆP DÙNG PHƯƠNG THỨC TRUYỀN MODBUS RTU VÀ GIAO THỨC MQTT

Thời gian thực hiện: Từ ngày: 04/10/2021 đến ngày: 20/01/2022.

2. Mục tiêu

Với đề tài: “ Giám sát và điều khiển thiết bị điện trong công nghiệp dùng phương thức truyền MODBUS ”, đây là đề tài bước đầu tạo cơ sở nền tảng kiến thức cơ bản để từ đó tìm tòi phát triển ứng dụng khác trong đời sống hằng ngày. Sau khi hoàn thành đồ án nhóm cần đạt các mục tiêu:

- Biết cách lập trình phần mềm STM32CUBE IDE và ARDUINO IDE, biết được phương pháp để điều khiển và giám sát thiết bị điện công nghiệp.
- Thiết kế được giao diện để quản lý và giám sát trên máy tính và di động.
- Xây dựng được mô hình mô giám sát điều khiển thiết bị.

3. Nội dung chính

- Tạo được giao diện để điều khiển cũng như giám sát được hệ thống .
- Viết chương trình giao tiếp giữa các cảm biến, truyền nhận từ các khối với nhau ổn định và chính xác nhất.
- Thiết kế và thi công mô hình.
- Viết báo cáo.

4. Kết quả dự kiến đạt được

Nhóm mong muốn từ việc tìm hiểu cũng như thực hiện đề tài để đạt được một số thành công nhất định , thiết kế được một mạch điều khiển đẹp và tối ưu và xây dựng được mô hình cụ thể :

- Hiểu và biết cách sử dụng được các cảm biến nhiệt độ, độ ẩm , cảm biến dòng , cảm biến điện áp .
- Hiểu và biết sử dụng STM32F103C8T6.
- Tìm hiểu và nghiên cứu về giao thức MQTT .
- Tìm hiểu và cách sử dụng module MP1584 ,...
- Tìm hiểu và xây dựng giao diện Website .
- Điều khiển các thiết bị công suất nhỏ được thiết kế bởi mạch ứng dụng từ Relay.
- Thiết kế , chọn linh kiện để hoàn thiện được board mạch chính đủ áp và phù hợp với yêu cầu đưa ra từ trước đó .
- Thi công thành công mô hình giám sát và điều khiển được thiết bị trong công nghiệp.
- Giám sát được các giá trị nhiệt độ, độ ẩm, dòng điện, áp suất qua Máy tính và di động từ đó điều khiển các thiết bị điện trong công nghiệp .
- Lập trình cho hệ thống chạy ổn định qua nhiều lần kiểm tra.
- Mô hình hoạt động ổn định.

5. Tiến độ thực hiện

TT	Thời gian	Nội dung công việc	Kết quả dự kiến đạt được
1	Tuần 1	Gặp giảng viên hướng dẫn và trao đổi về đề tài đồ án tốt nghiệp.	
2	Tuần 2	Viết đề cương và lịch trình thực hiện đồ án tốt nghiệp.	
3	Tuần 3	Tìm hiểu đề tài	
4	Tuần 4	Nghiên cứu nguyên lý hoạt động của đề tài	
5	Tuần 5	Vẽ sơ đồ nguyên lý , sơ đồ khối	
6	Tuần 6	Viết chương trình, kiểm tra các chức năng điều khiển trên máy tính ,di động.	
7	Tuần 7	Viết chương trình, kiểm tra các chức năng điều khiển trên máy tính ,di động.	
8	Tuần 8	Thi công sản phẩm và lắp ráp mô hình	
9	Tuần 9	Thi công sản phẩm và lắp ráp mô hình	

10	Tuần 10	Thi công sản phẩm và lắp ráp mô hình	
11	Tuần 11	Hoàn thiện mô hình, đóng gói hệ thống và chạy thử nghiệm	
12	Tuần 12	Hoàn thiện mô hình, đóng gói hệ thống và chạy thử nghiệm	
13	Tuần 13	Chạy thử nghiệm và cân chỉnh toàn hệ thống. Đánh giá kết quả đạt được, viết báo cáo.	
14	Tuần 14	Chạy thử nghiệm và cân chỉnh toàn hệ thống. Đánh giá kết quả đạt được, viết báo cáo.	
15	Tuần 15	Chạy thử nghiệm và cân chỉnh toàn hệ thống. Đánh giá kết quả đạt được, viết báo cáo.	

Đà Nẵng, ngày 18 tháng 02 năm 2022.

BỘ MÔN DUYỆT

NGƯỜI HƯỚNG DẪN

SINH VIÊN

LỜI CẢM ƠN

Để hoàn thành được đề tài này , lời đầu tiên cho phép nhóm chúng em được gửi những lời cảm ơn chân thành đến tất cả quý thầy cô trường Đại Học Sư Phạm Kỹ Thuật Đà Nẵng nói chung và các thầy cô trong Khoa Điện - Điện Tử nói riêng , và các thầy cô đã tận tình chỉ dạy hướng dẫn , trang bị cho nhóm những kiến thức nền tảng và chuyên ngành quan trọng , qua đó giúp nhóm chúng em có được cơ sở lý thuyết vững vàng và tạo điều kiện tốt nhất cho chúng e để thực hiện tốt đề tài .

Đặc biệt , nhóm em xin chân thành cảm ơn thầy Nguyễn Linh Nam đã tận tình giúp đỡ và đưa ra những định hướng nghiên cứu cũng như hướng giải quyết một số vấn đề để nhóm chúng em hoàn thành tốt đề tài . Trong thời gian làm việc với thầy , chúng em đã không ngừng tiếp thu thêm nhiều kiến thức được chỉ dạy từ thầy , luôn luôn thể hiện thái độ nghiêm túc , hiệu quả và đây là điều quan trọng và rất cần thiết trong quá trình học tập và làm việc của chúng em sau này.

Tuy nhóm đã cố gắng hết sức , nhưng do điều kiện thời gian và kinh nghiệm thực tế của nhóm còn chưa nhiều , nên việc đề tài còn một số thiếu sót . Vì vậy , nhóm chúng em rất mong nhận được sự đóng góp ý kiến từ thầy cô giáo .

Xin chân thành cảm ơn !

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

Hoàng Như Cương

Nghiêm Sỹ Bảo

CAM ĐOAN

Đề tài này là do nhóm em tự thực hiện dưới sự hướng dẫn của thầy Nguyễn Linh Nam dựa vào một số tài liệu trước đó và không sao chép từ tài liệu hay công trình đã có trước đó.

Những thống số hay số liệu trong đề tài chưa được bảo vệ hay báo cáo trước đó .

Những trích dẫn , ghi chú , số liệu , hình ảnh trong đề tài được ghi tên và nguồn gốc rõ ràng.

Chúng em cam đoan những lời ở trên là đúng sự thật và trung thực , nếu có bất kì phát giác nào thì nhóm xin chịu trách nhiệm .

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

Hoàng Như Cương

Nghiêm Sỹ Bảo

MỤC LỤC

TÓM TẮT.....	i
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	iii
ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP	vii
LỜI CẢM ƠN.....	x
CAM ĐOAN.....	xi
MỤC LỤC	xii
DANH SÁCH HÌNH VẼ VÀ CÁC BẢNG	xiv
CÁC CHỮ VIẾT TẮT VÀ CÁC KÝ HIỆU	xvi
MỞ ĐẦU	xvii
CHƯƠNG I : TỔNG QUAN.....	1
1.1. Tình hình nghiên cứu	1
1.1.1. Giới thiệu tình hình nghiên cứu ngoài nước hiện nay.	1
1.1.2. Giới thiệu tình hình khoa học công nghệ trong nước.	1
1.1.3. Ứng dụng	7
1.2. Tìm hiểu về Smart Factory	8
1.2.1. Khái niệm	8
1.2.2. Tính năng	8
1.2.3. Đặc trưng của nhà máy thông minh.....	9
1.3. Hướng chọn đề tài	12
1.4. Mục tiêu đề tài.....	12
1.5. Đối tượng và phạm vi nghiên cứu.....	13
1.6. Bố cục đồ án	13
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ	14
2.1. Khái quát về IoT trong công nghiệp.....	14
2.1.1. Khái niệm.....	14
2.1.2. IoT ở trong công nghiệp	14
2.2. Tìm hiểu về Web Server.....	15
2.2.1. Khái niệm về Web Server.....	15
2.2.2. Giao thức MQTT	15
2.3. Phương thức Modbus RTU (RS485).....	16
2.3.1. Modbus là gì?.....	16

2.3.2. Ứng dụng của Modbus RTU.....	17
2.2.3. Ưu điểm và nhược điểm	18
2.4. Phần mềm Aduino.....	19
2.4.1. Aduino là gì?	19
2.4.2. Phần mềm Visual Studio.....	21
2.5. Giới thiệu phần cứng	22
2.5.1. Giới thiệu về STM32F103C8T6.....	22
2.5.2. Giới thiệu về ESP8266.....	24
2.5.3. Relay	26
2.5.4. Module Mp1584EN	27
2.5.5. Cảm biến nhiệt độ, độ ẩm DHT11	29
2.5.6. Cảm biến điện áp ZMPT101B	30
CHƯƠNG 3 : THIẾT KẾ VÀ THI CÔNG	32
3.1. Sơ đồ khối hệ thống.....	32
3.1.1. Yêu cầu đề tài	32
3.1.2. Thiết kế sơ đồ khối đề tài.....	32
3.1.3. Chức năng từng khối.....	33
3.2. Thiết kế đề tài	35
3.2.1. Thiết kế các cảm biến	35
3.2.2. Thiết kế khối xử lý.....	37
3.2.3. Khối giao tiếp RS-485	42
3.2.4. Khối trung tâm dữ liệu(Web server).....	43
3.3. Thi công sản phẩm.....	45
3.3.1. Mạch in	45
3.3.2. Mạch sau khi thi công.....	49
3.4. Mô hình sản phẩm và quá trình hoạt động	51
3.4.1. Mô hình sản phẩm.....	51
3.4.2. Quá trình hoạt động của hệ thống.....	52
KẾT LUẬN	54
TÀI LIỆU THAM KHẢO	56
PHỤ LỤC	59

DANH SÁCH HÌNH VẼ VÀ CÁC BẢNG

BẢNG 2.1: Các thông số của ESP8266.

BẢNG 2.2: Các thông số của Relay [13].

BẢNG 2.3: Các thông số kỹ thuật của module Mp1584EN.

BẢNG 2.4: Thông số kỹ thuật cảm biến độ ẩm , nhiệt độ DHT11 [16].

BẢNG 2.5: Các thông số kỹ thuật của cảm biến điện áp ZMPT101B [17].

BẢNG 3.1: Mức tiêu thụ điện năng của 1 số linh kiện .

.....

HÌNH 1.1. Trí tuệ nhân tạo (AI) [1].

HÌNH 1.2. Tốc độ tăng năng suất lao động của Việt Nam 2006-2015 [3].

HÌNH 1.3. Tổng chi cho quốc gia cho việc phát triển và nghiên cứu [3].

HÌNH 1.4. Ứng dụng khoa học công nghệ vào đời sống [4].

HÌNH 1.5. Hình ảnh lắp đặt thực tế giải pháp SEEACT tại nhà máy sản xuất [6].

HÌNH 1.6. Đặc trưng của Smart Factory [7].

HÌNH 1.7. Nhà máy thông minh [7].

HÌNH 1.8. Phương thức truyền Modbus [14].

HÌNH 2.1. Khái niệm về IoT [9].

HÌNH 2.2. MQTT trong IoT [13].

HÌNH 2.3. Modbus RTU đưa lên internet [14].

HÌNH 2.4. Modbus slave truyền về máy tính thông qua RS485 [14].

HÌNH 2.5. Biểu tượng phần mềm Arduino [15].

HÌNH 2.6. Giao diện khi khởi tạo.

HÌNH 2.7. Tạo file mới.

HÌNH 2.8. Biểu tượng phần mềm Visual Studio [16].

HÌNH 2.9. Cấu tạo của STM32 [17].

HÌNH 2.10. Hình ảnh ESP8266 node MCU [18].

HÌNH 2.11. Một số Relay thông dụng [19].

HÌNH 2.12. Module Mp1584EN [21].

HÌNH 2.13. Cảm biến nhiệt độ , độ ẩm DHT11 [23].

HÌNH 2.14. Cảm biến điện áp ZMPT101B [24].

HÌNH 2.15. Cấu tạo của cảm biến điện áp ZMPT101B [24].

HÌNH 3.1. Sơ đồ khối hệ thống.

HÌNH 3.2. Adapter 12V-2A.

HÌNH 3.3. Cảm biến dòng điện ACS712 [26].

HÌNH 3.4. Giao diện làm việc của phần mềm Visual Studio Code.

HÌNH 3.5. Lưu đồ thuật toán khối cảm biến .

HÌNH 3.6. Cửa sổ làm việc của phần mềm STM32 cubeide.

HÌNH 3.7. Sơ đồ nguyên lý STM32 đọc cảm biến và điều khiển.

HÌNH 3.8. Sơ đồ nguyên lý ESP8266 dùng Modbus RTU RS-485.

HÌNH 3.9. Lưu đồ thuật toán Gateway.

HÌNH 3.10. Mức điện áp của RS485 và UART [25].

HÌNH 3.11. Sơ đồ mạch UART RS-485 [27].

HÌNH 3.12. Lưu đồ thuật toán của Web server.

HÌNH 3.13. Mạch in khối xử lý.

HÌNH 3.14. Mạch 3D khối xử lý.

HÌNH 3.15. Mạch in khối giao tiếp từ STM lên Webserver.

HÌNH 3.16. Mạch 3D khối giao tiếp

HÌNH 3.17. Mạch 1 sau khi thi công.

HÌNH 3.18. Mạch 2 sau khi thi công.

HÌNH 3.19. Hình ảnh mô hình sản phẩm.

HÌNH 3.20. Hình ảnh Website.

HÌNH 3.21. Giao diện điều khiển thiết bị.

CÁC CHỮ VIẾT TẮT VÀ CÁC KÝ HIỆU

CÁC KÝ HIỆU :

< : Dấu bé hơn .

% : Phần trăm .

... : Còn nữa .

CHỮ VIẾT TẮT :

IoT: Internet of Things.

MQTT : Là giao thức hoạt động theo cơ chế xuất bản / đăng ký .

DHT11: Cảm biến nhiệt độ, độ ẩm.

MES:Hệ thống máy tính được sử dụng trong sản xuất.

SPI: Serial Peripheral Bus.

ESP8266: Module MCU.

STM32: STM32F103C8T6.

MỞ ĐẦU

Ở trong thời kì mà công nghệ phát triển không ngừng như hiện nay, nền khoa học kĩ thuật của nước ta cần phải bắt kịp xu hướng chung của thế giới nếu không muốn bị tụt hậu. Khi mà các thiết bị điện tử, các linh kiện luôn không ngừng đổi mới, ở các khu công nghiệp thì việc giám sát và điều khiển rất quan trọng để sản phẩm được tối ưu và đúng tiến độ, nên nhóm em đã quyết định chọn đề tài “ Giám sát và điều khiển thiết bị điện trong công nghiệp dùng phương thức truyền MODBUS RTU và giao thức MQTT ”. Tuy đề tài giám sát và điều khiển thiết bị điện trong công nghiệp không phải là quá mới, nhưng nó đủ đáp ứng được một lượng kiến thức về điện tử và được ứng dụng rộng rãi trong công nghiệp.

Đề tài nhóm em được thực hiện dựa trên sự hướng dẫn tận tình của thầy Nguyễn Linh Nam , cũng như sự hỗ trợ về thiết bị của nhà trường, qua tìm hiểu trên mạng. Thực hiện đề tài rất mong nhận được sự góp ý của hội đồng bảo vệ, thầy cô, và các bạn sinh viên để có thể nâng cao chất lượng của đồ án.

Nội dung đề tài bao gồm : tổng quan đề tài , cơ sở lý thuyết, xây dựng đề tài gồm có phần mềm cùng như phần cứng .

Ở trong đề tài này , nhóm em sẽ tìm hiểu và xây dựng một hệ thống giám sát và điều khiển thiết bị điện trong công nghiệp . Nhưng do kiến thức và tìm hiểu còn có một số hạn chế nên những thứ nhóm đưa ra có thể đưa được tối ưu nhất hay còn một số thiếu sót nên mong được quý thầy cô đóng góp ý kiến để nhóm phát triển sản phẩm hơn trong tương lai .

Đề tài gồm có 3 chương chính :

Chương 1 : Tổng quan :

- Tình hình nghiên cứu khoa học kỹ thuật trong và ngoài nước
- Tìm hiểu về Smart Factory
- Lý do chọn đề tài
- Phạm vi và đối tượng nghiên cứu

Chương 2 : Cơ sở lý thuyết :

- Tìm hiểu về IoT
- Tìm hiểu về Web Server
- Tìm hiểu về phương thức Modbus RTU
- Tìm hiểu về Arduino
- Giới thiệu phần cứng

Chương 3:

- Giới thiệu sơ đồ khối hệ thống

- Tính toán và thiết kế hệ thống
- Thi công sản phẩm
- Kết quả

Nhóm thực hiện đề tài với mục tiêu cụ thể như sau :

Với đề tài: “ Giám sát và điều khiển thiết bị điện trong công nghiệp dùng phương thức truyền MODBUS ”, đây là đề tài bước đầu tạo cơ sở nền tảng kiến thức cơ bản để từ đó tìm tòi phát triển ứng dụng khác trong đời sống hằng ngày. Sau khi hoàn thành đồ án nhóm cần đạt các mục tiêu:

- Biết cách lập trình phần mềm Arduino, phương thức truyền Modbus RTU, phương thức MQTT, biết được phương pháp để điều khiển, giám sát động cơ .
- Tạo được ứng dụng và Web để điều khiển .
- Xây dựng được mô hình mô phỏng.
- Giúp chúng ta có được kỹ năng tìm kiếm tài liệu, nghiên cứu.

CHƯƠNG I : TỔNG QUAN

1.1. Tình hình nghiên cứu

1.1.1. Giới thiệu tình hình nghiên cứu ngoài nước hiện nay.

Ở những năm trở lại đây, các ngành kỹ thuật của các nước trong khu vực cũng như thế giới có những bước chuyển mình và phát triển mạnh mẽ. Khoa học công nghệ của thế giới thay đổi từng ngày từng giờ và tạo ra những sản phẩm ứng dụng đột phá phục vụ được những nhu cầu cho con người như : Trí tuệ nhân tạo, Người máy, công nghệ nano , tự động hoá , lưu trữ dữ liệu Big Data ,...



Hình 1.1. Trí tuệ nhân tạo (AI) [1].

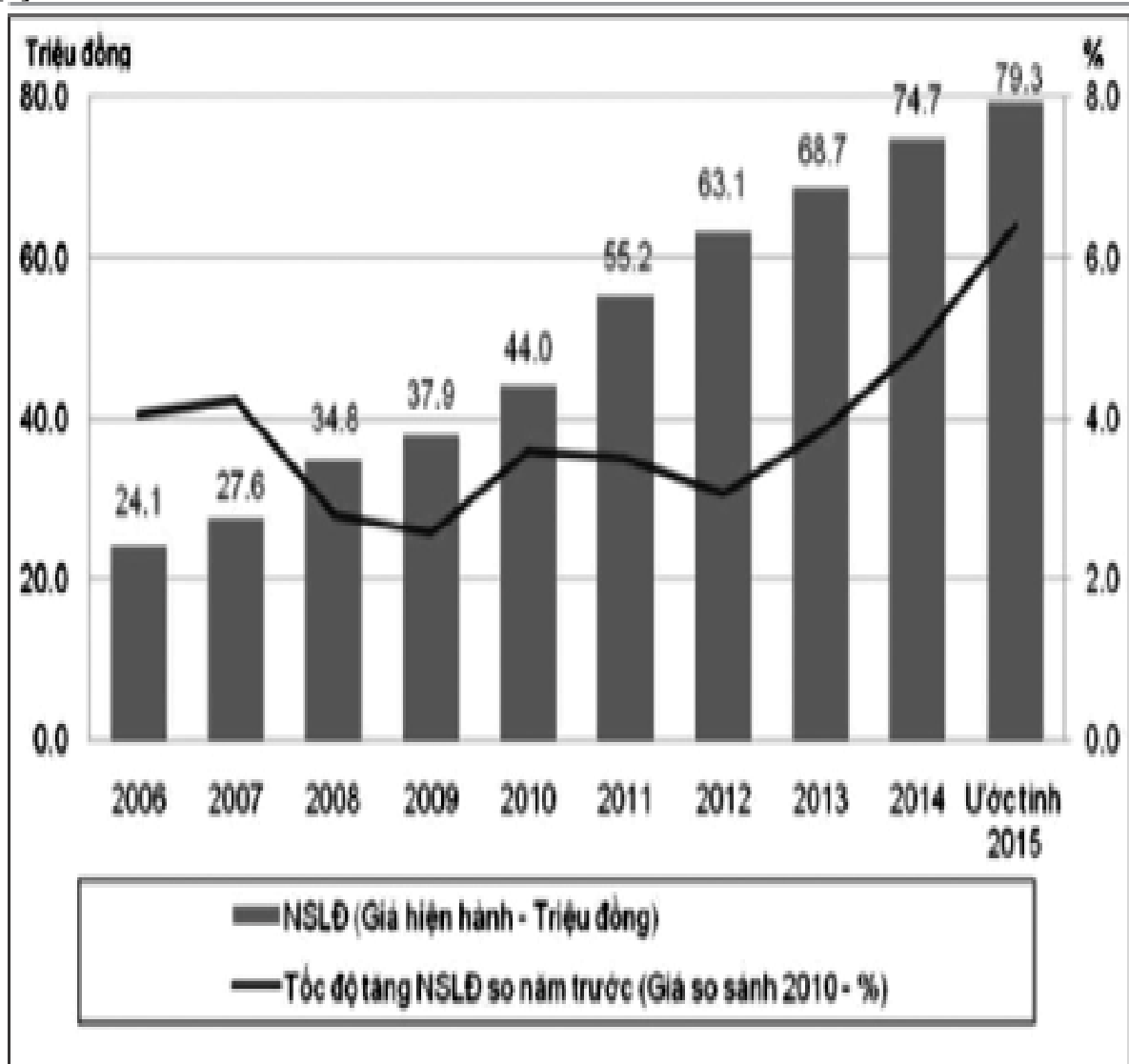
1.1.2. Giới thiệu tình hình khoa học công nghệ trong nước.

Trong những năm trở lại đây, các ngành kỹ thuật của các nước ta đã thay đổi rõ và đang bước vào giai đoạn phát triển không ngừng , cùng mức đời sống của con người thì đã nâng lên nhiều nên nhu cầu sử dụng những thiết bị thông minh và giám sát có tính chính xác cao để phục vụ cho công việc cá nhân hoặc tập thể công ty lớn nhưng chúng ta vẫn còn đang chịu ảnh hưởng rất nhiều vào công nghệ của ngoài nước nên nhóm em thực hiện đề tài đã quyết định lựa chọn đề tài “ Giám sát và điều khiển thiết bị điện trong công nghiệp sử dụng phương thức MODBUS RTU ” bằng giao thức MQTT dựa trên phần mềm lập trình arduino và kit vi xử lý arduino để chọn những phương án tốt nhất , hạn chế khuyết điểm để hoàn thành đề tài một cách tốt nhất từ đó ứng dụng vào thực tế một cách linh hoạt. Như chúng ta đã biết việc quản lý dữ liệu và giám sát thiết bị là rất

cần thiết trong đời sống. Hiện nay, chúng ta cũng đã có rất nhiều công cụ để làm việc đó với giá thành tương đối và kết quả thu được cũng rất đáng được chú ý, ở đề tài này nhóm em sử dụng các cảm biến để giám sát thiết bị, cụ thể ở đây là giám sát tốc độ, nhiệt độ, độ ẩm, đo dòng, đo điện áp của động cơ và dựa vào những thiết bị giám sát để điều khiển thiết bị theo từng mục đích của người sử dụng.

Có rất nhiều thứ ảnh hưởng đến quá trình gia NSLĐ, nó đóng vai trò quyết định trong trình độ khoa học công nghệ. Mà ở đó chiến lược phát triển kinh tế xã hội của Việt Nam trong giai đoạn 2006 → 2020 khẳng định: “Phát triển khoa học và công nghệ là quốc sách hàng đầu, đóng vai trò then chốt cho việc phát triển lực lượng sản xuất hiện đại, bảo vệ tài nguyên, môi trường, tăng năng suất lao động, tốc độ phát triển và sức cạnh tranh của nền kinh tế”. Ở đây chúng ta bàn đến khía cạnh phát triển khoa học công nghệ để tạo động lực nhằm tăng năng suất lao động của nước nhà trong điều kiện cách mạng công nghiệp 4.0.

Những thành tựu của Việt Nam trong nền khoa học và công nghệ vào năng suất lao động [2]:



Hình 1.2. Tốc độ tăng năng suất lao động của Việt Nam 2006-2015 [3].

Ở “BÁO ĐIỆN TỬ CHÍNH PHỦ NƯỚC CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM” có viết nền kinh tế Việt Nam năm 2015 có năng suất lao động tăng 6,4% với với 2014 , còn trong những năm 2006 – 2015 thì tăng 3,9% trên năm ,còn tính theo giá hiện hành thì đạt 79,3 triệu / lao động (xấp xỉ 3657 USD / lao động).

Theo Nguyễn Thị Minh Thu viết ở tạp chí tài chính online thì năng suất lao động của Việt Nam và các nước ASEAN đã càng gần nhau và thu hẹp. Dù vậy , năng suất lao động của nước ta vẫn còn khá thấp so với những nước khác trong khu vực.

Việt Nam có năng suất lao động ngành nông , lâm , thuỷ sản thua từ 1,7 → 2 lần so với Thái Lan , Philippinnes và Indonesia. Tương tự thì Việt Nam cũng thua nhiều với các nước trong khu vực về năng suất lao động trong ngành công nghiệp chế biến ,chế tạo.

Có thể nói , mặc dù đã được cải thiện khá nhiều nhưng sau 30 năm đổi mới thì năng suất lao động của Việt Nam còn tăng khá chậm ,còn có sự chênh lệch với các nước trong khu vực lẫn thế giới và có nguy cơ bị rơi rụng ,làm cho chúng ta có nguy cơ tụt lại xa hơn .

Chậm phát triển trình độ khoa học công nghệ là một trong những nguyên nhân làm năng suất lao động tăng chậm , làm cho đóng góp của khoa học công nghệ vào sự gia tăng năng suất lao động của chúng ta trong những năm gần đây là khá thấp. Theo những số liệu thống kê từ Tổ chức năng suất châu Á , trong những năm 2010→2013 tốc độ tăng năng suất lao động của Việt Nam khoảng 4,3 % / năm, trong đó đóng góp của yếu tố tổng hợp khoảng 27%và đóng góp tăng cường vốn khoảng 75% . So với nhiều nước châu Á đóng góp của tăng năng suất yếu tố tổng hợp vào tăng năng suất lao động của Việt Nam còn khá thấp .

Tình hình hiện tại của sự phát triển khoa học công nghệ của Việt Nam :

Một trong những tác động quan trọng tới việc nâng cao năng suất lao động là thành tựu về kinh tế - xã hội , việc đó đã được chứng minh qua 2 phương diện : Thứ nhất là tạo hoặc tìm ra được nguyên liệu mới hay sản phẩm mới để có khả năng cạnh tranh tốt hơn , từ đó giá trị tăng cao hơn, thay thế việc nhập khẩu hàng hoá từ nước ngoài với chi phí rẻ hơn . Thứ hai là tối ưu hoá , cải tiến các quá trình kinh doanh sản xuất , qua đó thay thế dần sức lao động bằng máy móc để giảm bớt sức lao động , lao động thủ công để tăng năng suất lao động và rút gọn thời gian làm việc .

Nhận thấy được sự quan trọng về việc phát triển khoa học công nghệ tới việc tăng năng suất lao động ở Việt Nam , nhà nước ban hành nhiều chính sách , chủ trương , pháp luật với mục đích phát triển khoa học công nghệ. Ở giai đoạn 2011→ 2020 về chiến lược phát triển khoa học công nghệ quy định rằng việc tăng đầu tư cho khoa học công nghệ ở mức 1,5 % GDP ở năm 2015 và hơn 2% ở năm 2020 . Nhà nước cũng ban hành các luật để tạo nguồn nhân lực khoa học công nghệ , hỗ trợ chuyển giao công nghệ ,...

Mục tiêu đặt ra gồm : Làm chủ công nghệ và đào tạo các kỹ sư (khoảng 80000 kỹ sư) .Nâng cấp và cải tiến công nghệ với tốc độ 15% một năm , kỹ thuật viên và nhà quản lý làm việc trong các doanh nghiệp nhỏ trong quản lí công nghệ và quản trị . Luật khoa

học công nghệ năm 2013 còn quy định việc hỗ trợ tài chính từ ngân sách nhà nước để thực hiện những hoạt động khoa học công nghệ ,...

Nếu doanh nghiệp thực hiện các dự án ứng dụng các kết quả khoa học để tạo ra các sản phẩm mới hay tăng năng suất , khả năng cạnh tranh sản phẩm và chất lượng sản phẩm ,...

Mặc dù chủ trương chính sách ưu tiên phát triển nhưng thực tế thì các chính sách trên còn nhiều hạn chế và năng lực khoa học công nghệ của Việt Nam sau 30 năm thay đổi ghi nhận nhưng thay đổi còn khá nhiều hạn chế so với các nước trong khu vực cũng như thế giới .

Đầu tiên ,tổng chi quốc gia cho nghiên cứu khoa học và phát triển công nghệ còn thấp so với những quốc gia khác trong khu vực cũng như thế giới .Theo các số liệu nghiên cứu năm 2014 thì tỉ trọng tổng chi quốc gia cho khoa học công nghệ /GDP trong năm 2013 là khoảng 88% , trong đó chiếm 44 % cho việc nghiên cứu và phát triển.

Nguồn đầu tư	Tổng chi quốc gia cho Nghiên cứu và Phát triển (GERD) (tỷ đồng)	Tỷ lệ cơ cấu đầu tư Nghiên cứu và Phát triển theo nguồn (%)
Tổng số	13.390,6	100,0
NSNN	7.591,6	56,7
Doanh nghiệp	5.594,3	41,8
Vốn nước ngoài	201,7	1,5

Hình 1.3. Tổng chi cho quốc gia cho việc phát triển và nghiên cứu [3].

Theo thống kê năm 2013, thì tỷ lệ chi quốc gia cho nghiên cứu và phát triển/GDP đạt 0,37%. Tỷ lệ chi cho nghiên cứu và phát triển/GDP của Việt Nam so với các nước là rất thấp. Đặc biệt NSNN chiếm 57% , nguồn đầu tư từ doanh nghiệp chiếm 42 % và nguồn nước ngoài là 1,5 % trong tổng chi quốc gia và phát triển [2].

Thứ 2 về đội ngũ khoa học công nghệ của nước ta tuy tăng về số lượng nhưng so với tổng số dân số thì còn khá thấp so với các nước trong khu vực. Trung bình cán bộ nghiên cứu trên vạn dân ở năm 2013 tính theo đầu người ở nước ta là khoảng 14 người .

Tỉ lệ đó thấp hơn Trung Quốc năm 2012 (15,4) và bằng 1/5 của Nhật Bản (70,3), bằng 1/6 của Hàn Quốc (82) và so với Singapore(75) là khoảng 1/5. Cùng với đó là chất lượng nguồn nghiên cứu của chúng ta chưa cao ,thiếu hụt chuyên gia giỏi ,cán bộ đầu ngành và các chuyên gia công nghệ .

Từ đó dẫn đến chất lượng nghiên cứu khoa học công nghệ của chúng ta năm 2014 xếp thứ 89 , trong khi đó thì các nước như Indonesia 46, Thái Lan 60, Malaysia xếp thứ 27 , Philippines xếp thứ 91. Tỉ lệ bằng sáng chế và ứng dụng / một triệu dân của Việt Nam năm 2014 xếp thứ 92 trên thế giới , trong khi đó Thái Lan 71 ,Malaysia , Philippines 84.

Thứ 3 , không được đồng bộ , chậm được đổi mới và trình độ công nghệ còn thấp. Tỉ lệ ứng dụng khoa học công nghệ vào sản xuất và đời sống còn hạn chế . Theo thống kê Công nghệ và cạnh tranh ngành chế biến , chế tạo giai đoạn 2009 -2012 thì chỉ có khoảng 11% số doanh nghiệp đã phát triển các loại hình công nghệ mới .

Về các hoạt động nghiên cứu và phát triển thì có 9% số doanh nghiệp hoạt động và khoảng 6% là cải tiến những công nghệ đang có sẵn . Đặc biệt thì 85% doanh nghiệp cho hay là không có bất kì chương trình phát triển hay cải tiến nào về công nghệ . Nền tảng của phương thức phát triển mới phụ thuộc vào mức độ của công nghệ mới và cũng như là công nghệ thông tin .

Còn theo Diễn đàn Kinh tế thế giới thống kê được thì Việt Nam chỉ xếp ở vị trí thứ 102 trên thế giới và trong giai đoạn từ năm 2008-2014 thì mức độ sẵn sàng trong áp dụng công nghệ mới của chúng ta đã tụt từ vị trí 71/134 ở năm 2008-2009 xuống vị trí 134/148 ở trong năm 2013-2014 và nó thấp hơn khá nhiều so với các nước Philippines (47), Indonesia (60), Malaysia (vị trí 37) và Thái Lan (75).

Thứ 4 , Tuy chúng ta đã ban hành nhiều chính sách thúc đẩy chuyển giao công nghệ nhưng hiệu quả và những chuyển giao công nghệ đó từ các doanh nghiệp Fdi vẫn còn khá thấp . Chủ yếu các chuyển giao công nghệ của chúng ta được diễn ra giữa các doanh nghiệp trong nước. [3].

Sau 35 năm thay đổi và phát triển, chúng cũng đã có cho mình ít nhiều những thành tựu quan trọng trong việc công nghiệp hoá – hiện đại hoá đất nước . Từ đó để đưa đất nước chúng ta từ là nước đang rất nghèo dần dần trở thành một quốc gia có thu nhập trung bình thấp và từng bước hội nhập với các nước trong khu vực và thế giới , từ đó mà đời sống của nhân dân ngày càng được nâng cao và cải thiện rõ ràng . Vị thế và tầm ảnh hưởng của Việt Nam trên trường quốc tế ngày càng rõ ràng .Để mang lại những thành quả to lớn này có đóng góp vào một phần không nhỏ và quan trọng của ngành Công thương với việc mà chúng ta đã từng bước khẳng định được là một trong những nước có vị thế và là trung tâm sản xuất công nghiệp của đông nam á nói riêng và của thế giới nói chung.

Chúng ta đã dần trở thành một trong những quốc gia có nền công nghiệp với sức cạnh tranh trên thế với nằm ở mức khá cao, thuộc một trong những quốc gia mà năng lực cạnh tranh ở công nghiệp nằm mức cao với vị trí thứ 44 của thế giới ở năm 2018

theo UNIDO . Ở thống kê trong những năm 1990→2018 chúng ta tăng 50 bậc , còn từ năm 2010→2018 tăng lên 23 bậc , nằm trong những nước tăng nhanh nhất thuộc khu vực ASEAN , tiến dần tới nhóm 4 nước có tiềm lực cạnh tranh mạnh nhất trong khối và có khả năng soán ngôi thứ 5 của philippines khi chỉ kém 0,001 điểm .

Ngành mà có tốc độ tăng trưởng nhanh nhất trong ngành kinh tế quốc dân . Theo thống kê thì nó đóng góp khoảng 31% GDP và là một trong những ngành chủ lực để xuất khẩu ra nước ngoài để dần đưa chúng ta trở thành quốc gia thứ 22 về xuất khẩu trên thế giới ở thống kê năm 2018 . Một số doanh nghiệp lớn như dệt may , da giày , điện tử , ... với một số các ngành mũi nhọn, một số ngành ưu tiên đi theo định hướng chiến lược dần trở thành những ngành công nghiệp lớn nhất đất nước , từ đó đưa Việt Nam dần dần hội nhập vào chuỗi giá trị toàn cầu .

Ở trong số 32 mặt hàng được xuất khẩu với kim ngạch lớn hơn 1 tỉ USD được thống kê năm 2019 thì hàng công nghiệp đạt 29 trên 32 mặt hàng , 5 trên 5 mặt hàng có kim ngạch xuất khẩu lớn hơn 10 tỉ USD ví dụ như đồ gỗ , da giày , dệt may , điện tử , thiết bị , máy móc . Trong đó thì những ngành công nghiệp mà có được vị trí vững vàng ở trên toàn cầu hiện nay như dệt may xếp thứ 7 về xuất khẩu , xếp thứ 3 về sản xuất và xếp thứ 2 về xuất khẩu , trong điện tử thì nằm ở vị trí 12 về xuất khẩu, còn riêng về xuất khẩu điện thoại thì nó nằm ở vị trí thứ 2 .

Ở thống kê về những doanh nghiệp lớn nhất Việt Nam ở năm 2019 , xét 10 doanh nghiệp lớn nhất thì có tới 8 doanh nghiệp nằm trong lĩnh vực công nghiệp , gồm có 7 doanh nghiệp nội địa và doanh nghiệp tư nhân cả nước chiếm tới 5 . Các lĩnh vực mà các doanh nghiệp tập trung chủ yếu là ô tô , dầu khí , điện , khoáng sản , sữa , thép và thực phẩm .

Trong quá trình tái thiết và đổi mới ngành công nghiệp thì cách làm mới mô hình tăng trưởng khá là quan trọng . Công nghiệp là ngành có năng suất lao động cao nhất trong các ngành kinh tế quốc dân mà ở đó trong các năm 2011 , 2015 , 2019 tăng lần lượt lên 26 % , 28 % , 29% .

Từ năm 2011 đến năm 2019 thì cơ cấu nội ngành công nghiệp giảm dần tỉ lệ khai khoáng công nghiệp từ 36 % xuống còn 26 % và tăng tỉ trọng công nghiệp chế biến chế tạo từ 50% lên 55% và trở thành tiên phong tăng trưởng chính cho các ngành công nghiệp .

Trong ngành công nghiệp đã có nhiều thay đổi tích cực để tiếp cận với công nghệ một cách tiên tiến , hiện đại hơn với những sự chuyển giao từ các ngành lao động như da dày , dệt may dần chuyển qua các ngành công nghiệp hiện đại cao hơn như điện thoại , máy tính , ô tô ...

Chúng ta nên tập trung đầu tư vào các ngành công nghiệp để nó được ngày càng phát triển rộng rãi lấy nền cốt đầu tư FDI và chuyển dịch cơ cấu phát triển của nước ta theo hướng càng ngày càng phát triển , hiện đại hoá. Theo thống kê thì nó chiếm tỉ trọng khoảng 71 % tổng vốn đầu tư vào những ngành kinh tế và khoảng 61% là vào công nghiệp chế biến, chế tạo .

Đề thúc đẩy sự phát triển của công nghiệp hoá , hiện đại hoá đất nước cùng với những mục tiêu đặt ra tới năm 2030 để trở thành một trong các nước hiện đại , nằm trong top 3 nước đi đầu khu vực ASEAN về công nghiệp , có 1 vài ngành công nghiệp có thể

cạnh tranh với các nước hàng đầu trong khu vực cũng như toàn thế giới . Hướng mục tiêu đến năm 2045 thì đất nước chúng ta sẽ trở thành nước có công nghiệp hiện đại . Bộ chính trị đã đưa ra nghị quyết số 23 – NQ/TW vào ngày 22/03 /2018 về hướng phát triển công nghiệp đất nước đến năm 2030 , tầm nhìn đến năm 2045 .

Theo những hướng đi và chỉ đạo của Đảng với chính sách công nghiệp quốc gia là gắn liền giữa những chính sách của cách ngành kinh tế khác trong đó có chính sách thương mại quốc gia tài chính tiền tệ , đào tạo , khoa học , công nghệ , ứng phó với biến đổi khí hậu và bảo vệ môi trường . Để có sự liên kết giữa quy hoạch phát triển các ngành kinh tế khác nhau để hình thành các cụm liên kết , vùng công nghiệp , khu công nghiệp , chuỗi giá trị công nghiệp , những mạng sản xuất trong số đó thì cụm liên kết ngành công nghiệp được coi là trọng tâm . Có sự kết hợp hoàn hảo giữa phát triển công nghiệp theo chiều rộng và sâu , đặc biệt quan trọng chiều sâu , nhằm tạo bước đột phá trong chiều sâu .Tận dụng điểm mạnh của nước ta là nước đang trong cơ cấu thời kì dân số vàng , tận dụng triệt để thành tựu của Cách mạng công nghệ lần thứ 4 , chuyên sâu một số ngành nền tảng , có lợi thế cạnh tranh , lợi thế cách mạng phát triển nhanh , chiến lược . Thúc đẩy phát triển công nghiệp điện tử , công nghệ thông tin là con đường chủ chốt , phát triển công nghiệp chế tạo thông minh , chế tạo là trung tâm là bước đột phá , phát triển công nghiệp chế biến , chú trọng và tập trung phát triển công nghiệp xanh.

Nhờ sự chỉ đạo tài tình của Đảng và nhà nước , thời gian tiếp theo thì ngành công thương sẽ tích cực tập trung thể hiện , cụ thể hoá những quan điểm , chủ trương đường lối của Đảng , hiện đại hoá đất nước , chính sách pháp luật của nhà nước công nghiệp hoá , tạo dựng môi trường minh bạch và pháp lý cho những doanh nghiệp thuộc mọi thành phần kinh tế phát triển , cạnh tranh , khơi thông các nguồn lực trong nước và thu hút nguồn đầu tư từ bên ngoài , quyết tâm đưa nước ta trở thành một nước công nghiệp hiện đại [3].

1.1.3. Ứng dụng

Ngày nay với sự phát triển của khoa học kỹ thuật, đặc biệt trong lĩnh vực tự động hóa, thì các nhà máy thường phải thường xuyên giám sát các thông số của các thiết bị máy móc như : áp suất , nhiệt độ , dòng... từ đó để điều khiển và giám sát các thiết bị , nội dung đề tài là một ứng dụng trong quy trình đó, việc giám sát các thông số của thiết bị điện trong công nghiệp giúp cho quá trình sản xuất được ổn định, thúc tiến được quá trình cho ra sản phẩm một cách đúng thời điểm, giảm thiểu được lượng sản phẩm bị lỗi,tăng năng suất sản xuất, khả năng cảnh báo chính xác khi có sự cố , tiện lợi dễ sử dụng... từ đó giúp cho nhà máy sẽ sản xuất được ổn định và phát triển hơn, việc giám sát các thông số của động cơ là quan trọng bậc nhất trong sản xuất.

Ứng dụng về việc theo dõi các thông số tốc độ , nhiệt độ , áp suất , độ ẩm động cơ công suất của các thiết bị điện trong công nghiệp, nhà máy , ... từ đó giúp chúng ta có thể giám sát và theo dõi tình trạng của máy móc có ổn định không, để từ đó có thể đưa ra các giải pháp tối ưu nhất để khắc phục các sự cố đáng tiếc xảy ra. Nhờ vào những thông số đó, ta có thể giám sát được tình trạng sản xuất có ổn định hay bất ổn định , điều chỉnh được các thông số, ... Tóm lại khoa học công nghệ hiện nay đang phát triển rất mạnh mẽ và quan trọng giúp chúng ta bước vào giai đoạn công nghiệp hoá hiện đại hoá đất nước khi mà chúng ta đã và đang đi sau họ rất nhiều.



Hình 1.4. Ứng dụng khoa học công nghệ vào đời sống [4].

1.2. Tìm hiểu về Smart Factory

1.2.1. Khái niệm

Nhà máy thông minh hay Connector Factory là sự phát triển đột phá từ 1 hệ thống sản xuất truyền thống theo hướng tự động hoá chuyển sang một hệ thống linh hoạt hơn có thể xử lý dữ liệu cũng như kết nối với hệ thống để phù hợp với nhiều đối tượng hay với thị trường .

Hệ thống nhà máy thông minh có thể giúp con người có thể giám sát hiệu quả quá trình bảo trì , sản xuất, số hoá mọi hoạt động , theo dõi kho ,... Nó cũng có thể tích hợp nhiều nguồn dữ liệu khác nhau từ các thiết bị sản xuất đến quá trình cung ứng để theo dõi và giám sát hệ thống linh hoạt hơn, đỡ tốn thời gian hơn có khả năng tự hiệu chỉnh cũng như dự báo [5].

1.2.2. Tính năng

Gồm các tính năng chính :

- Tối ưu hoá
- Kết nối
- Minh bạch
- Chủ động và nhanh nhẹn

Với những tính năng có những vai trò cho phép cho việc giúp các tổ chức nâng cao những quá trình sản xuất và cho phép các quyết định sáng suốt . Bên cạnh đó cũng có điều cần lưu ý là các nhà máy thông minh không thể nào giống nhau , các nhà ưu tiên các lĩnh vực cũng như tính năng phù hợp với yêu cầu .



Hình 1.5. Hình ảnh lắp đặt thực tế giải pháp SEEACT tại nhà máy sản xuất [6].

Tính năng được cho là quan trọng nhất của nhà máy thông minh là nó có thể tự động kết nối , nó cũng là nguồn giá trị quan trọng nhất . Yêu cầu của các nhà máy thông minh là về các quy trình cũng như các vật liệu cơ bản được kết nối với nhau để đưa ra những dữ liệu cần thiết nhằm đưa ra đưa ra các quyết định theo thời gian thực .

1.2.3. Đặc trưng của nhà máy thông minh

Sau khi tìm hiểu khái niệm về nhà máy thông minh là gì ở trang daco.vn , nhóm chúng em đã có cái nhìn khái quát về nhà máy thông minh . Để tìm hiểu đặc trưng của Smart Factory và hoạt động của mô hình này gồm :

Tự động hoá

Là sự kết hợp giữa công nghệ của cảm biến cùng các thiết bị máy móc thông minh , cùng với đó là khả năng tự động hoá trong sản xuất , ở đó chúng ta sẽ tham gia rất ít vào quá trình sản xuất hay có thể là không cần tham gia vào . Do đó mà nhà máy thông minh sẽ vận hành một cách tự động .

Kết nối

Là đặc trưng quan trọng của đặc trưng Smart Factory . So với những cách mạng công nghiệp có ở trước đó , Smart Factory qua tính năng kết nối đã để lại điểm nhấn . Nhằm mục đích phân tích xử lý dữ liệu quan trọng nhằm phục vụ việc giám sát sản xuất , quản lý , mô hình nhà máy để tạo ra mô hình kết nối tiên tiến giữa các tầng máy móc vận hành với các phần mềm [7].



Hình 1.6. Đặc trưng của Smart Factory [7].

Tính thông minh

Là một trong những đặc trưng quan trọng và không thể thiếu được khi nhắc đến Smart Factory đó là việc tận dụng tối ưu các thiết bị quét mã QR , các thiết bị thông minh như các loại cảm biến ,... Thông qua những thiết bị hiện đại đó dẫn đến thu nhập thông tin dữ liệu cho việc phục vụ dây chuyền sản xuất hàng hoá ở trong kho ,... hiệu quả , chính xác và nhanh chóng hơn .

Thời gian thực

Smart Factory là sự kết hợp từ nhiều phương tiện giữa con người và máy móc . Vậy nên việc giám sát hay vận hành đều được diễn ra trong thời gian thực . Từ đó đem lại khá nhiều lợi ích đáng kể cho doanh nghiệp . Cụ thể như doanh nghiệp có thể tối ưu hoá sử dụng nguồn lực 1 cách hiệu quả , hiệu suất ,...

Số hoá

Nó được tạo nên từ 2 hệ thống nền tảng ERP và MES . Nhà máy thông minh có khả năng số hoá tất cả các quy trình vận hành của doanh nghiệp . Tiếp theo đó là xây dựng sự liên kết , thống nhất giữa các ban , các phòng . Không những vậy nó còn giúp dữ liệu được nhất quán và đó là đặc trưng của Smart Factory .

Trực quang hoá

Ở những cơ sở sản xuất , những dữ liệu liên quan đến máy móc được truyền đạt lại bằng máy tính bảng công nghiệp . Smart Factory sử dụng các phương tiện đồ hoạ để truyền đạt thông tin và dữ liệu . Trực quan hoá giúp ích cho việc cập nhật tình trạng sản xuất , đưa ra các biện pháp xử lý khi gặp sự cố của doanh nghiệp .



Hình 1.7. Nhà máy thông minh [7].

Chủ động

Như chúng ta đã biết thì nhà máy thông minh là mô hình nhà máy hoạt động cơ bản dựa trên thiết bị , máy móc . Do đó , sự chủ động là một trong những khả năng ưu việt

của nhà máy thông minh . Khả năng lập kế hoạch và dự đoán được áp dụng triệt để trong nhà máy thông minh qua đó giúp việc quản lý chất lượng , quản lý vật tư được hiệu quả hơn

Toàn diện

Sau khi phân tích các thông tin , thu thập các dữ liệu thì hệ thống sẽ hoạt động và đưa ra toàn bộ hoạt động sản xuất kinh doanh bằng biểu đồ . Với quy trình làm việc như vậy , doanh nghiệp sẽ có cái nhìn khách quan cũng như tổng quát và toàn diện về hệ thống của mình , để từ đó đưa ra các quyết định .

Linh hoạt

Không những tự động mà nhà máy thông minh còn giúp doanh nghiệp có tính linh hoạt hơn . Lúc có biến động thị trường , nhà máy thông minh sẽ thích nghi một cách linh hoạt đảm bảo cân bằng yếu tố năng lượng sản xuất và thời gian thực .

Tối ưu hoá

Với sự liên kết chặt chẽ giữa các yếu tố con người , công nghệ đã tạo ra công trình nghiên cứu thông minh giúp doanh nghiệp đạt chất lượng cũng như đạt năng suất tốt hơn . Quan trọng , nhà máy thông minh giúp ích lớn cho việc tối ưu hoá chi phí và tiến độ giao hàng của DN .

Trong cuộc sống ngày nay , đặc biệt trong ngành sản xuất công nghiệp , đường đi đúng đắn là nhà máy thông minh , mang nhiều triển vọng . Để khái thác , áp dụng mô hình này một cách hiệu quả thì đầu tiên doanh nghiệp nên hiểu rõ về nó . [7].

1.3. Hướng chọn đề tài

Trong thời kì công nghiệp luôn phát triển không ngừng, nên khoa học kĩ thuật của nước ta cũng cần phải bắt kịp xu hướng chung của thế giới nếu không muốn bị tụt hậu. Các mạch điện, thiết bị điện tử luôn đổi mới, phát triển, công nghiệp tự động hóa ngày càng được phổ biến trong các nhà máy, các khu công nghiệp, việc đòi hỏi phải giám sát và điều khiển là một vấn đề quan trọng bậc nhất để những sản phẩm sản xuất ra đúng tiến độ và giảm thiểu được lỗi, nên em đã quyết định chọn với đề tài “Giám sát và điều khiển thiết bị điện trong công nghiệp sử dụng phương thức Modbus rtu rs485”.

1.4. Mục tiêu đề tài

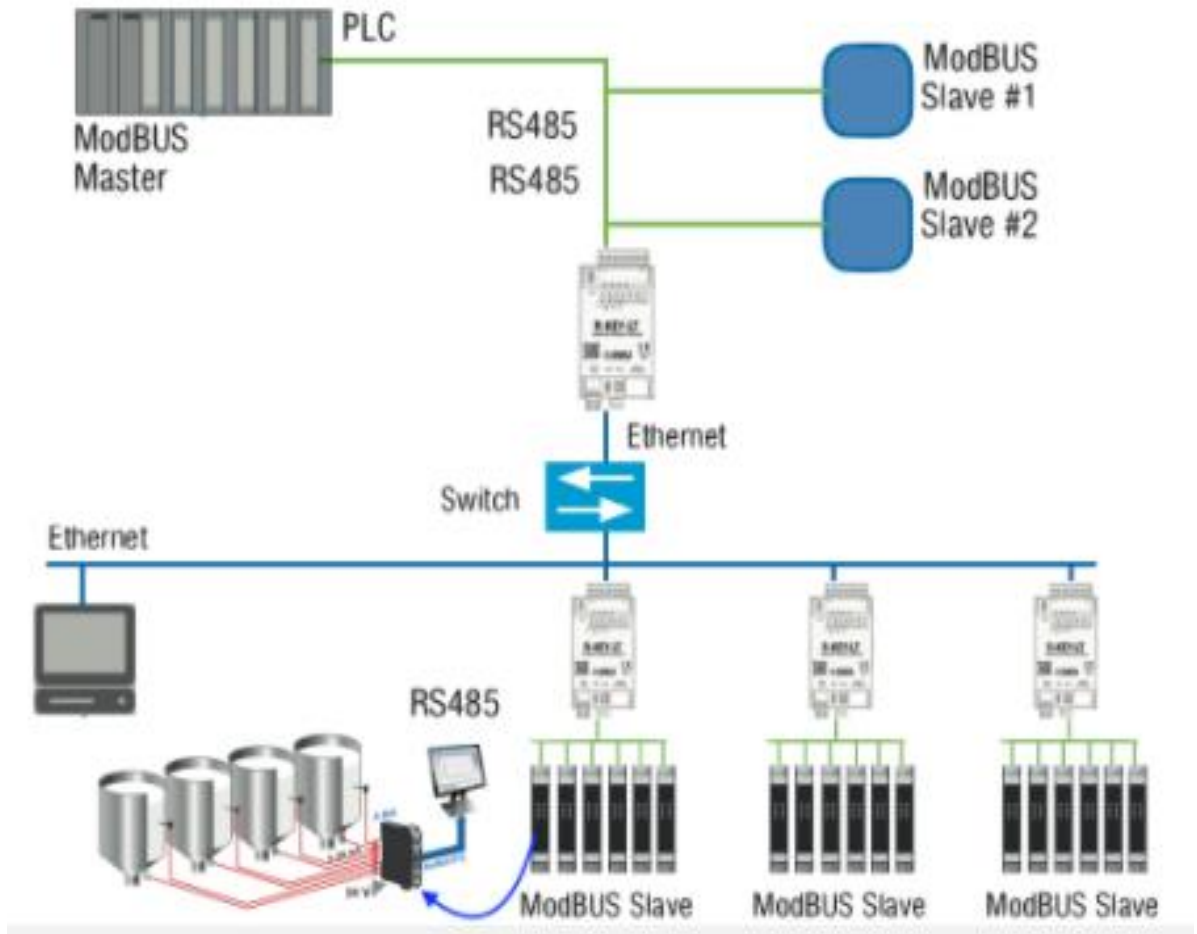
Ở đề tài : “ Giám sát và điều khiển thiết bị điện trong công nghiệp ”, đây là đề tài bước đầu tạo cơ sở nền tảng kiến thức cơ bản để từ đó tìm tòi phát triển ứng dụng khác trong đời sống hằng ngày. Sau khi hoàn thành đồ án nhóm cần đạt các mục tiêu:

- Biết cách lập trình phần mềm Arduino, biết được phương pháp để điều khiển các thiết bị điện trong công nghiệp .
- Thiết kế được giao diện để giám sát trên máy tính và điện thoại.
- Xây dựng thành công mô hình mô phỏng .
- Giúp chúng ta có được kỹ năng tìm kiếm tài liệu, nghiên cứu.

1.5. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: Phần mềm lập trình arduino, Web server, phương thức truyền Modbus rtu rs485, phương pháp điều khiển tốc độ , nhiệt độ , độ ẩm , điện áp các thiết bị trong công nghiệp.

Phạm vi nghiên cứu : Ở đề tài này thì ngoài những kiến thức đã học thì nhóm phải tìm hiểu thêm phương thức truyền MODBUS RTU và giao thức MQTT.



Hình1.8. Phương thức truyền Modbus [14].

1.6. Bộ cục đồ án

Đồ án “ Giám sát và điều khiển thiết bị điện trong công nghiệp” gồm 3 chương:

Chương 1 : Tổng quan :

Trình bày tổng quan về tình hình nghiên cứu, lý do chọn đề tài , mục tiêu và bố cục.

Chương 2 : Cơ sở lý thuyết :

Trình bày các khái niệm, cơ sở lý thuyết liên quan đến sản phẩm.

Chương 3: Thiết kế và thi công.

Ở chương này nhóm em thiết kế chi tiết từng khối , lưu đồ thuật toán và chọn linh kiện rồi thi công và hoàn thiện sản phẩm .

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

2.1. Khái quát về IoT trong công nghiệp

2.1.1. Khái niệm

Internet of Things hay còn gọi là IoT cũng còn có cách gọi khác là internet vạn vật được biết đến là một mạng lưới kết nối tất cả mọi thứ với nhau qua internet, những thiết bị kết nối có khả năng thu thập và xử lý cũng như truyền tải được những thông tin qua lại với nhau bằng 1 mạng duy nhất mà không cần đến con người.

Ở một ví dụ đơn giản là lái xe tự động. Trên xe sẽ được trang bị các cảm biến và camera để quét được hình ảnh trên đường đi, nhiệt độ, người tham gia giao thông khác,... [8].



Hình 2.1. Khái niệm về IoT [9].

2.1.2. IoT ở trong công nghiệp

Lợi ích của việc áp dụng công nghệ IoT vào các hoạt động sản xuất công nghiệp giúp cho chúng ta có thể nâng cao năng suất lao động, tiết kiệm được chi phí vận hành, từ đó mở ra nhiều cơ hội kinh doanh hơn cho doanh nghiệp.

Ứng dụng Internet vạn vật trong các ngành sản xuất công nghiệp rất quan trọng và cần thiết để giúp doanh nghiệp nhỏ và vừa có những lợi thế trong quá trình phát triển, với mục đích cải thiện tất cả mọi thứ trong quá trình vận hành. Những lợi thế cũng như tiện ích iot mang lại rất rõ ràng ở hiệu quả kinh doanh. [10].

2.2. Tìm hiểu về Web Server

2.2.1. Khái niệm về Web Server

Máy chủ cài đặt các chương trình phục vụ các ứng dụng web. Webserver có khả năng tiếp nhận request từ các trình duyệt web và gửi phản hồi đến client thông qua giao thức HTTP hoặc các giao thức khác. Có nhiều web server khác nhau như: Apache, Nginx, IIS, ...được gọi là Web Server [11].

Tương tự như các phần mềm được sử dụng khác, web server cũng chỉ là 1 ứng dụng phần mềm. Nó được cài đặt trên một máy mà bạn lựa chọn để giúp người dùng có thể tìm kiếm các thông tin mà website của bạn cung cấp. Nên cần phải có một máy tính cấu hình tốt, đáp ứng được khối lượng lớn người dùng truy cập cũng như lưu trữ được dung lượng dữ liệu cao.

2.2.2. Giao thức MQTT

Là 1 giao thức nhắn tin gọn nhẹ thiết kế để liên lạc giữa hệ thống máy tính và các thiết bị. Nó ban đầu được thiết kế cho các mạng Scada, băng thông thấp và các kịch bản sản xuất. Gần đây nó trở nên phổ biến một phần không nhỏ là do sự ngành càng thông dụng của Iot.

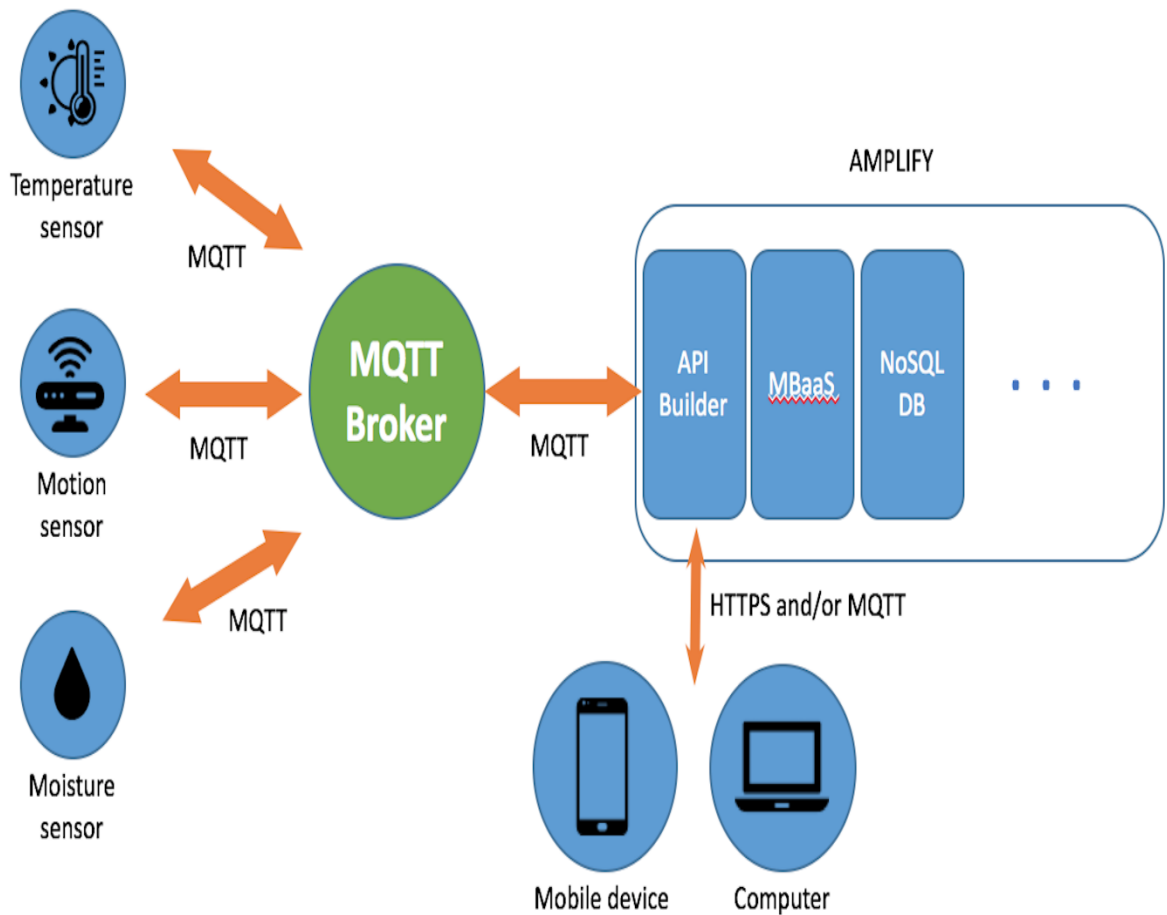
Trong đó, broker được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ clients.

Ưu điểm

Giao thức MQTT cho phép hệ thống SCADA cũng như truy cập dữ liệu iot. Những lợi ích mà MQTT mang lại chúng ta:

- Khả năng mở rộng tốt hơn
- Trao đổi thông tin tốt hơn
- Tốc độ truy cập giám đáng kể xuống giây
- Băng thông mạng tiêu thụ ít đi
- Thích hợp với việc do thám cũng như điều khiển
- Tận dụng tối đa băng thông ban đầu
- Tiết kiệm được thời gian
- An toàn

- Giao thức Publish / subscribe ít băng thông hơn băng thông cũ, bên cạnh đó nó thu thập được nhiều dữ liệu hơn [12].



Hình 2.2. MQTT trong IoT [13].

2.3. Phương thức Modbus RTU (RS485)

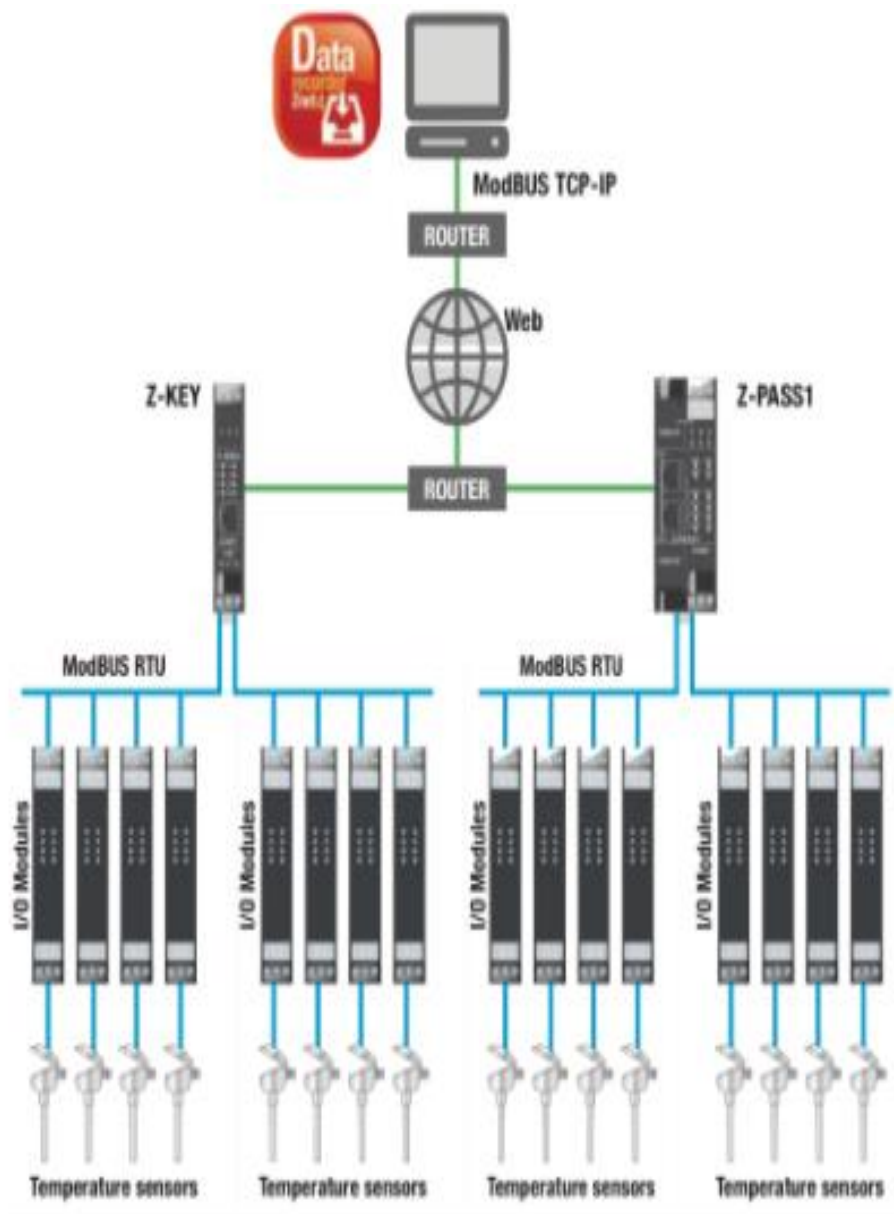
2.3.1. Modbus là gì?

Modbus là một chuẩn truyền thông công nghiệp được phát triển bởi Modicon (nay thuộc về Schneider) vào năm 1979 thay thế các chuẩn truyền thông truyền thống để truyền về PLC hoặc Scada. Cách hoạt động của Modbus dựa trên nguyên tắc Master – Slave trên đường truyền RS232 và RS485.

Modbus trở thành một chuẩn truyền thông công nghiệp tiêu chuẩn bởi nó : ổn định – đơn giản – dễ sử dụng & miễn phí. Điều này có nghĩa rằng các nhà sản xuất có thể tự tích hợp chuẩn Modbus vào sản phẩm của họ mà không cần trả tiền bản quyền.

Chỉ cần các thiết bị cùng chung một chuẩn với nhau thì có thể giao tiếp với nhau mà không cần quan tâm nó là thiết bị gì của hãng nào sản xuất. Đây chính là ưu điểm vượt trội so với các chuẩn truyền thông khác như : Profibus , Canopen , Manchester.

Modbus được sử dụng để kết nối với máy tính với các thiết bị đầu cuối (RTU) hay hệ thống SCADA). Một số loại Modbus : Modbus RTU , Modbus ASCII , Modbus TCP/IP [14].



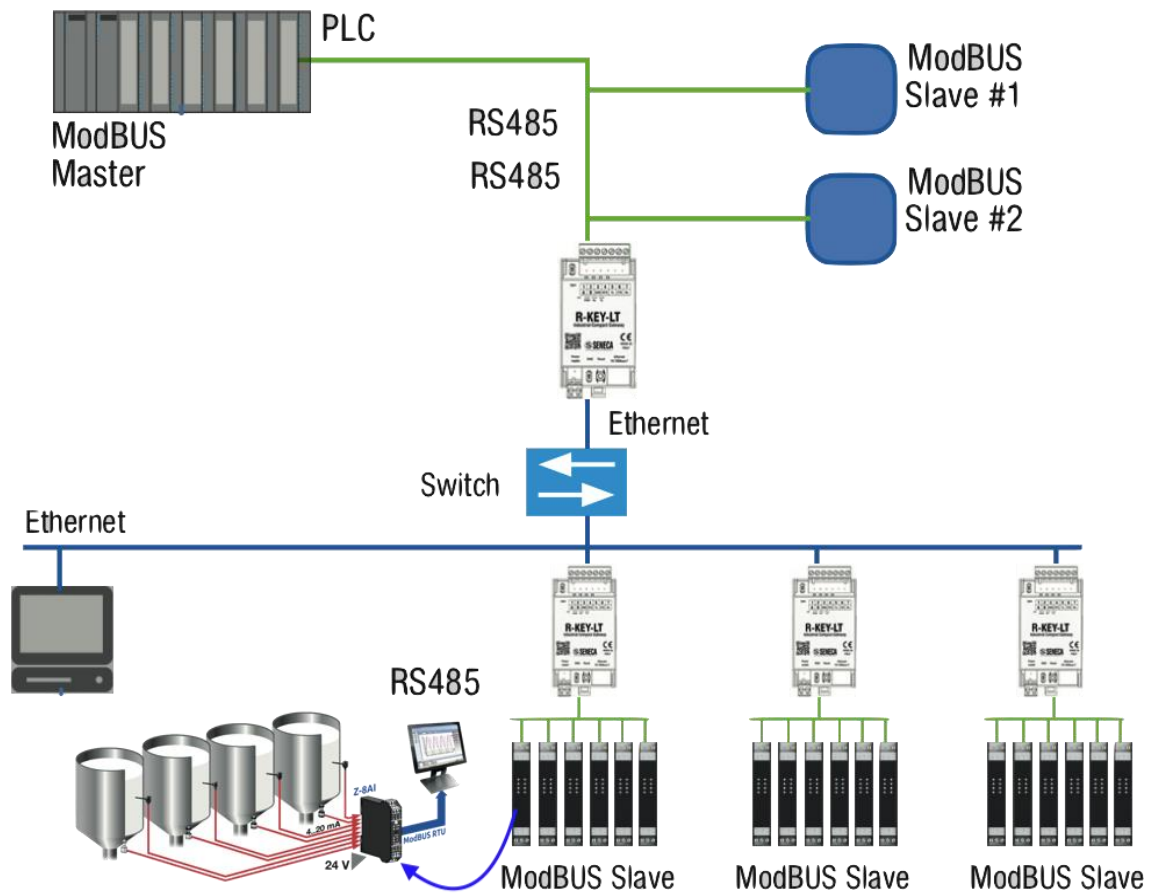
Hình 2.3. Modbus RTU đưa lên internet [14].

2.3.2. Ứng dụng của Modbus RTU

Sử dụng chuẩn truyền thông Modbus RTU giúp kết nối tất cả các thiết bị (max 128 địa chỉ) trên 2 dây tín hiệu RS485

Với mỗi modbus Slave tương ứng với một từ 4-8 tín hiệu Analog đầu vào hoặc nhiều hơn với 10 Digital. Các tín hiệu mắc song song nhau truyền vào Gateway để truyền lên internet thông qua Modul R-Key hoặc Z-key.

Như vậy tất cả các tín hiệu chỉ truyền đi trên 2 dây cho rất nhiều loại tín hiệu khác nhau thông qua các Modul Modbus RTU. PLC hay các thiết bị có chuẩn Modbus có thể giao tiếp với các thiết bị đo tại bất kỳ nơi nào thông qua Modbus TCP hay còn gọi là Modbus IP [14].



Hình 2.4. Modbus slave truyền về máy tính thông qua RS485 [14].

2.2.3. Ưu điểm và nhược điểm

Những ưu điểm mà Modbus RTU sở hữu :

- Tất cả các tín hiệu truyền trên 2 dây tín hiệu RS 485 với khoảng cách truyền xa 1200m.
- Giảm tối đa các dây vào PLC
- Giảm một số lượng khá lớn Modul mở rộng PLC
- Giảm không gian lắp đặt do bộ chuyển đổi có thiết kế mỏng nhỏ gọn so với Modul mở rộng của PLC
- Ít nhiễu và độ ổn định so với tín hiệu analog 4→20mA
- Các module hoạt động tách biệt nên dễ dàng quản lí
- Có thể dùng chung các hãng khác nhau có chuẩn Modbus RTU

Nhược điểm của Modbus RTU

- Tín hiệu không nhanh bằng việc dùng trực tiếp như analog hoặc Digital
- Chỉ phù hợp cho điều khiển có thời gian đáp ứng 1s trở xuống

- Cần PLC hay Scada có cấu hình mạnh đủ để đọc tất cả các thanh ghi khi dùng nhiều bộ chuyển đổi Modbus RTU [14].

2.4. Phần mềm Aduino

2.4.1. Aduino là gì?

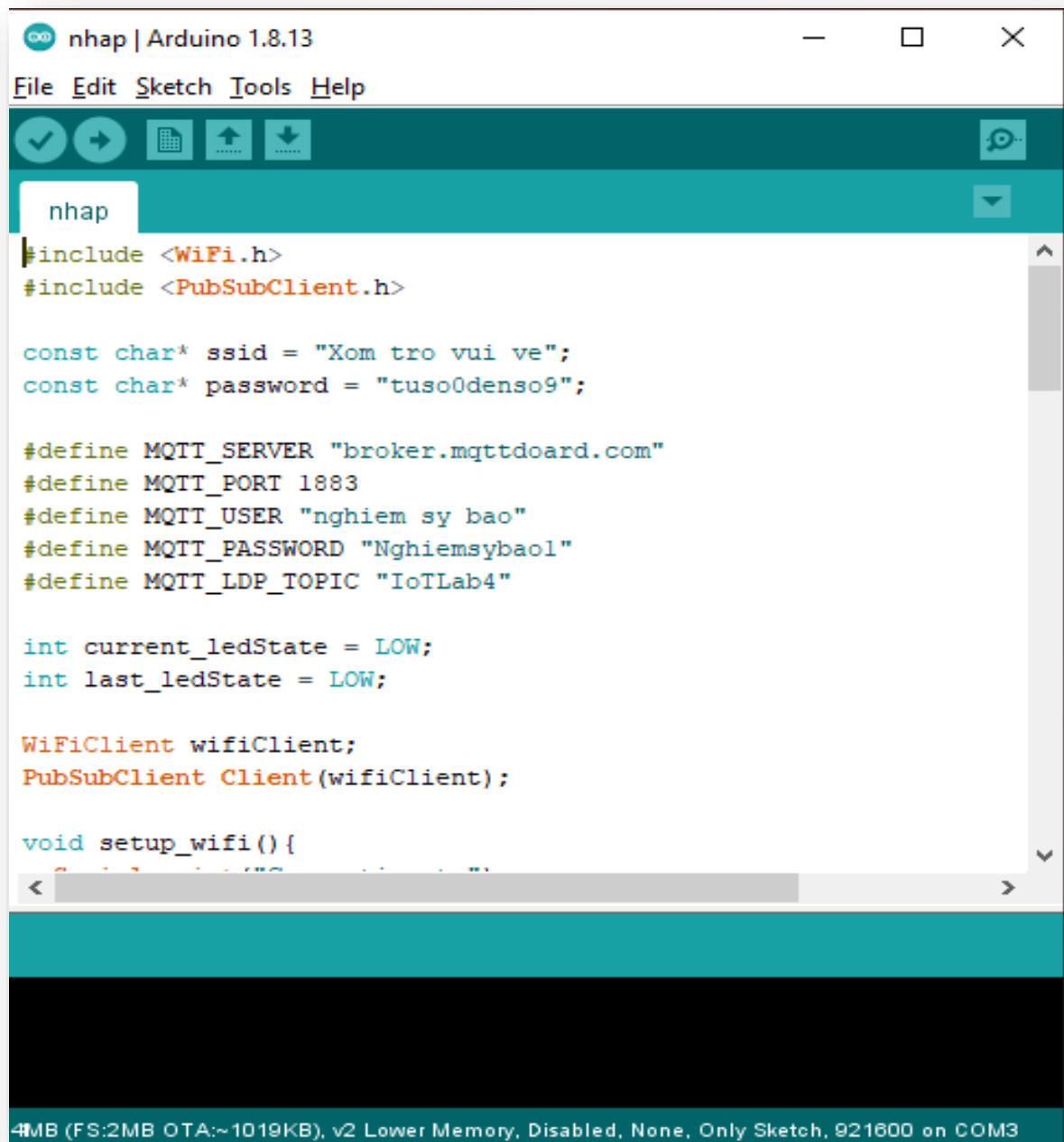
Arduino là một dự án được các sinh viên trường Interaction Design Institute Ivrea tại Interaction Institute Ivrea thực hiện. Là một trong những nền tảng mã nguồn mở được nhiều người trên thế giới đón nhận và tốc độ lan tỏa một cách nhanh chóng. Việc sử dụng nền tảng một cách đơn giản đã giúp cho những người yêu thích điện tử, sinh viên và giới chuyên nghiệp hoàn thành các đề tài, dự án một cách nhanh chóng và tiết kiệm chi phí [15].



Hình 2.5. Biểu tượng phần mềm Aduino [15].

Hướng dẫn sử dụng phần mềm Aduino cơ bản:

- Bước 1: Tạo 1 thư mục để lưu và quản lý các chương trình lập trình cho vi điều khiển.
- Bước 2 : Khởi tạo chương trình, giao diện sẽ hiện như hình dưới, nó sẽ lưu chương trình đã lập trình trước đó .
- Bước 3: Tiến hành chọn File → New để tạo 1 thư mục lập trình mới.
- Bước 4 : Tạo chương trình cho vi điều khiển bằng cách viết những dòng lệnh vào trong các hàm trong chương trình, sau khi viết chương trình xong, bấm tổ hợp phím “Ctrl + S” để tiến hành lưu file.

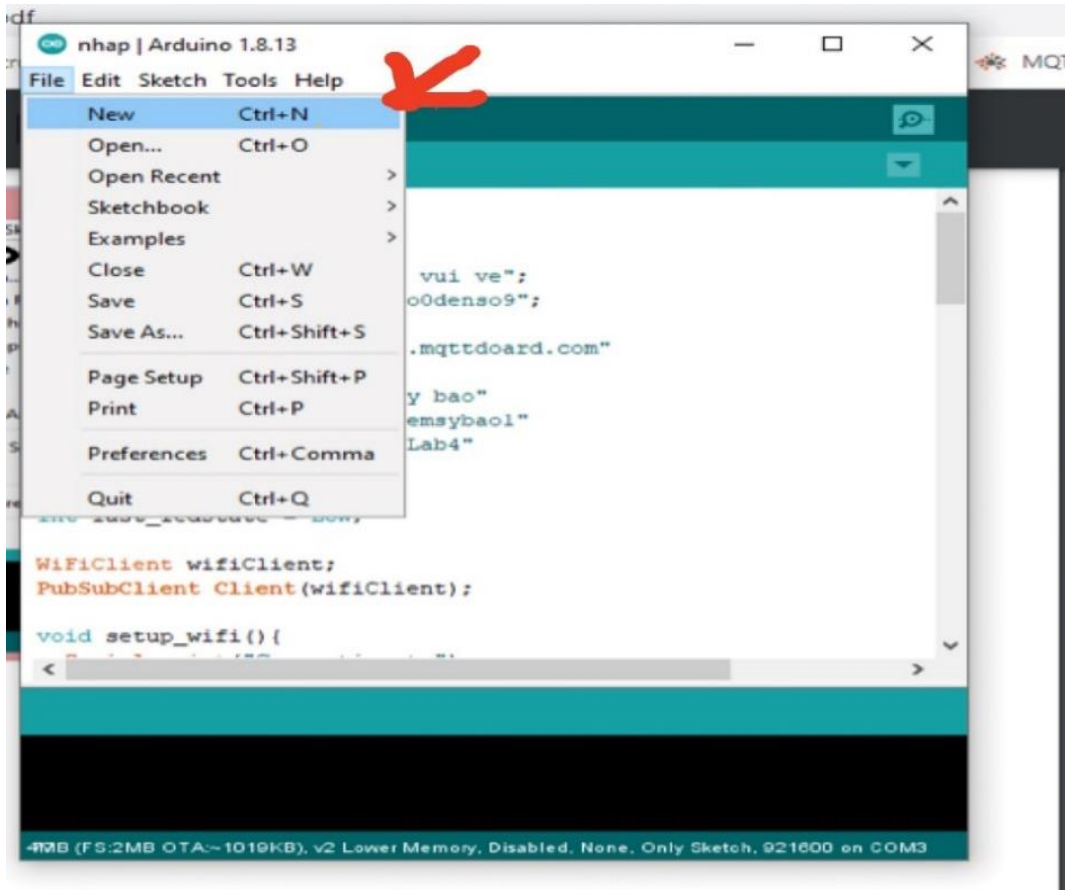


Hình 2.6. Giao diện khi khởi tạo

Ngôn ngữ này là ngôn ngữ lập trình C, nhưng có 1 chút khác là có những dòng lệnh có phân biệt chữ viết hoa và viết thường.

Ta biến hành biên dịch bằng cách nhấn vào biểu tượng  và nạp chương trình vào Arduino bằng cách nhấn biểu tượng 

Lưu ý: Chọn Board Arduino và port giao tiếp phù hợp



Hình 2.7. Tạo file mới

Nếu chương trình không có thư viện bạn có thể tìm kiếm thư viện trên internet để copy vào mục library của Arduino hoặc có thể search trực tiếp trên phần mềm và download ngay trên phần mềm, Arduino có hỗ trợ Add file Zip (file nén) bằng cách vào mục “Sketch” chọn “Include Library” chọn “Add .Zip Library..” để add file thư viện vào.

Nếu bạn muốn tìm thư viện ngay trên phần mềm thì ta chọn “Sketch” chọn “Include Library” chọn “Manager Library” sau đó tìm kiếm thư viện và download bình thường là phần mềm tự động Add thư viện vào cho bạn.

2.4.2. Phần mềm Visual Studio

Phần mềm Visual Studio là một trong những phần mềm hỗ trợ và giúp ích cho việc lập trình . Microsoft là nơi sáng tạo và tạo nên phần mềm này . Lúc trước , Visual Studio có tên mã là Project Boston . Nhưng sau này , Microsoft đã hợp tác và đóng gói thành 1 sản phẩm duy nhất .

Visual Studio là 1 hệ thống mà ở đó tổng hợp tất cả những gì liên quan tới phát triển ứng dụng , trong đó có trình thiết kế , trình chỉnh mã , gỡ lỗi . Có nghĩa là chúng ta có thể sử dụng , viết code tiện ích và dễ dàng hơn với phần mềm Visual Studio . Không những thế , chúng ta có thể thiết kế cũng như tạo giao diện theo ý của mình , trải nghiệm trong Visual Studio như lúc phát triển ứng dụng UWP , Xamarin , ...

Visual Studio gồm có 2 phiên bản là Visual Studio Professional và Visual Studio Enterprise, những phiên bản này có tính phí và thông dụng với những công ty chuyên về lập trình.

- Gồm nhiều tính năng :
- + Sử dụng nhiều ngôn ngữ lập trình
- + Nhiều nền tảng
- + Tính năng comment
- + Có hỗ trợ Git
- + Có hỗ trợ đầu cuối
- + Màn hình đa nhiệm
- + Lưu trữ an toàn
- + Phân cấp trong lưu trữ
- + Kho tiện ích phong phú
- + Dễ dàng viết Code [16].



Hình 2.8. Biểu tượng phần mềm Visual Studio [16].

2.5. Giới thiệu phần cứng

2.5.1. Giới thiệu về STM32F103C8T6.

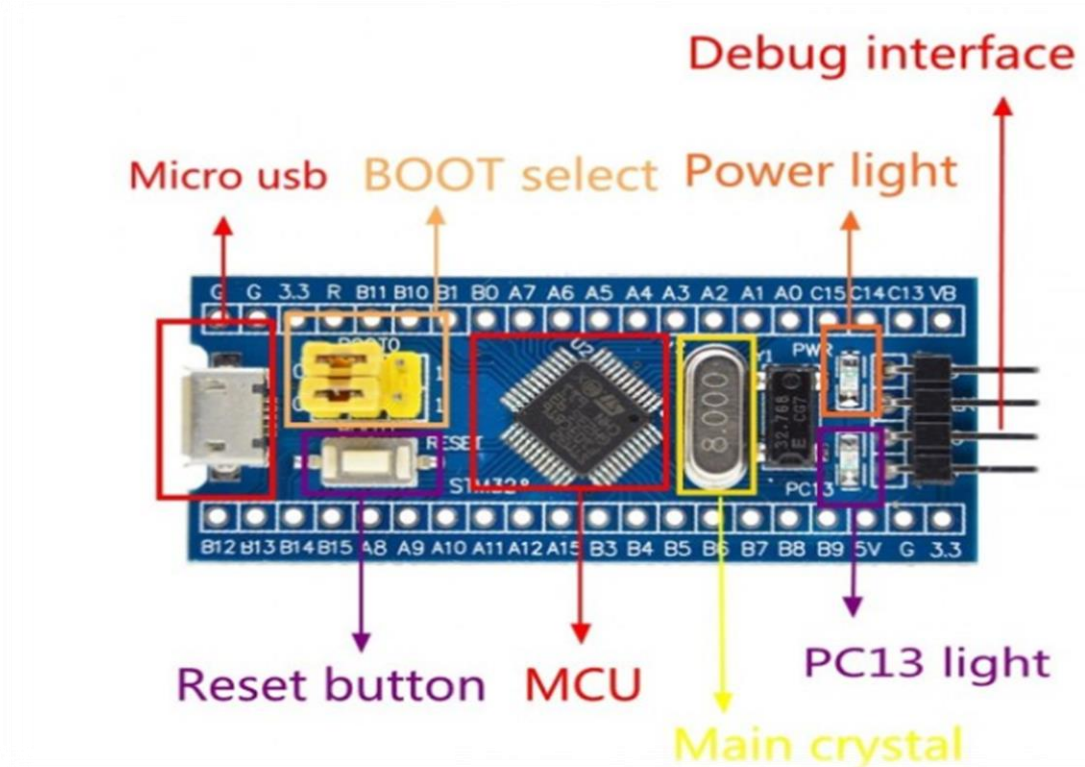
STM32 là chip khá phổ biến của dòng ST gồm có rất nhiều họ phổ biến là F0 , F1 , F2 , F3 , F4 Ở đó thì STM32f103 là họ F1 gồm lõi ARM COTEX M3 . Các thông số cơ bản của STM32 là nó có thể có tốc độ tối đa là 72 Mhz và nó là vi điều khiển 32 bit . Bên cạnh đó thì giá thành nó lại phù hợp và rẻ so với những vi điều khiển có chức năng tương tự . STM32 có mạch nạp dễ sử dụng .

Gồm những ứng dụng cơ bản : Điều khiển ứng dụng thông thường , thiết bị cầm tay và thuốc , dùng cho driver để điều khiển ứng dụng , máy tính và thiết bị ngoại vi chơi game , thiết bị lập trình PLC , GPS cơ bản , hệ thống cảnh báo , các ứng dụng trong công nghiệp , ...

Cấu hình chi tiết của STM32F103C8T6:

- ARM 32 bit Cortex M3 , Clock max : 72 Mhz
- Reset , quản lý nguồn và Clock .
 - Điện áp hoạt động từ 2 đến 3.6 V .
 - Power on reset(POR), Power down reset(PDR) và programmable voltage detector (PVD).
 - Thạch anh ngoài từ 4 đến 20 Mhz
 - Thạch anh nội dùng dao động RC ở mode 8Mhz hoặc 40khz.
 - Sử dụng thạch anh ngoài 32.768khz được sử dụng cho RTC.
- Về bộ nhớ :
 - 20KB SRAM
 - 64 KB bộ nhớ Flash .
- Trong trường hợp điện áp thấp:
 - Những module như ngủ , dừng đều có thể hoạt động ở chế độ chờ .
 - Cấp nguồn ở chân Vbat bằng pin để hoạt động bộ RTC và sử dụng lưu trữ data khi mất nguồn cấp chính.
- 2 bộ ADC 12 bit với 9 kênh cho mỗi bộ.
 - Giá trị chuyển đổi từ 0 đến 3.6 V .
 - Lấy mẫu 1 kênh hay nhiều kênh .
 - Hỗ trợ cảm biến nội .
- DMA: bộ chuyển đổi này giúp tăng tốc độ xử lý do không có sự can thiệp quá sâu của CPU.
 - 7 kênh DMA.
 - Hỗ trợ DMA cho UART , I2C , SPI , ADC .
- Gồm 7 time :
 - 3 timer 16 bit hỗ trợ các mode IC/OC/PWM.
 - 1 timer 16 bit hỗ trợ để điều khiển động cơ với các mode bảo vệ như ngắt input, dead-time..
 - 2 watdog timer dùng để bảo vệ và kiểm tra lỗi.
 - 1 sysTick timer 24 bit đếm xuống dùng cho các ứng dụng như hàm Delay....
- 9 kênh giao tiếp gồm có :
 - 2 bộ I2C(SMBus/PMBus).
 - 3 bộ USART.

- 2 SPIs .
- 1 bộ CAN interface .
- USB 2.0 full-speed interface
- Kiểm tra lỗi CRC và 96-bit ID [17].



Hình 2.9. Cấu tạo của STM32 [17].

2.5.2. Giới thiệu về ESP8266

ESP8266 là một hệ thống trên chip (SoC), do công ty Espressif của Trung Quốc sản xuất. Nó bao gồm bộ vi điều khiển Tensilica L106 32-bit (MCU) và bộ thu phát Wi-Fi. ESP8266 có 11 chân GPIO (Chân đầu vào / đầu ra đa dụng) và một đầu vào analog, có nghĩa là bạn có thể lập trình nó giống như với Arduino hoặc vi điều khiển khác. Bản thân chip ESP8266 có 17 chân GPIO, nhưng 6 trong số các chân này (6-11) được sử dụng để giao tiếp với chip nhớ flash trên bo mạch. Ngoài ra nó có kết nối Wi-Fi, vì vậy bạn có thể sử dụng nó để kết nối với mạng Wi-Fi, kết nối Internet, lưu trữ máy chủ web với các trang web thực, để điện thoại thông minh của bạn kết nối với nó, ... Khả năng là vô tận! Không có gì lạ khi con chip này đã trở thành thiết bị IoT phổ biến nhất hiện có.

ESP8266 có thể được dùng làm module Wifi bên ngoài, sử dụng firmware tập lệnh AT tiêu chuẩn bằng cách kết nối nó với bất kỳ bộ vi điều khiển nào sử dụng UART nối tiếp hoặc trực tiếp làm bộ vi điều khiển hỗ trợ Wifi, bằng cách lập trình một chương trình cơ sở mới sử dụng SDK được cung cấp.

Các chân GPIO cho phép IO Analog và Digital, cộng với PWM, SPI, I2C, v.v.

ESP8266 có nhiều ứng dụng khi nói đến IoT. Đây chỉ là một số chức năng mà chip này được sử dụng:

Kết nối mạng: Ăng-ten Wi-Fi của module cho phép các thiết bị nhúng kết nối với bộ định tuyến và truyền dữ liệu.

Xử lý dữ liệu: Bao gồm xử lý đầu vào cơ bản từ cảm biến analog và kỹ thuật số để tính toán phức tạp hơn nhiều với RTOS hoặc SDK không phải hệ điều hành

Kết nối P2P: Tạo giao tiếp trực tiếp giữa các ESP và các thiết bị khác bằng kết nối IoT P2P

Máy chủ Web: Truy cập các trang được viết bằng HTML hoặc ngôn ngữ phát triển[18].



Hình 2.10. Hình ảnh ESP8266 node MCU [19].

Chip xử lý	ESP8266EX
Mức điện áp hoạt động	3.3V
Mức điện áp làm việc	3.0V-3.6V
Các cổng giao tiếp	SDIO 2.0, (H) SPI, UART, 12C, 12S, IR Remote Control, PWM, GPIO
Giới hạn nhiệt độ cho phép	-40°C ~125°C
Wi-Fi 2.4GHz, hỗ trợ WPA/WPA2	

SPI flash	16 Mbs
Clock speed	27 Mhz đến 52 Mhz

Bảng 2.1: Các thông số của ESP8266 [19].

2.5.3. Relay

Công tắc điện tử được vận hành bởi dòng điện khá nhỏ mà có thể tùy ý bật hay tắt một dòng điện lớn hơn nhiều được gọi là Relay . Nam châm điện là trái tim của Relay (Cuộn dây trở thành một nam châm tạm thời khi mà có dòng điện chạy qua nó) . Chúng ta có thể coi Relay như 1 loại đòn bẩy : khi chúng ta bật lên bằng 1 dòng điện nhỏ và nó bật thiết bị khác sử dụng dòng điện lớn hơn nhiều .

Tại sao nó hữu ích ? Như tên cho thấy, nhiều cảm biến là những thiết bị điện tử cực kỳ nhạy cảm và chỉ tạo ra dòng điện nhỏ. Nhưng thường thì chúng ta cần chúng để điều khiển những bộ máy lớn hơn sử dụng dòng điện lớn hơn. Relay thu hẹp khoảng cách, tạo điều kiện cho dòng điện nhỏ kích hoạt dòng điện lớn hơn. Điều đó có nghĩa là relay có thể hoạt động như công tắc (bật và tắt) hoặc là bộ khuếch đại (chuyển đổi dòng điện nhỏ thành dòng lớn hơn) [20].



Hình 2.11. Một số Relay thông dụng [20].

Thông số kỹ thuật	Giá trị
Điện áp tối đa	AC 250V-10A / DC 30V-10A
Điện áp làm việc	5 VDC
Dòng kích Relay	5mA
Trạng thái kích	Mức thấp (0V)
Đường kính Lỗ ốc	3.1 mm
Kích thước	50 * 26 * 18.5 mm

Bảng 2.2. Các thông số của Relay [20].

2.5.4. Module Mp1584EN

Dùng để chuyển đổi điện áp DC / DC này được dựa trên IC MP1584, nó chuyển đổi điện áp đầu vào từ 4.5V đến 28V ra một điện áp nhỏ hơn từ 0.8V đến 18V, với dòng 3A.

Lưu ý: Đầu vào phải > 1.5V đầu ra, con chip này sẽ nhảy sang chế độ bảo vệ nhiệt độ cao khi nó làm việc quá tải mà không cần tản nhiệt [21].



Hình 2.12. Module Mp1584EN [21].

Thông số	Giá trị
Điện áp vào	4.5~ 28V
Điện áp ra	0.8~ 20V (có thể điều chỉnh)
Dòng ra	Tối đa 3A
Tần số chuyển đổi	1MHz (bình thường), 1.5Mhz (tối đa)
Độ gợn sóng đầu ra	<30mV

Hiệu suất chuyển đổi	90% (tối đa)
Nhiệt độ hoạt động	-40°C đến +85°C
Kích thước	22 x 17 x 4 mm

Bảng 2.3: Các thông số kỹ thuật của module Mp1584EN [21].

2.5.5. Cảm biến nhiệt độ, độ ẩm DHT11

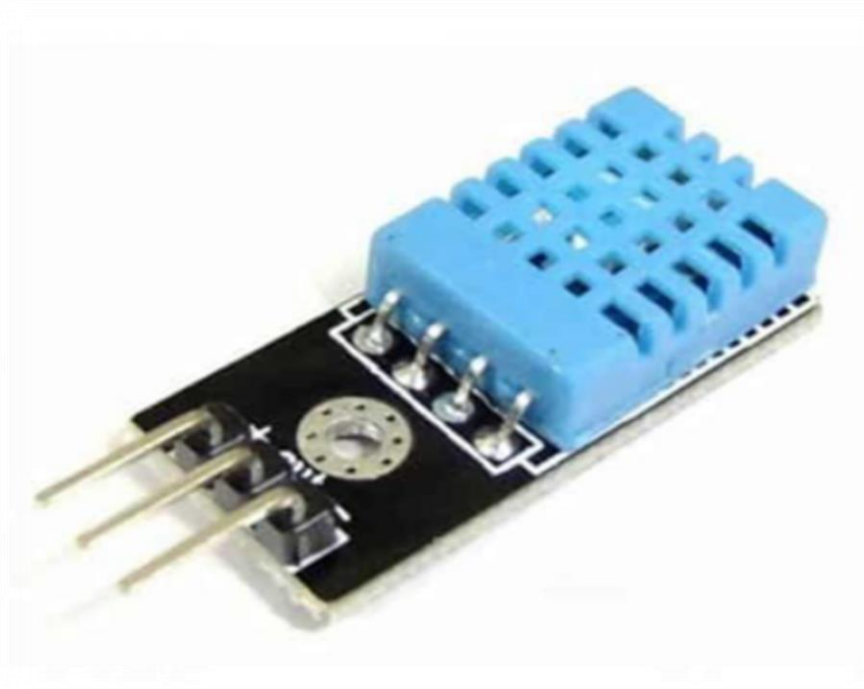
Cảm biến độ ẩm và nhiệt độ DHT11 là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua chuẩn giao tiếp 1 wire.

Chuẩn giao tiếp 1 wire là dùng 1 chân Digital để truyền dữ liệu.

Bộ tiền xử lý tín hiệu được tích hợp trong cảm biến giúp bạn có thể đọc dữ liệu chính xác mà không phải qua bất kỳ tính toán nào [22].

Thông số	Giá trị
Điện áp hoạt động	5VDC
Chuẩn giao tiếp	TTL, 1 wire
Khoảng giá trị đo độ ẩm	20% – 80%RH với sai số $\pm 5\%RH$
Khoảng giá trị đo nhiệt độ	0°C đến 50°C sai số $\pm 2^\circ C$
Tần số lấy mẫu tối đa	1Hz (1 giây 1 lần)
Kích thước	28mm × 12mm × 10mm

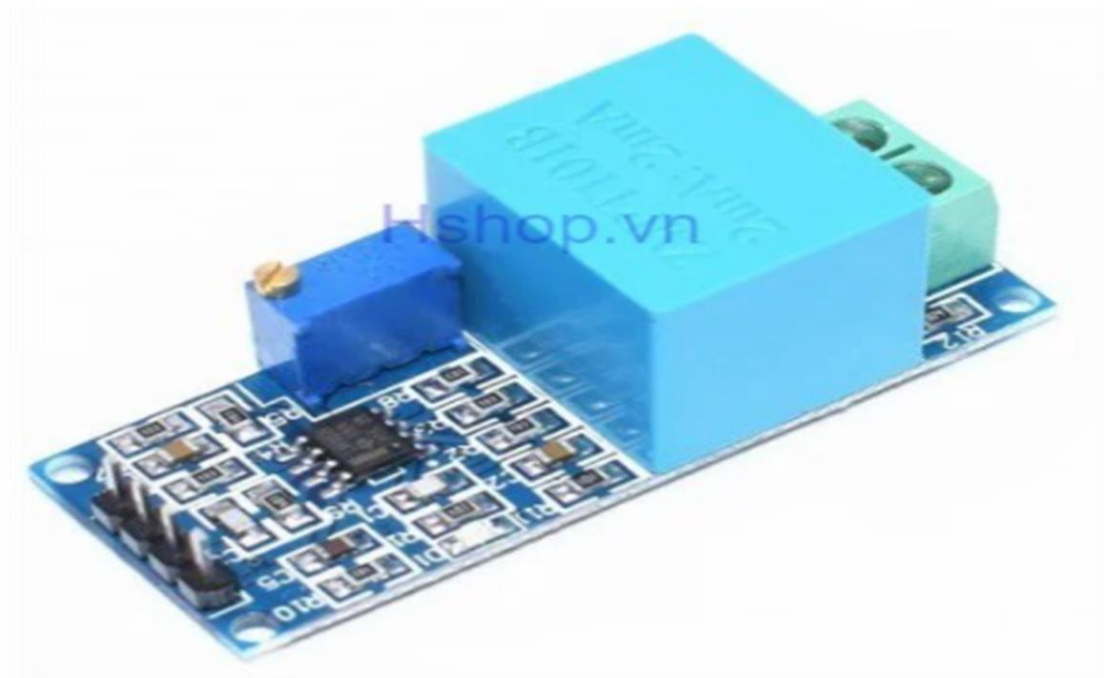
Bảng 2.4: Thông số kỹ thuật cảm biến độ ẩm , nhiệt độ DHT11 [23].



Hình 2.13. Cảm biến nhiệt độ ,độ ẩm DHT11 [23].

2.5.6. Cảm biến điện áp ZMPT101B

Cảm biến điện áp AC Voltage Sensor ZMPT101B được sử dụng để đo điện áp AC 1 cách chính xác bằng cuộn biến áp ZMPT101B (voltage Transformer), cảm biến có khả năng đo tối đa 250VAC với biến trở tinh chỉnh giá trị Analog đầu ra tích hợp, sử dụng cho các ứng dụng đo điện áp AC với Arduino hoặc Raspberry Pi,...[24].



Hình 2.14. Cảm biến điện áp ZMPT101B [24].

Thông số	Giá trị
Nguồn sử dụng	5~30VDC
Điện áp cảm biến tối đa	250VAC
Tín hiệu đầu ra	Analog 0~5VDC
Khoảng giá trị đo nhiệt độ	40°C đến 70°C
Tần số lấy mẫu tối đa	1Hz (1 giây 1 lần)
Kích thước	49.5mm × 19.4mm

Bảng 2.5: Các thông số kỹ thuật của cảm biến điện áp ZMPT101B [24].



Hình 2.15. Cấu tạo của cảm biến điện áp ZMPT101B [24].

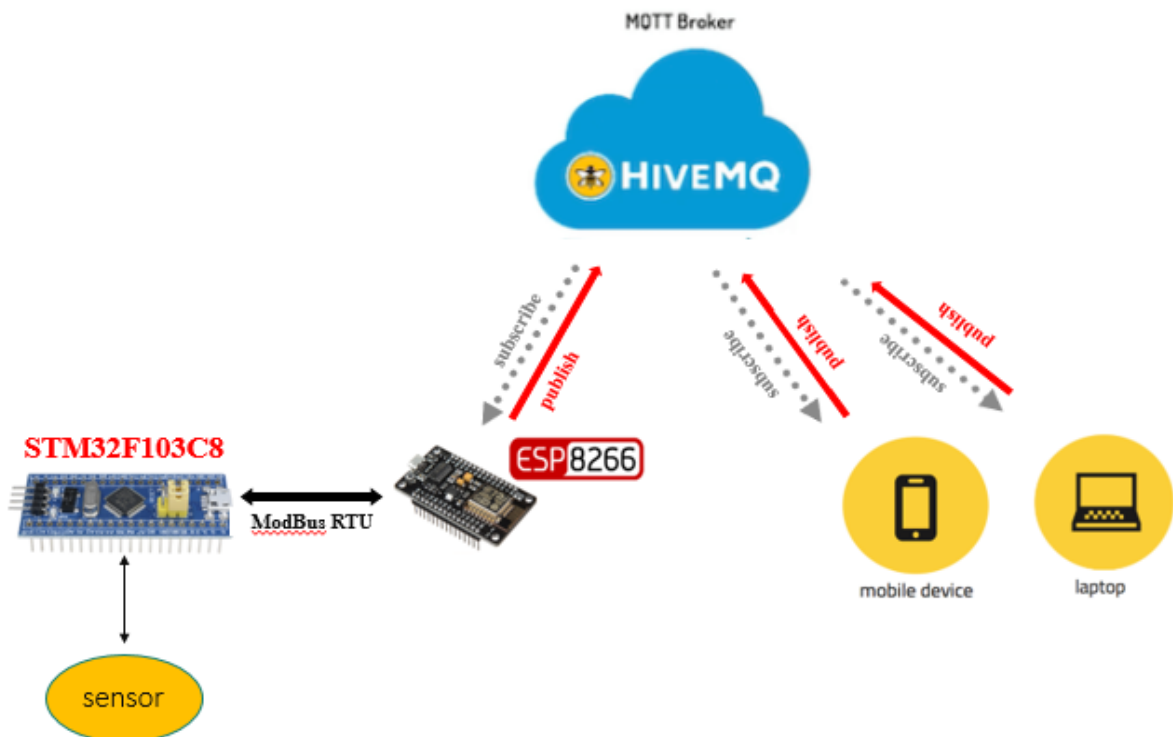
CHƯƠNG 3 : THIẾT KẾ VÀ THI CÔNG

3.1. Sơ đồ khối hệ thống

3.1.1. Yêu cầu đề tài

Yêu cầu : Hệ thống giám sát và điều khiển được thiết bị điện bằng phương thức truyền Modbus .Giám sát ,điều khiển được nhiệt độ , độ ẩm , điện áp và đo dòng được thiết bị điện trong công nghiệp.

3.1.2. Thiết kế sơ đồ khối đề tài



Hình 3.1. Sơ đồ khối hệ thống

Mô hình có mục đích sẽ thu thập những dữ liệu từ các thiết bị điện như: nhiệt độ, độ ẩm, điện áp, dòng điện. Khối Node là nơi đặt cảm biến để thu thập dữ liệu, trong các khối node được thiết kế tích hợp các module cảm biến và các chip vi điều khiển, module thu phát không dây sẽ nắm chức năng để truyền dữ liệu đã thu thập trước đó đến trạm trung tâm dữ liệu (Gateway), khi dữ liệu đến trung tâm sẽ được xử lý và phân tích nếu đúng với yêu cầu thì dữ liệu sẽ được gửi đến trung tâm dữ liệu và tiến hành hiển thị, đưa lên web. Tại đây người dùng và các nhà quản lý có thể quan sát, điều khiển được dữ liệu thông qua các thiết bị có kết nối mạng.

Hệ thống gồm các khối cơ bản :

- + Khối Node .
- + Khối Gateway.
- + Khối Web server .

3.1.3. Chức năng từng khối

- **Khối nguồn :** Khối này sẽ cung cấp nguồn cho hệ thống. Nguồn chính sử dụng cho mạch là adapter 12V – 2A, Nguồn này sẽ qua module giảm áp để cho về 5V để cung cấp cho các module mp1584 để cung cấp cho vi điều khiển và các cảm biến.
Nguồn 220V sẽ cung cấp cho rơ-le bán dẫn



Hình 3.2. Adapter 12V-2A [25].

- **Khối xử lý :** Khối này dùng để xử lý tín hiệu, truyền ,nhận tín hiệu và giao tiếp để điều khiển thiết bị.

Khối xử lý được chia làm 2 phần chính :

Khối node nhóm dùng vi điều khiển STM32f103c8t6 có nhiệm vụ thu thập cảm biến và cập nhập trạng thái thiết bị sau đó sẽ truyền sang khối Getway.

Từ khối Getway chúng ta sử dụng vi điều khiển chính là ESP8266, nó sẽ có trách nhiệm truyền và nhận giữ liệu từ node lên Web Server.

- Khối cảm biến : Khối này có chức năng thu thập các giá trị nhiệt độ , độ ẩm , áp suất , dòng điện sau đó gửi lên Web server.

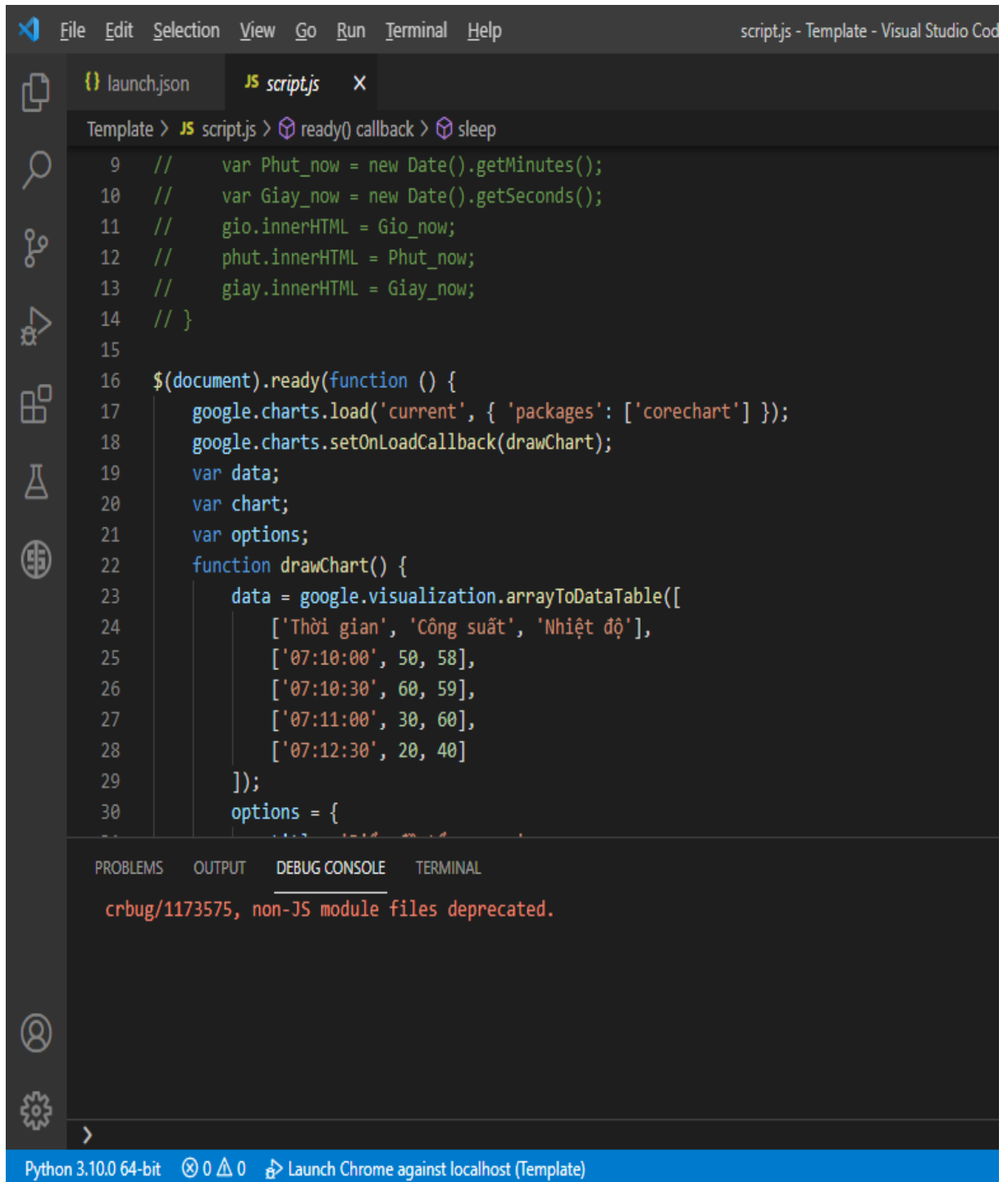
Khối sẽ thu thập các giá trị như nhiệt độ , độ ẩm , áp suất , dòng điện của thiết bị điện cần giám sát và điều khiển từ các cảm biến nhiệt độ , độ ẩm: DHT11, cảm biến dòng: ACS712 , cảm biến điện áp: ZMPT101B sau đó sẽ gửi lên Web Server.



Hình 3.3. Cảm biến dòng điện ACS712 [26].

- Khối Web Server : Khối này có chức năng thu thập , hiển thị và điều khiển dữ liệu. Khối Web Server chịu trách nhiệm làm trung gian truyền nhận giữ dữ liệu và điều khiển thiết bị thông qua Web Server.

Nhóm sử dụng phần mềm Visual Studio Code để tạo giao diện Web



Hình 3.4. Giao diện làm việc của phần mềm Visual Studio Code.

3.2. Thiết kế đề tài

3.2.1. Thiết kế các cảm biến

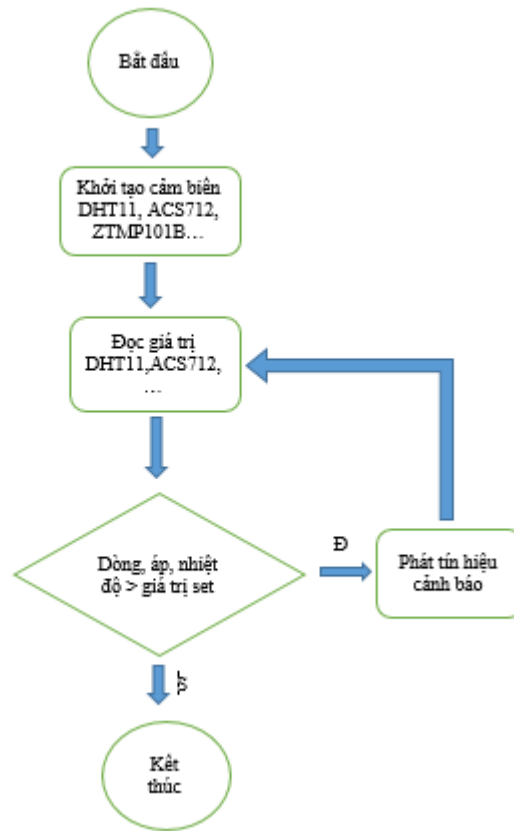
Các cảm biến có nhiệm vụ thu thập giá trị độ ẩm, nhiệt độ, áp suất, dòng điện sau đó gửi về Web server đồng thời phát ra cảnh báo khi dòng vượt quá ngưỡng quy định. Từ đó mà chúng ta chọn các linh kiện:

- + Cảm biến nhiệt độ , độ ẩm: DHT11
- + Cảm biến dòng: ACS712
- + Cảm biến điện áp: ZMPT101B
- + Vi điều khiển trung tâm: STM32

Linh kiện	Mức tiêu thụ
STM32	36mA
DHT11	2.5mA
ZMPT101B	1~ 2 mA
ESP8266	Max 320mA

Bảng 3.1: Mứ tiêu thụ điện năng của 1 số linh kiện .

Lưu đồ thuật toán:



Hình 3.5. Lưu đồ thuật toán khối cảm biến

Bắt đầu khi cấp nguồn thiết bị sẽ khởi tạo các giá trị cần thiết để cảm biến. Tiếp theo khối Node sẽ tính toán ra giá trị các cảm biến. Nếu phát hiện giá trị các cảm biến đo được dòng vượt quá mức giá trị mà ta set, thiết bị sẽ phát ra giá trị cảnh báo để người dùng có thể kịp thời khắc phục sự cố. Đồng thời thiết bị cũng gửi giá trị đến Gateway liên tục khi có thay đổi để người dùng giám sát từ xa.

3.2.2. Thiết kế khối xử lý

a. Khối xử lý trung tâm

Nhóm em chọn vi điều khiển STM32 với cấu hình mạnh mẽ cho dù với phân khúc thấp nhất STM32F0x cũng có thể hoạt động lên tới 48Mhz, 64kB Flash, 16kB RAM, 8 bộ Timer 16 bit, 1 bộ Timer 32 bit, 10 bộ ADC 12 bit, 8 bộ USART, 2 bộ SPI, 2 bộ I2C.

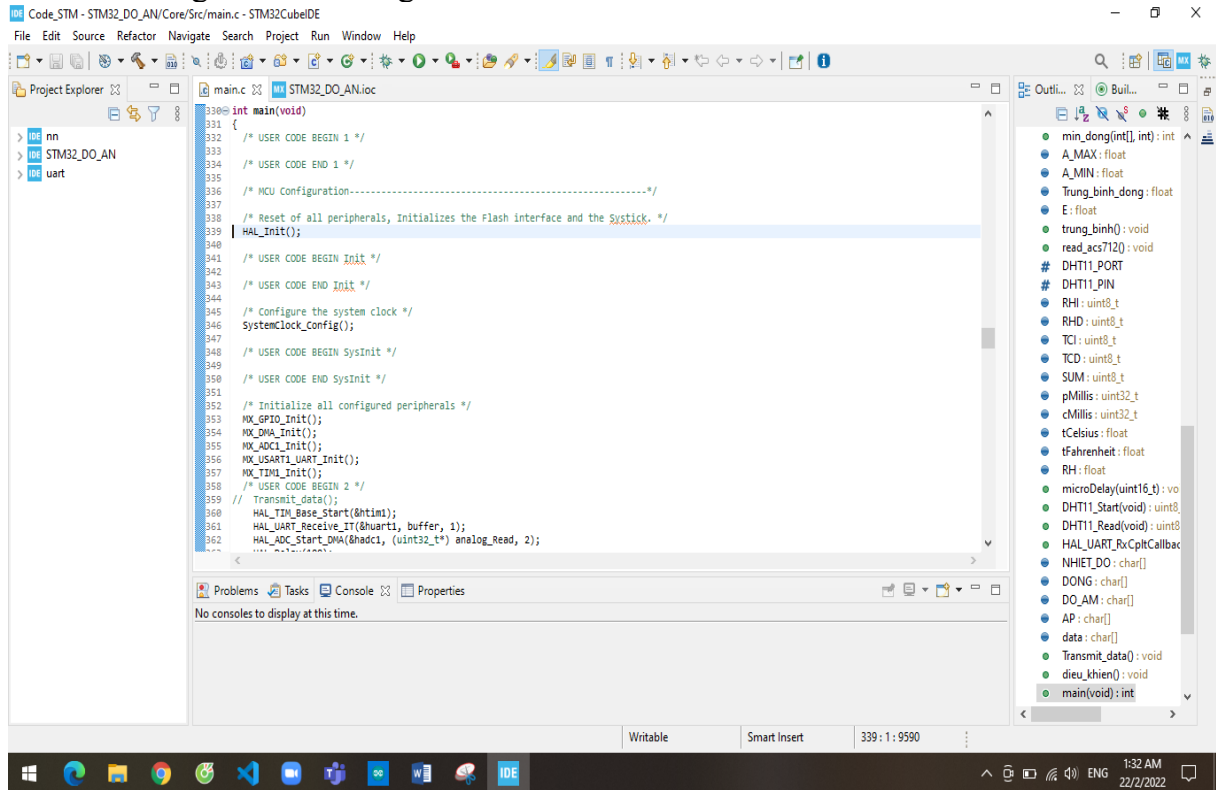
Phần mềm lập trình cho vi điều khiển là phần mềm STM32 cubeide.

Chip này hoạt động lên đến 48MHz, có 32kB Flash, 4kB RAM, 5 bộ Timer 16 bit, 10 bộ ADC 12 bit, 1 bộ USART, 1 bộ I2C, 1 bộ SPI. Đây là mức giá rẻ nhất so với các dòng vi điều khiển có cấu hình tương đương.

STM32F103C8T6 có chi phí thấp cho sinh viên và người yêu thích nó để khám phá thiết bị ARM Cortex-M3. Thiết bị này có sẵn và cực kỳ tiềm năng. Cuối cùng, thiết kế

của PCB nhỏ cho phép hàn chân vào các cạnh và cắm vào breadboard. Breadboard là board thuận tiện nhất để thực hiện các thí nghiệm.

Tuy vậy, vẫn còn có một số hạn chế. Ví dụ như , bộ điều khiển USB và CAN không thể hoạt động cùng một lúc. Các thiết bị ngoại vi khác có thể khác với các chân I / O được sử dụng. Hầu hết sự khác nhau giữa các chân được quản lý thông qua cấu hình AFIO (Đầu ra chức năng thay thế), cho phép các chân khác nhau được sử dụng cho chức năng của thiết bị ngoại vi.



Hình 3.6. Cửa sổ làm việc của phần mềm STM32 cubeide

Trong cấu hình ngoại vi, một số xung riêng biệt có thể được bật riêng để điều chỉnh mức sử dụng năng lượng. Khả năng tiên tiến của MCU làm cho nó phù hợp để nghiên cứu.

Thẻ nhớ flash được sử dụng ở 64K byte , nhưng có thể thấy rằng nó hỗ trợ 128K. Điều này cho phép các ứng dụng tốt được tận dụng vào thiết bị. Mạch được thiết kế gồm 4 gòai chính

- Khối nguồn :

Đề tài nhóm dùng module mp1584 hạ áp từ 12 xuống 5 v cung cấp cho vi điều khiển và các cổng vào ra.

- Khối vi điều khiển :

Sử dụng vi điều khiển chính là STM32F103C8T6.

Dùng để đọc các cách tín hiệu cảm biến và các thông tin từ khối GateWay điều khiển relay.

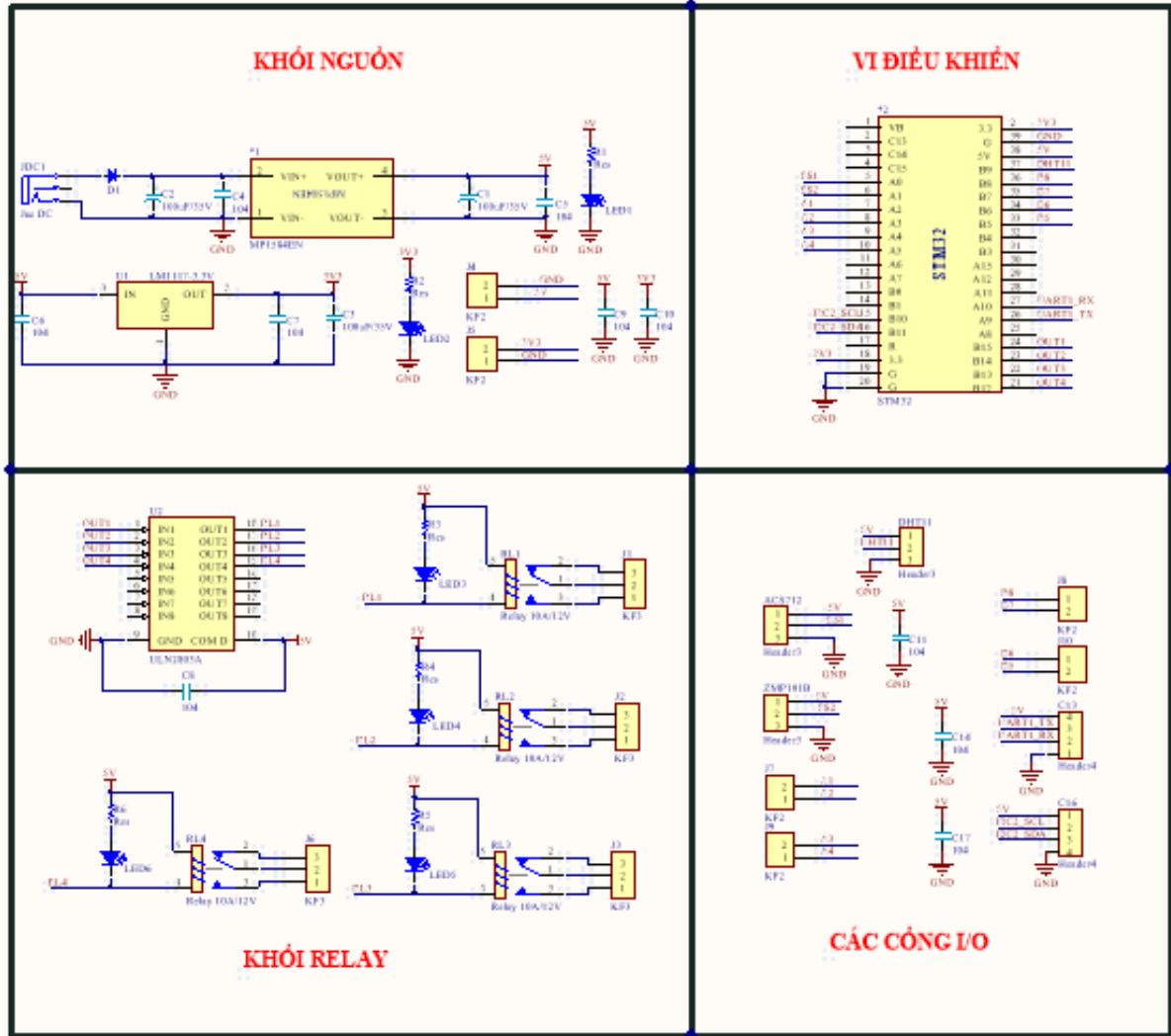
STM cũng có nhiều ưu điểm đó là tương thích với nhiều hệ điều hành, chương trình lập trình đơn giản rõ ràng, dễ sử dụng, sử dụng mã nguồn mở và có thể kết hợp với nhiều module khác nhau.

- Khối RELAY :

Dùng ULB2803 điều khiển 4 relay.

-Khối I/O:

Gồm cảm biến DHT11, ACS712, ZMPT101B các công I/O.



Hình 3.7. Sơ đồ nguyên lý STM32 đọc cảm biến và điều khiển

b. ESP8266

Để đưa dữ liệu lên cơ sở dữ liệu cần có một thiết bị kết nối Wifi và có thể giao tiếp với STM32F103C8T6 đó chính là ESP8266 . Có thể lập trình và nạp chương trình qua Internet nên nhóm chọn ESP8266.

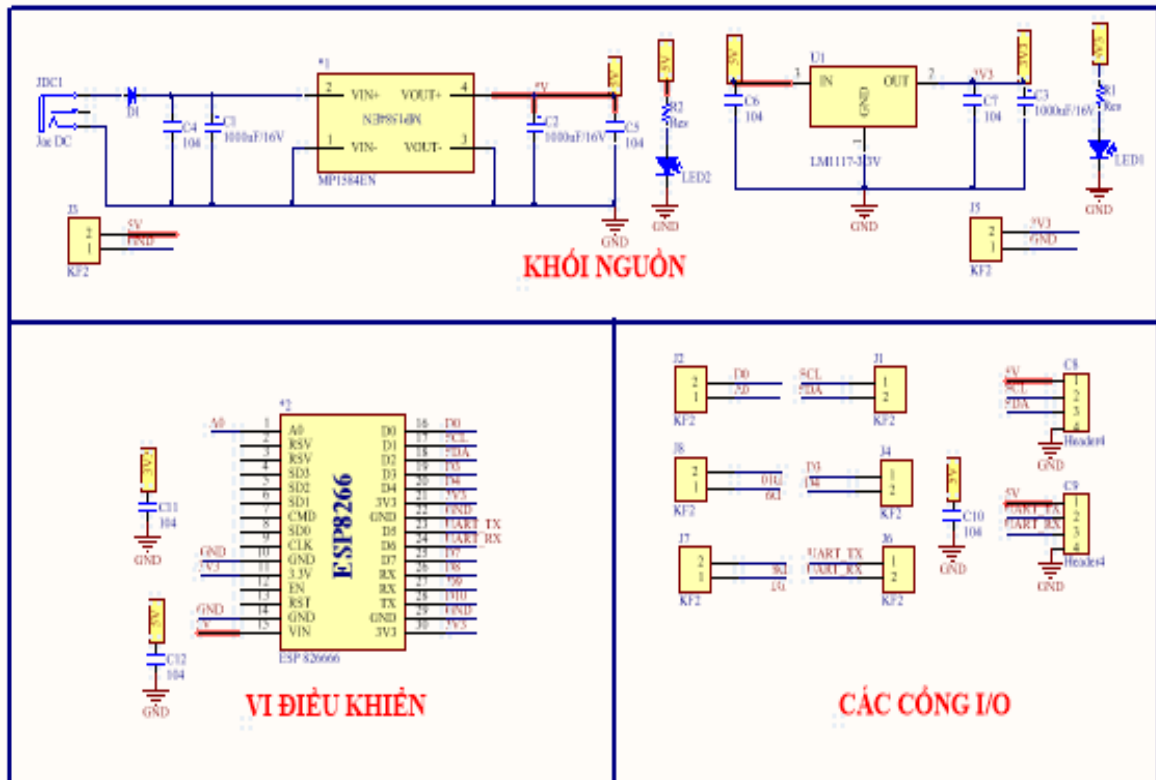
Khối vi điều khiển:

Khối vi điều khiển có chức năng kết nối và truy cập Wifi để lấy dữ liệu từ Web Server xuống cho khối xử lý trung tâm bằng giao thức UART , ngược lại lấy giữ liệu từ khối xử lý trung tâm lên Website để lưu trữ .

Chúng ta đưa dữ liệu cần 1 thiết bị kết nối được Wifi cũng như giao tiếp được STM32 đó chính là ESP8266 chịu trách nhiệm xử lý các thông tin gửi từ khối node mà gửi lên sever.

Khối i/o :

Gồm các nút nhấn và các chân I/O vi điều khiển.



Hình 3.8. Sơ đồ nguyên lý ESP8266 dùng Modbus RTU RS-485.

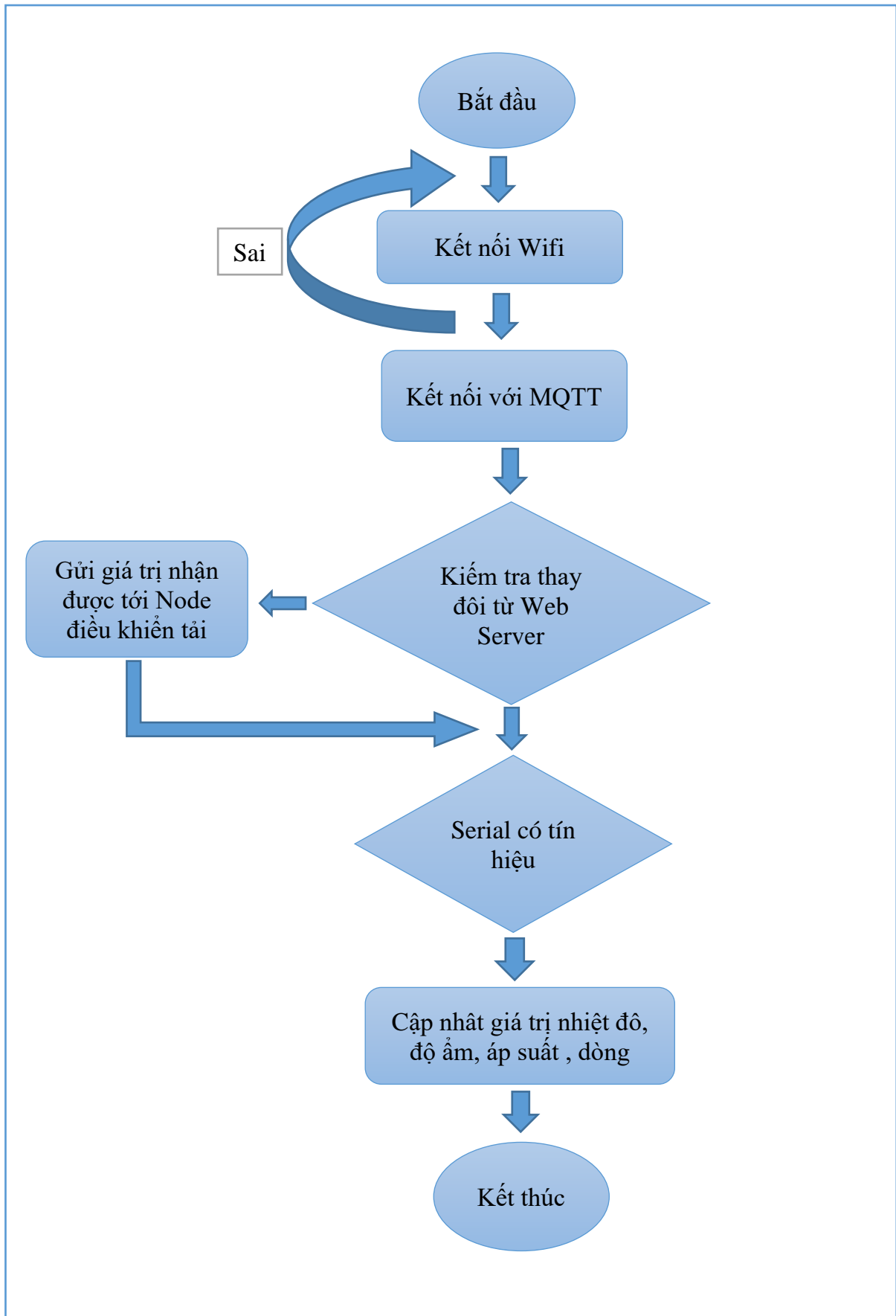
Khối vi điều khiển:

Khối vi điều khiển có chức năng kết nối và truy cập Wifi để lấy dữ liệu từ Web Server xuống cho khối xử lý trung tâm bằng giao thức UART , ngược lại lấy giữ liệu từ khối xử lý trung tâm lên Website để lưu trữ .

Chúng ta đưa dữ liệu cần 1 thiết bị kết nối được Wifi cũng như giao tiếp được STM32 đó chính là ESP8266 chịu trách nhiệm xử lý các thông tin gửi từ khối node mà gửi lên sever.

Khối i/o :

Gồm các nút nhấn và các chân I/O vi điều khiển.



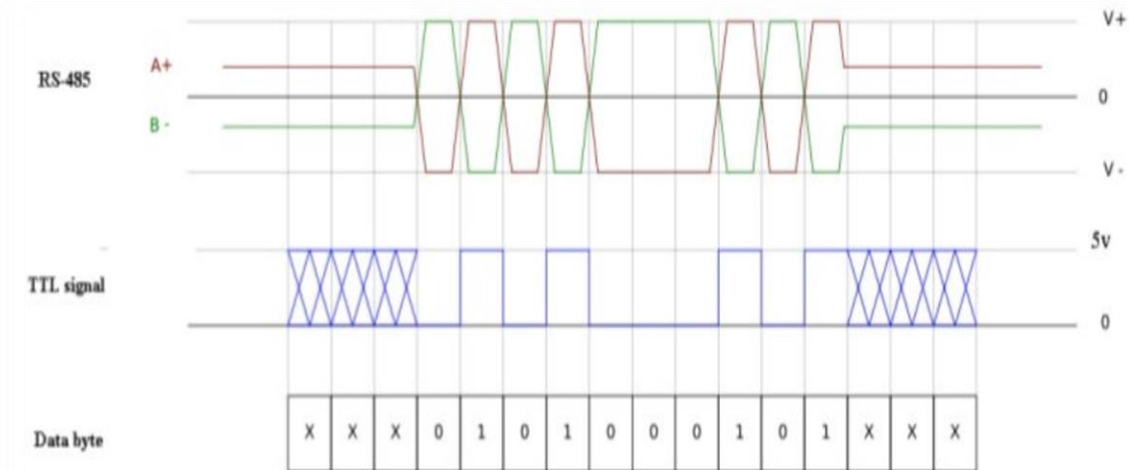
Hình 3.9. Lưu đồ thuật toán Gateway.

Khi cấp nguồn Getway sẽ khởi tạo các thông số cần thiết để giao tiếp với Module và kết nối Wifi. Sau đó sẽ cập nhật các giá trị thay đổi trên MQTT để gửi đến Node điều khiển tải bật tắt thiết bị. Sau đó Getway sẽ chờ để nhận các gói tin từ các node, nếu gói tin nhận được có ký tự cuối là dấu “#” Getway sẽ hiểu đây là gói tin của Node cảm biến gửi về sau đó tiến hành phân tích chuỗi để nhận được các giá trị nhiệt độ, độ ẩm, áp suất, dòng điện gửi về MQTT để hiển thị giá trị thay đổi trên app. Ngược lại nếu gói tin không có ký tự “#” ở cuối thì nó sẽ hiểu là gói tin từ node điều khiển tải gửi về và tiến hành phân tích chuỗi để cập nhật giá trị lên MQTT để đồng bộ giá trị nút nhấn trên app.

3.2.3. Khởi giao tiếp RS-485

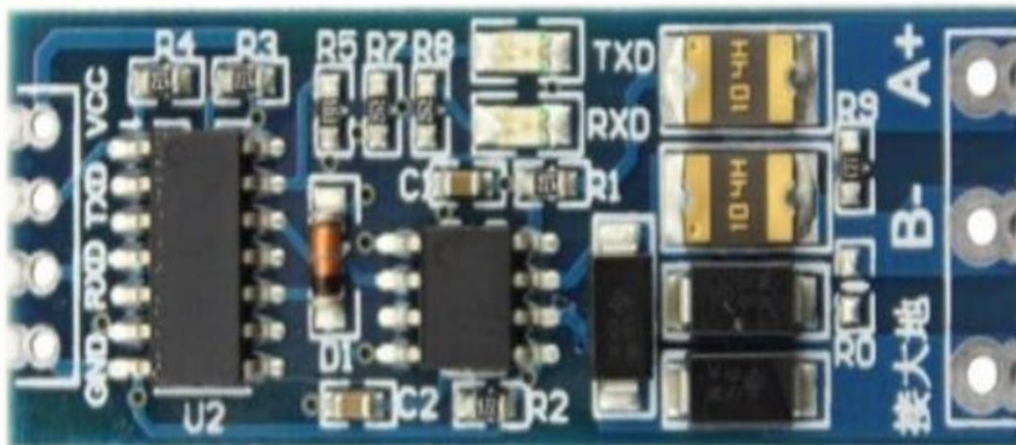
Khởi có chức năng là chuyển đổi giao tiếp RS-485 thành giao tiếp UART và tự động chọn chiều để khởi xử lý trung tâm truyền nhận dữ liệu.

Do sự khác nhau về tín hiệu RS-485 và tín hiệu UART không có cùng một mức điện áp được thể hiện như hình :



Hình 3.10. Mức điện áp của RS485 và UART [27].

Do tín hiệu của RS-485 là sự chênh lệch điện áp giữa V+ và V- và tín hiệu UART là chênh lệch điện áp 5V và 0V nên có thể truyền dữ liệu từ STM32 qua ESP8266 và



Hình 3.11. Sơ đồ mạch UART RS-485 [27].

ngược lại phải khối chuyên đổi cổng giao tiếp RS-485 của STM thành UART được hỗ trợ bởi arduino.

3.2.4. Khối trung tâm dữ liệu(Web server)

Với nhiều người khái niệm Web Server thực tế vẫn còn khá xa lạ , chính vì thế khi nghe đến nhiều người vẫn luôn thắc mắc Web Server là gì? Đây là tên gọi dành cho một loại máy chủ có dung lượng lớn, tốc độ cao có công dụng lưu trữ các thông tin trên internet như một ngân hàng dữ liệu bao gồm các website đã được thiết kế và các thông tin, tài khoản... có liên quan.

Ở mức độ cơ bản nhất , toàn bộ các website cần một chương trình máy tính, phân phối các trang web khi có yêu cầu từ người dùng. Chiếc máy tính chạy chương trình này là web server. Khi một người dùng sử dụng máy tính để truy cập một website, họ nhập và gửi yêu cầu tới internet về việc xem một trang web [27].

Điều này có thể thực hiện được là vì mỗi máy tính/ thiết bị kết nối internet đều được định danh với một địa chỉ nhận dạng duy nhất IP (viết tắt của từ Internet Protocol – giao thức internet). Thông qua địa chỉ này, các máy tính có thể tìm kiếm nhau.

Mỗi trang web tương ứng có một địa chỉ duy nhất là URL (viết tắt của từ Uniform Resource Locator). Khi người dùng sử dụng máy tính (máy khách) nhập URL của website vào trình duyệt web. Ví dụ: nhập: <https://longvan.net/> vào trình duyệt. Máy khách sẽ gửi yêu cầu truy cập đến IP của Web server sẽ nhận được yêu cầu về việc xem nội dung trang web của Long Vân thông qua giao thức http – giao thức được thiết kế để gửi các file đến trình duyệt web và các giao thức khác. Web server sẽ gửi nội dung web (bài viết, hình ảnh, video,...) đến máy khách thông qua đường truyền internet.

Ở mỗi web server đều có một địa chỉ IP hoặc cũng có thể có một domain name. Bất kỳ máy tính nào cũng có thể trở thành web server bởi việc cài đặt lên nó một chương trình phần mềm server software và sau đó kết nối vào Internet.

Tương tự như các phần mềm được sử dụng trên máy tính khác, Web Server software cũng chỉ là 1 ứng dụng phần mềm. Chúng được cài đặt trên một máy mà bạn lựa chọn để giúp người dùng có thể tìm kiếm các thông tin mà website của bạn cung cấp. Vì thế mà cần phải có một máy tính cấu hình tốt, đáp ứng được khối lượng lớn người dùng truy cập cũng như lưu trữ được dung lượng dữ liệu cao [27].

Về mặt phần cứng

Web server là một máy tính lưu trữ các file thành phần tạo nên một website (ví dụ: HTML, images, CSS, và file javascript...) và truyền chúng tới người dùng cuối.

Web server được kết nối đến internet và truy cập thông qua một domain giống như mozilla.org.

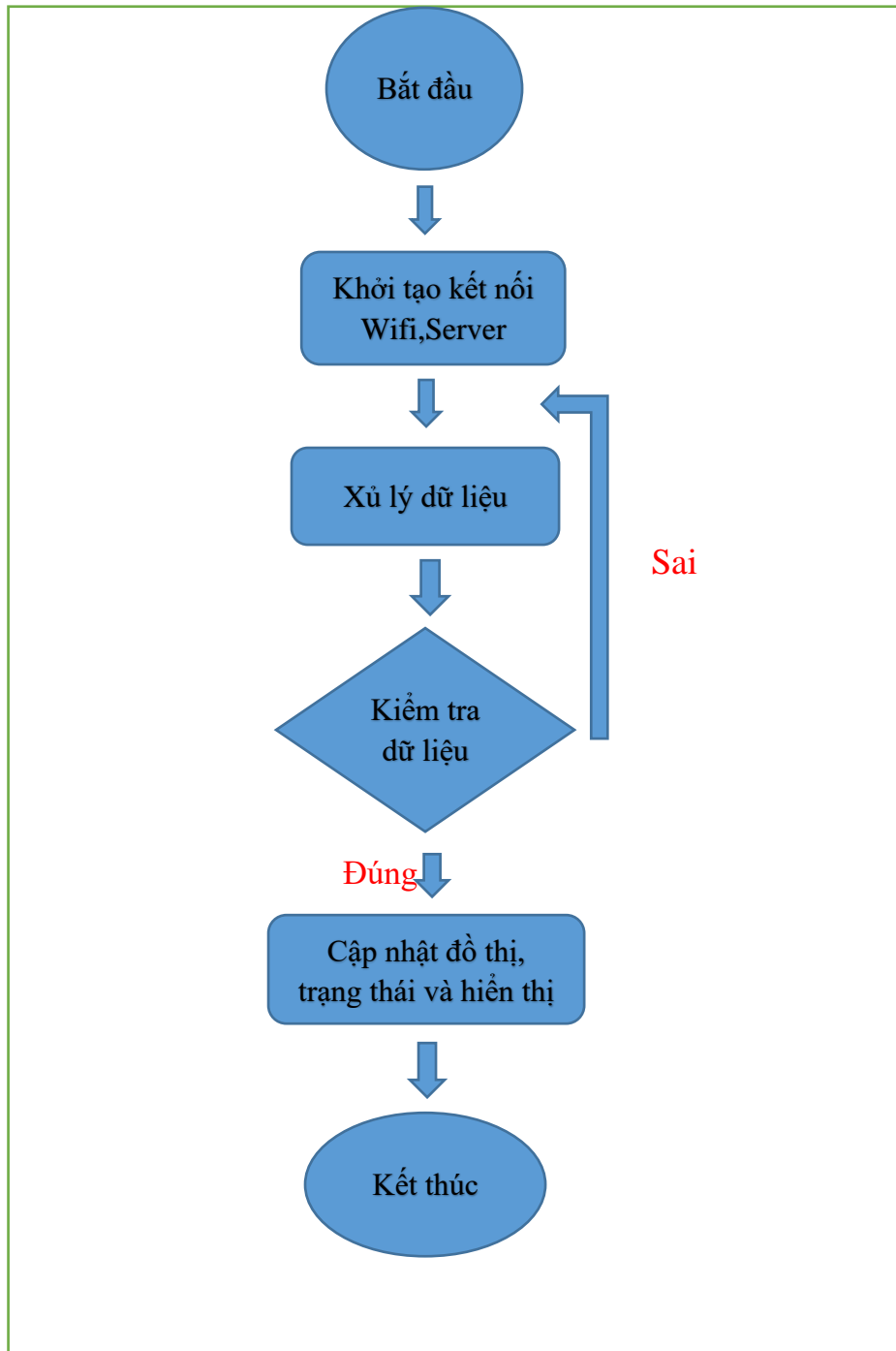
Về mặt phần mềm

Web server bao gồm một số phần kiểm soát người dùng web truy cập đến file host tại tối thiểu một HTTP server. Một HTTP server là một phần của phần mềm nó hiểu là URLs (web address) và HTTP (là phương thức để trình duyệt của bạn hiển thị trang web) Ở mức cơ bản nhất, bất cứ một trình duyệt nào cần một file host trên một web

server, trình duyệt đó sẽ request file đó thông qua HTTP. Khi một yêu cầu được gửi đến địa chỉ web server đúng thì HTTP server gửi trở lại một yêu cầu thông qua HTTP [28].

Lưu đồ thuật toán Web Server : Trước tiên nó sẽ khai báo thư viện cho module Wifi , sau đó nó sẽ tiếp tục kiểm tra kết nối Wifi . Truy cập vào Web Server nhằm lấy dữ liệu và truyền cho STM32 bằng cách gửi qua cổng giao tiếp Serial một chuỗi kí tự có cả kí tự đặc biệt. Tiếp đó dữ liệu sẽ được kiểm tra và xử lý tiếp theo rồi cập nhật thông số và hiển thị . Quá trình lặp lại tuần hoàn và liên tục .

Lưu đồ thuật toán :

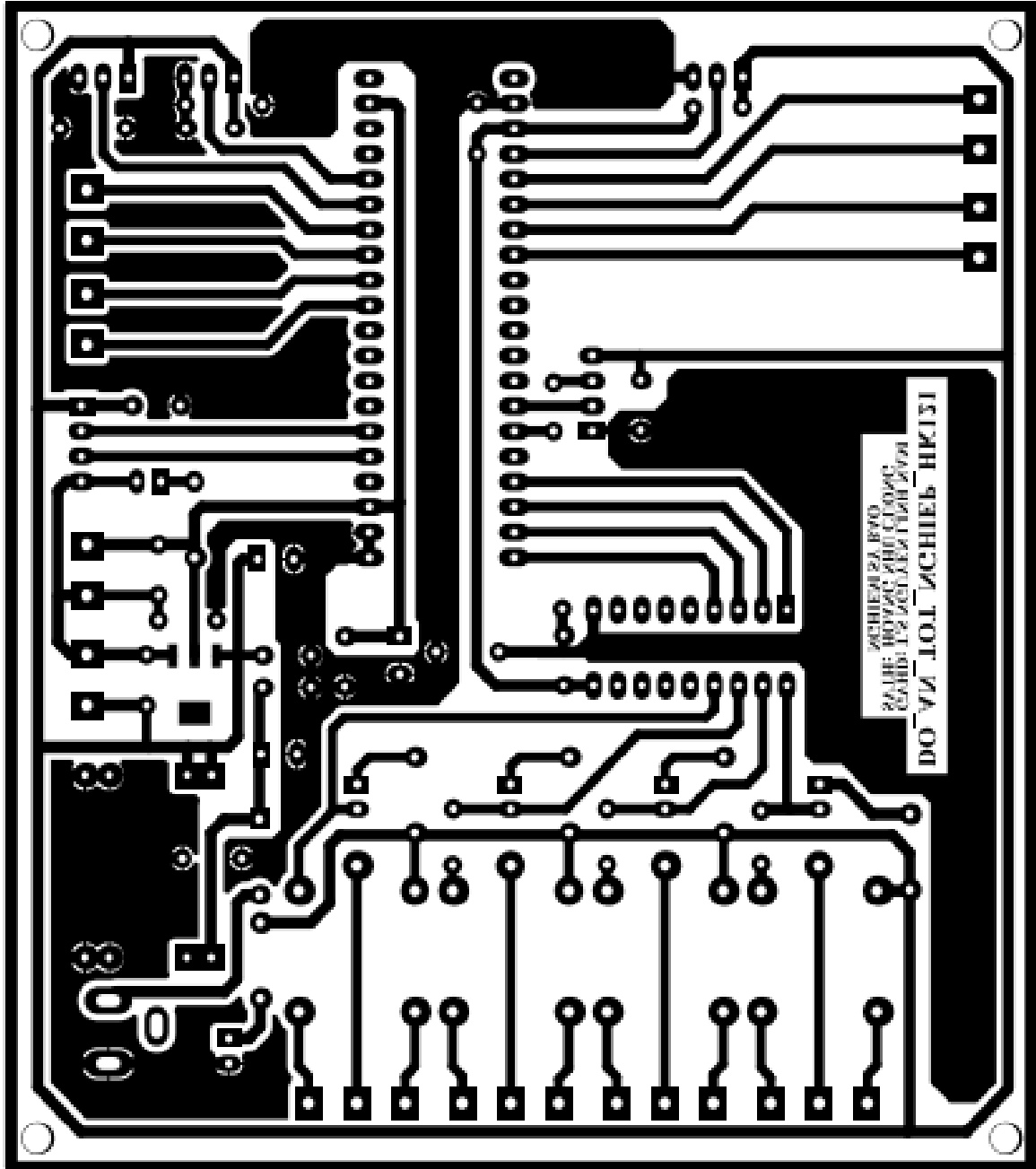


Hình 3.12. Lưu đồ thuật toán của Web server

3.3. Thi công sản phẩm

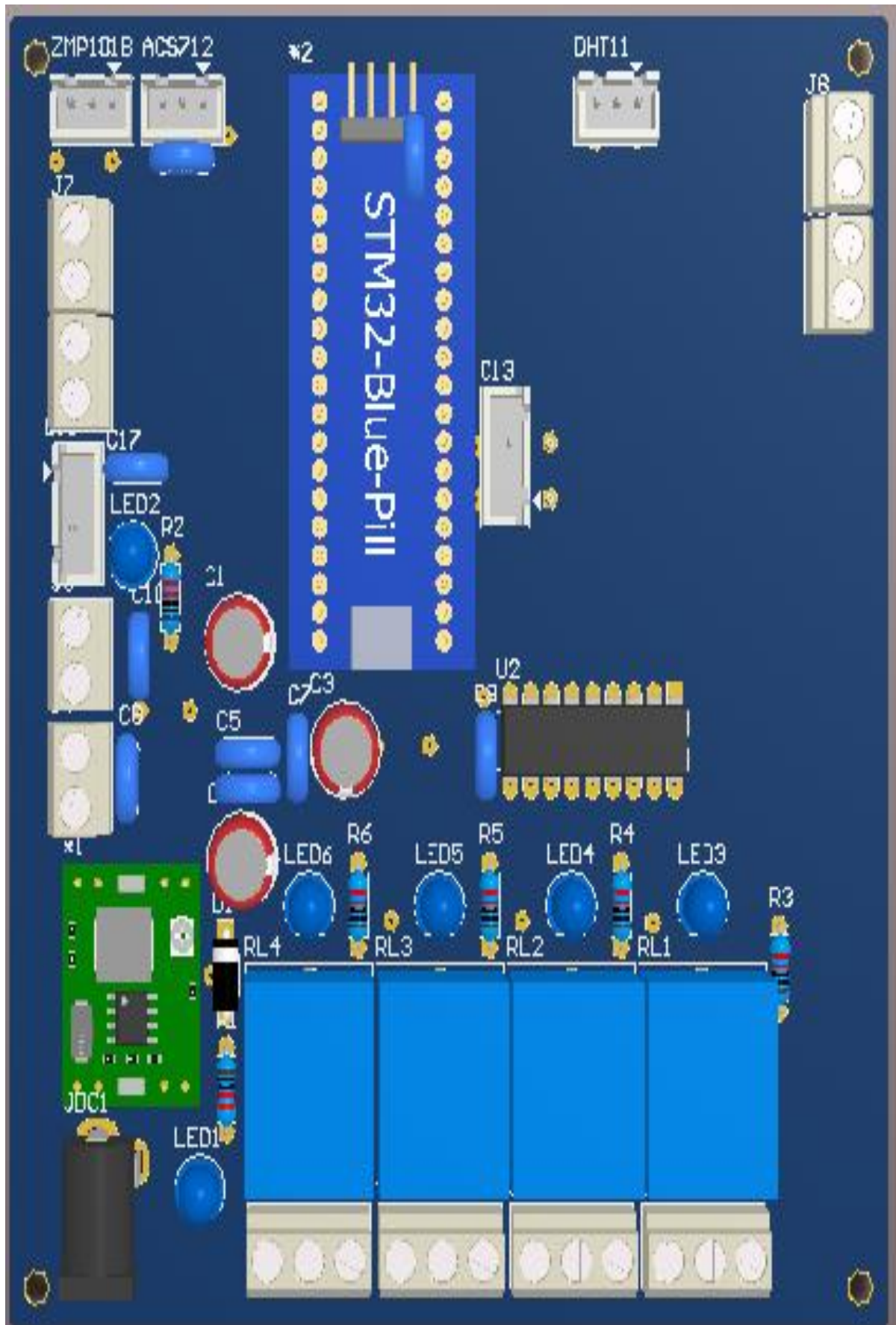
Ở phần thi công sản phẩm : nhóm em tự thiết kế các khối mạch in cho các khối của hệ thống và giới thiệu các quá trình làm việc thiết kế cũng như là thi công sản phẩm. Bên cạnh đó cũng có phần demo sản phẩm .

3.3.1. Mạch in

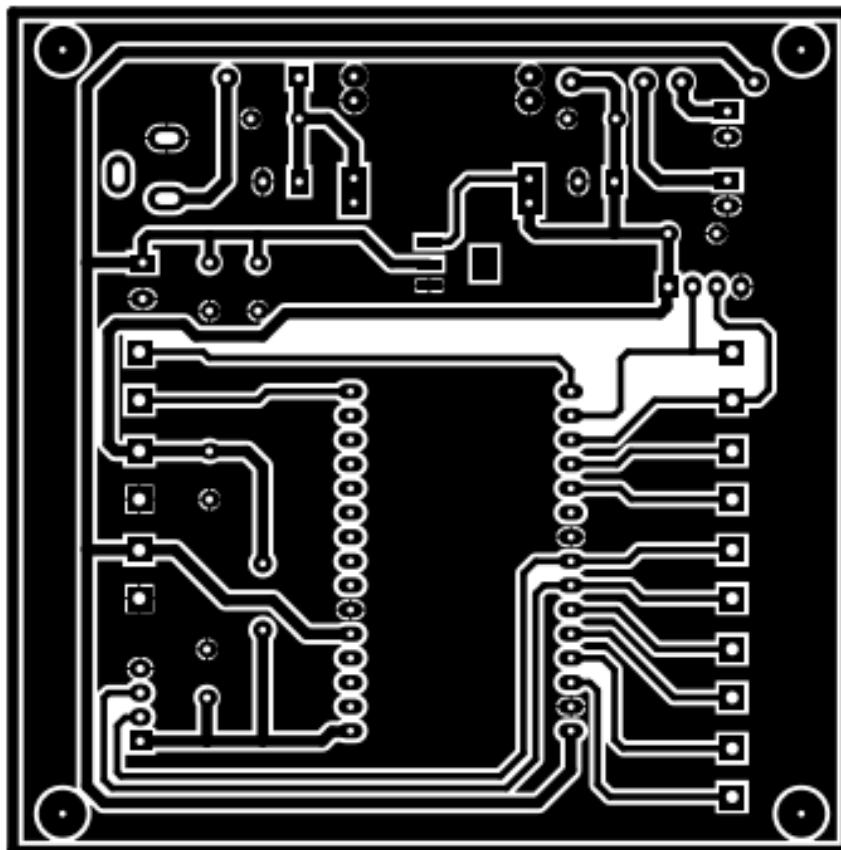
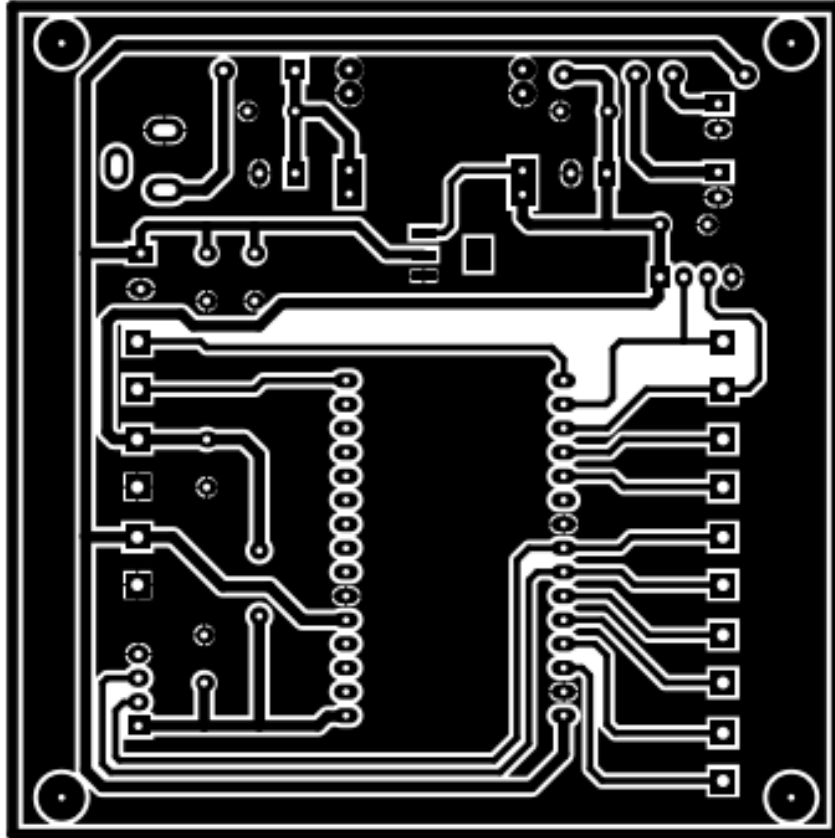


Hình 3.13. Mạch in khối xử lý

Trong hình 3.13 là sơ đồ đi dây mặt dưới của mạch điều khiển chính được vẽ bằng phần mềm Altium Designer dựa trên sơ đồ nguyên lý mà nhóm đưa ra . Các linh kiện và vị trí sẽ được nhóm bố trí và sắp xếp như hình 3.14 .

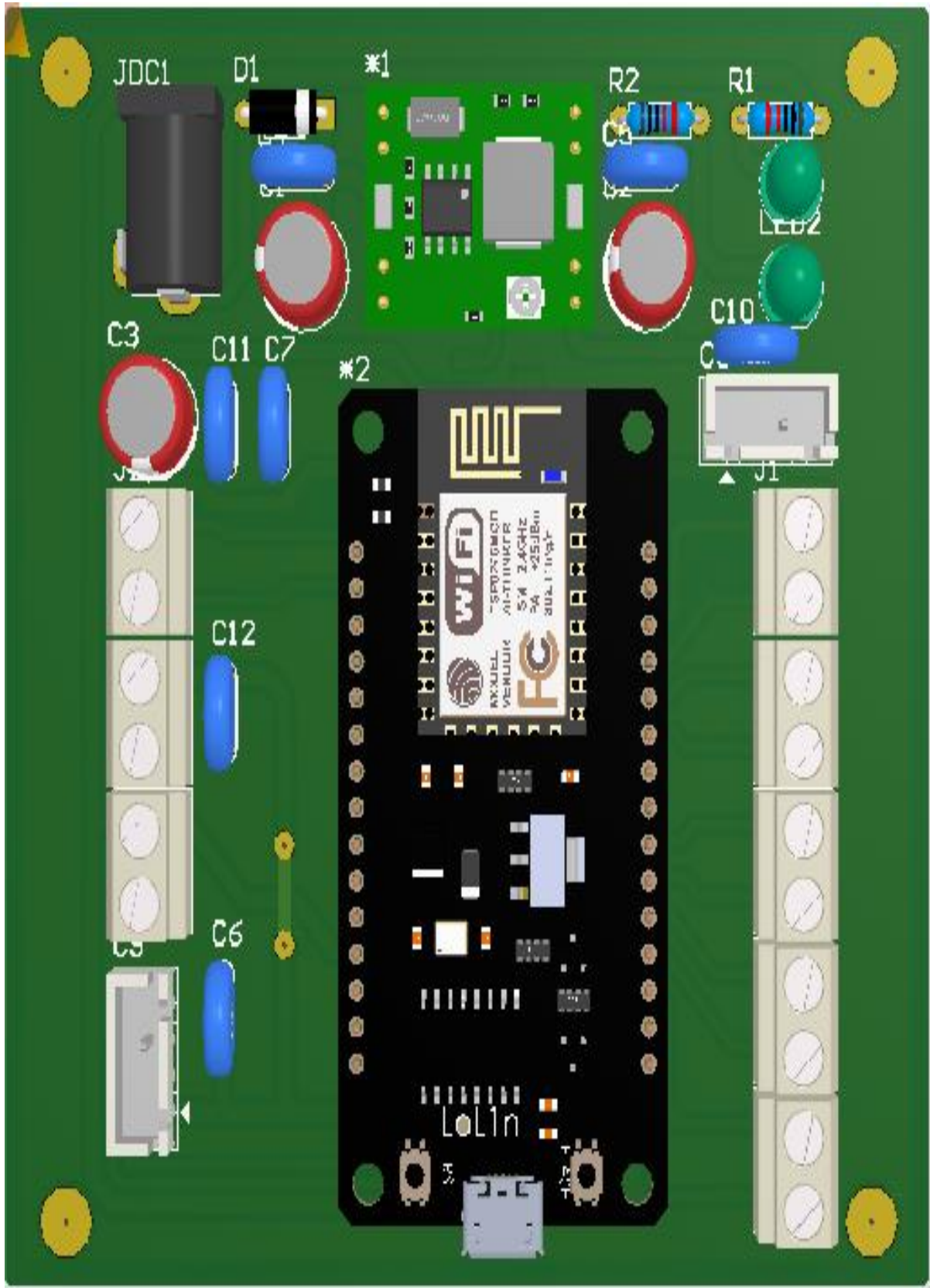


Hình 3.14. Mạch 3D khối xử lý.



Hình 3.15. Mạch in khối giao tiếp từ STM lên Webserver

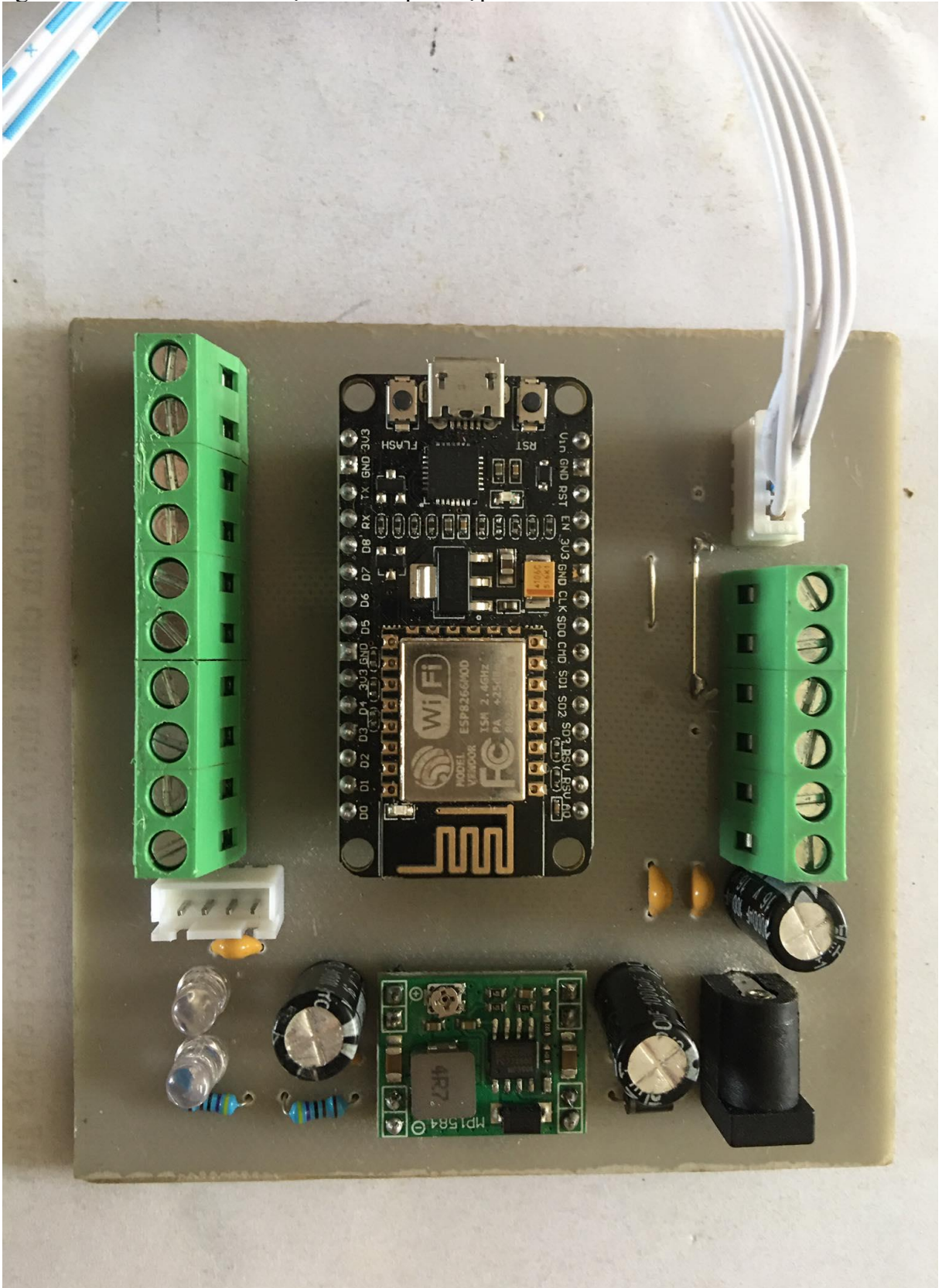
Tương tự như mạch in khối xử lý, mạch in hình 3.15. cũng được thiết kế theo sơ đồ nguyên lý và các linh kiện được bố trí và sắp xếp như hình 3.16.



Hình 3.16. Mạch 3D khối giao tiếp

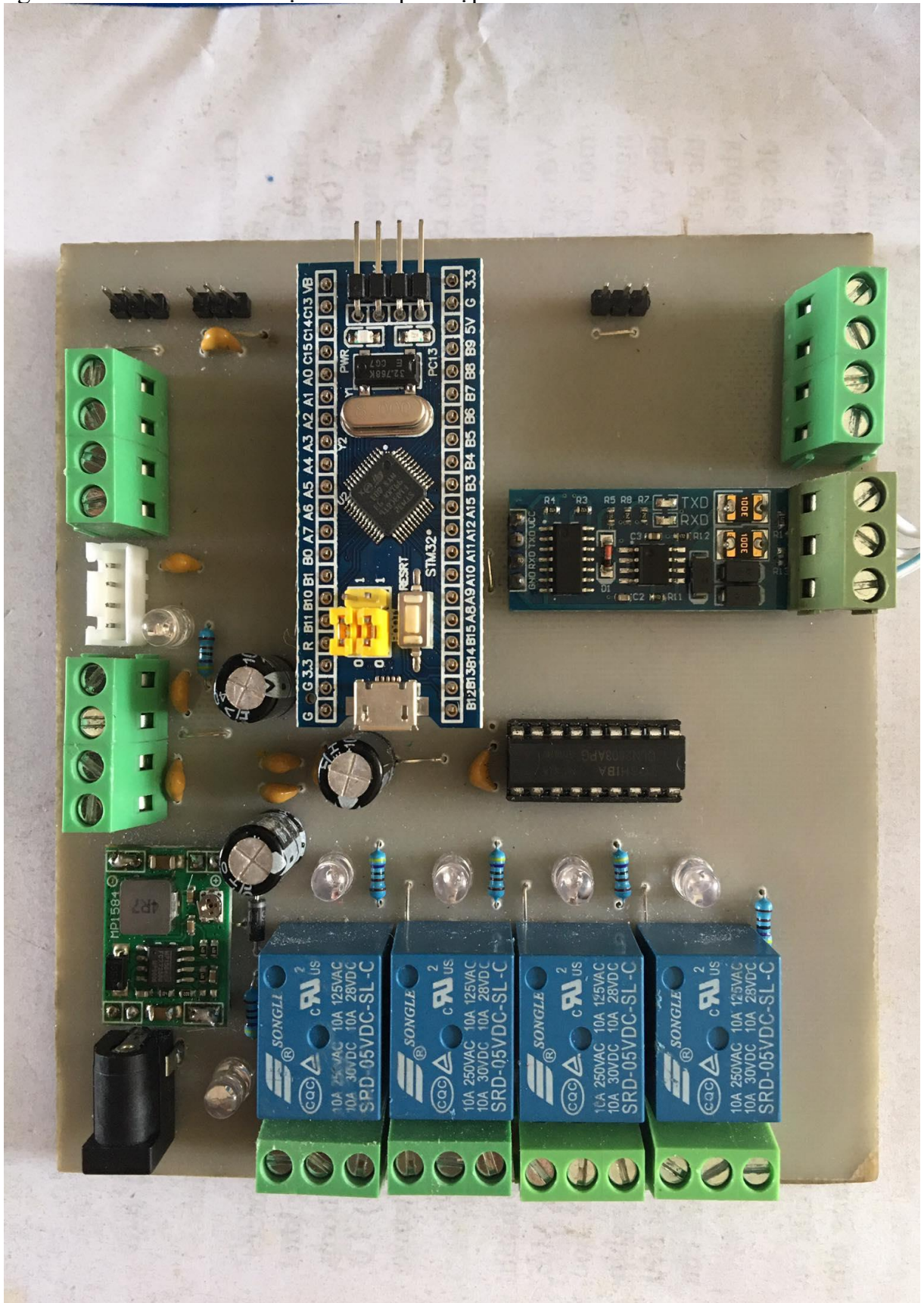
3.3.2. Mạch sau khi thi công

Nhóm hàn linh kiện và lắp ráp mạch của khối giao tiếp . Tiến hành đo và kiểm tra nguồn vào và ra rồi chỉnh lại sao cho phù hợp .



Hình 3.17. Mạch 1 sau khi thi công.

Nhóm hàn linh kiện và lắp ráp mạch của khối giao tiếp . Tiến hành đo và kiểm tra nguồn vào và ra rồi chỉnh lại sao cho phù hợp .

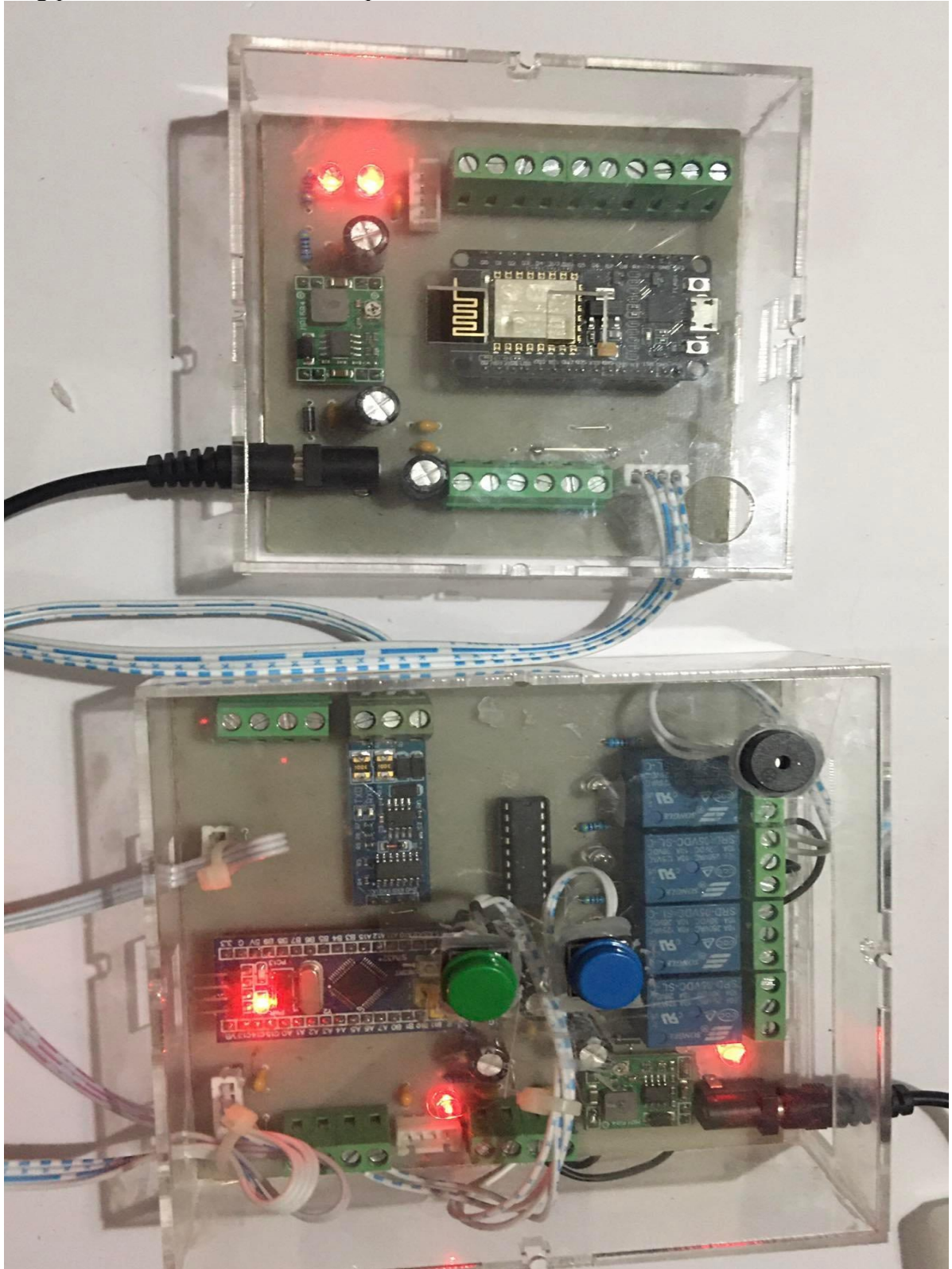


Hình 3.18. Mạch 2 sau khi thi công.

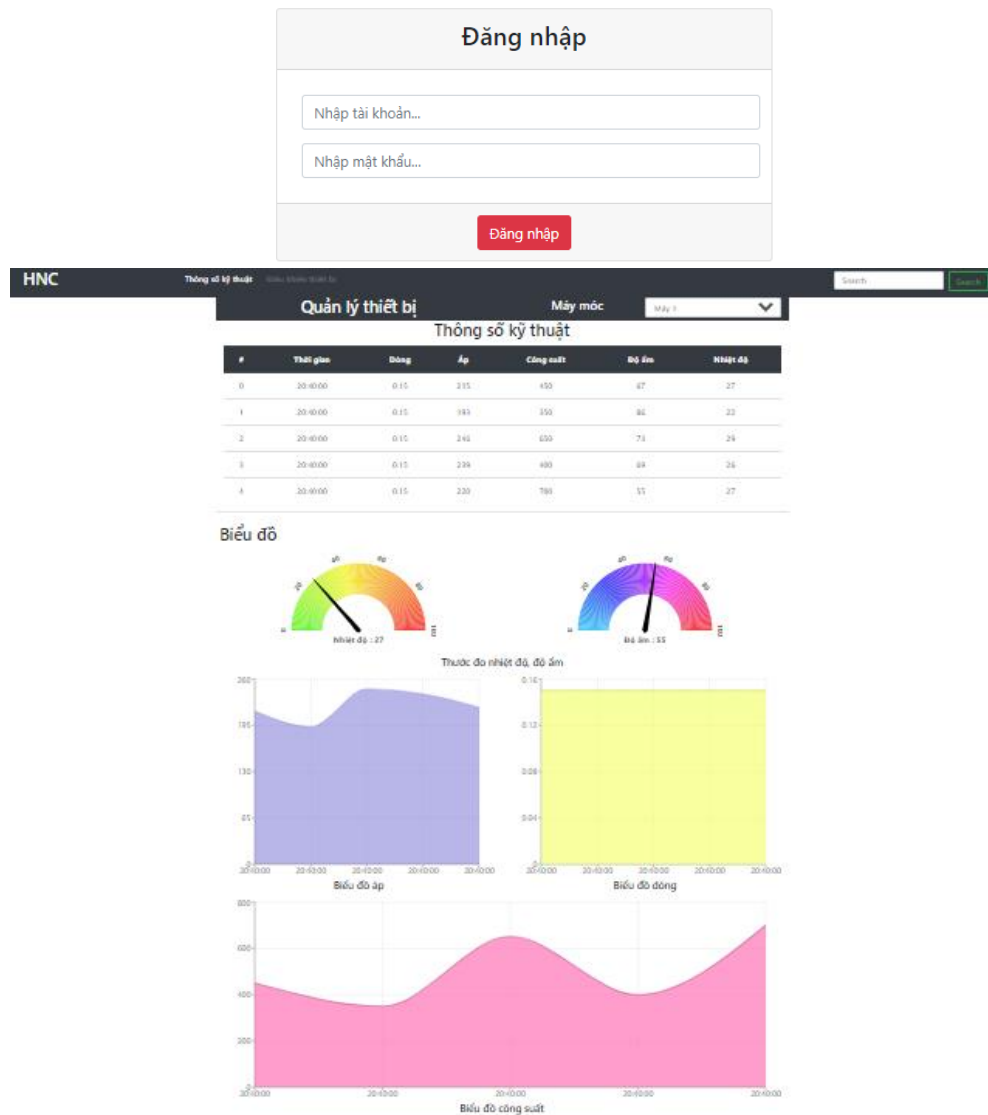
3.4. Mô hình sản phẩm và quá trình hoạt động

3.4.1. Mô hình sản phẩm

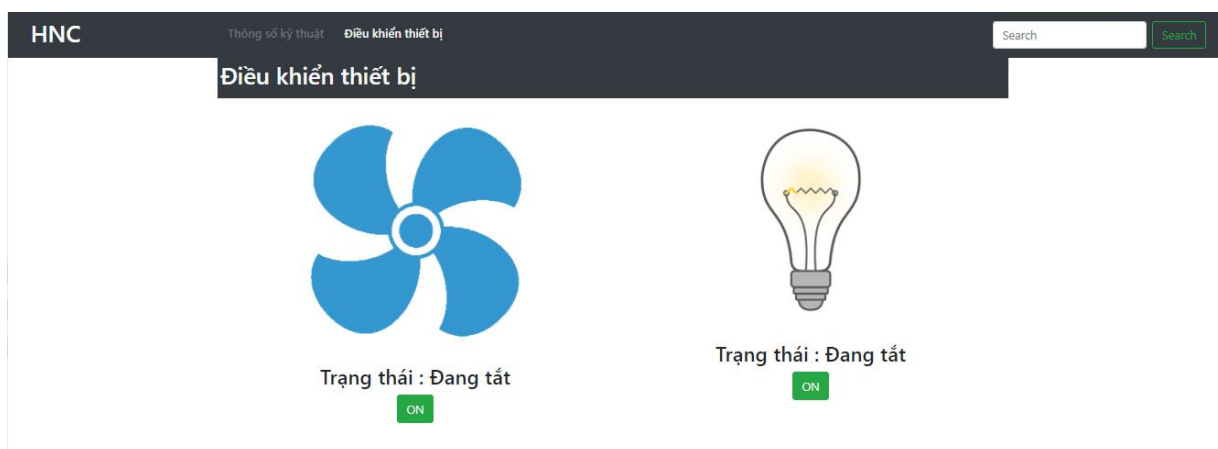
Ở đề tài này , nhóm chúng em đã tìm hiểu và sử dụng Web miễn phí và lập trình bằng phần mềm Visual studio và phần mềm STM32 cubeide



Hình 3.19. Hình ảnh mô hình sản phẩm



Hình 3.20. Hình ảnh Website.



Hình 3.21. Giao diện điều khiển thiết bị.

3.4.2. Quá trình hoạt động của hệ thống

❖ Node

Khi được cấp nguồn các Node sẽ tiến hành đọc dữ liệu thu được từ các cảm biến và sẽ đóng gói tin gửi cho khối Gateway khi có yêu cầu, đồng thời chờ tiến hiệu điều khiển từ bên gateway gửi về thông qua ModBus RTU rs485.

❖ Gateway

Sau khi khởi động nguồn, Gateway sẽ kết nối với broker hivemq thông qua Wifi. Cùng lúc đó Gateway sẽ gửi yêu cầu nhận dữ liệu thông qua Modbus RTU RS485 từ khối node. Đồng thời chờ dữ liệu từ kênh của broker gửi về.

❖ Sever

Khi có dữ liệu từ Gateway gửi lên, dữ liệu sẽ được bóc tách ra dưới dạng nhiệt độ, độ ẩm, dòng, áp với mức độ cảnh báo riêng cho từng Node. Tất cả sẽ được hiển thị dưới dạng biểu đồ rọc quan trên nền tảng Website.

Kết quả đạt được:

- Sử dụng được công nghệ , xây dựng được hệ thống điều khiển và giám sát .
- Dữ liệu gửi lên hệ thống là thời gian thực .
- Hệ thống cảnh báo các hiện tượng sự cố đúng với quy ước.
- Giám sát điều khiển thiết bị, cập nhập trạng thái thiết bị.
- Tìm hiểu từ đó làm chủ được công nghệ qua đó áp dụng trên nhiều lĩnh vực khác nhau liên quan đến đề tài .

KẾT LUẬN

Kết luận:

Sau quá trình nghiên cứu và tìm hiểu dưới sự giúp đỡ tận tình của thầy Nguyễn Linh Nam và toàn thể giảng viên của khoa điện thì chúng em cũng đã hoàn thành được mô hình sản phẩm và rút được những kinh nghiệm, các kiến thức cơ bản về ngành học. Nhóm em đã hoàn thành được mục tiêu lúc đầu đề ra:

- Xây dựng hệ thống có thể truyền tải dữ liệu từ hệ thống lên Web server
- Giám sát và điều khiển thông qua Web server
- Sử dụng giao thức Modbus rtu rs485 để giao tiếp giữa STM32 và ESP8266
- Thiết kế giao diện Web server đơn giản để giám sát và điều khiển.

Từ đó rút ra được những kiến thức cơ bản về IoT nói riêng cũng như điện tử nói chung

Ưu điểm của sản phẩm:

- Hệ thống hoạt động ổn định và chính xác với sai số không đáng kể.
- Thiết kế tiện lợi là nhỏ gọn.
- Cảnh báo sự cố chính xác.
- Tiết kiệm được chi phí.
- Sử dụng được lâu dài.
- Cần thiết và có tính ứng dụng cao.
- Tiện lợi và dễ sử dụng.
- Điều khiển dễ dàng và hoạt động ổn định.

Nhược điểm của sản phẩm:

- Do thời gian tiếp cận và hoàn thành còn chưa nhiều nên hệ thống còn chưa được tối ưu nhất.
- Nếu có thêm 1 số cảm biến thì hệ thống sẽ có nhiều chức năng với các công dụng khác.

Hướng phát triển sản phẩm:

- Nhóm em sẽ tối ưu sản phẩm để nó có thể hoạt động chính xác hơn lúc nào động cơ gặp các vấn đề khác nhau.
- Nhóm cũng sẽ hoàn thiện và nâng cấp sản phẩm hơn để sản phẩm có nhiều chức năng mới hơn. Ưu ưu sản phẩm để nó càng ngày càng nhỏ gọn, tiện ích và hiệu quả hơn.
- Nhóm sẽ hướng sản phẩm theo hướng thông minh hơn như việc điều khiển bằng giọng nói ,...
- Thêm nhiều tính năng và thiết kế giao diện điều khiển được tốt hơn .
- Mở rộng số thiết bị cần điều khiển với nhiều dây chuyền. Một Web Server có thể điều khiển giám sát nhiều nhà máy đặt ở nhiều nơi.
- Để phát triển hệ thống lớn hơn , nhiều chức năng hơn thì cần phát triển thêm đó là sử dụng một MCU có tốc độ nhanh hơn , mạnh hơn . Thay thế Wifi bằng sóng lora để truyền đi xa hơn.

TÀI LIỆU THAM KHẢO

- [1]. TS. Nguyễn Việt Lâm(2021). *Tác động của trí tuệ nhân tạo trong quan hệ quốc tế: Cơ hội, thách thức và đề xuất chính sách tham chiếu đối với Việt Nam*,
<https://www.tapchiconsan.org.vn/web/guest/the-gioi-van-de-su-kien/-/2018/821708/tac-dong-cua-tri-tue-nhan-tao-trong-quan-he-quoc-te--co-hoi%2C-thach-thuc-va-de-xuat-chinh-sach-tham-chieu-doi-voi-viet-nam.aspx>
- [2]. Chính phủ(2015). *Năng suất lao động năm 2015 tăng 6,4%*,
<http://baochinhphu.vn/Doi-song/Nang-suat-lao-dong-nam-2015-tang-64/244650.vgp>
- [3]. Nguyễn Thị Minh Thu(2017). *Phát triển khoa học công nghệ nhằm nâng cao năng suất lao động ở Việt Nam*.
- [4]. HẢI PHÒNG XANH(2021). *Khoa học công nghệ là gì? Vai trò và những ứng dụng trong cuộc sống*,
<http://haiphongxanh.net/index.php/2021/10/02/khoa-hoc-cong-nghe-la-gi-vai-tro-va-nhung-ung-dung-trong-cuoc-song/>
- [5]. UPVIET(2020). *Nhà máy thông minh (Smart factory), vai trò và xu hướng phát triển trong thời đại công nghiệp 4.0*,
<https://www.upviet.com.vn/giai-phap-tu-dong-hoa/nha-may-thong-minh-smart-factory-vai-tro-va-xu-huong-phat-trien-trong-thoi-dai-cong-nghiep-40.html>
- [6]. DACO(2016). *NHÀ MÁY THÔNG MINH (SMART FACTORY) LÀ GÌ? LÀM THẾ NÀO ĐỂ XÂY DỰNG, TRIỂN KHAI MÔ HÌNH NHÀ MÁY SẢN XUẤT THÔNG MINH?*
<https://daco.vn/san-pham/nha-may-thong-minh-la-gi-lam-the-nao-de-xay-dung-mo-hinh-san-xuat-thong-minh-7640>
- [7]. QTCO(2021). *Nhà máy thông minh là gì? Tìm hiểu 10 đặc trưng của Smart Factory*,
<https://smartfactoryvietnam.com/tim-hieu-10-dac-trung-cua-smart-factory/>
- [8]. Khu công nghiệp(2021). *IoT và Các ứng dụng của IoT trong sản xuất công nghiệp*,
<http://www.khucongnghe.com.vn/iot-va-ung-dung-trong-cong-nghiep/>
- [9]. Mona Media(2020). *IoT là gì? Những lợi ích mà Internet Of Thing đem đến*,
<https://mona.media/iot-la-gi/>
- [10]. Digital(2020). *Ứng dụng công nghệ IoT trong ngành công nghiệp sản xuất*,
<https://digital.fpt.com.vn/ung-dung-cong-nghe-iot-trong-nganh-cong-nghiep-san-xuat/>
- [11]. Topdev(2018). *Web server là gì? Hiểu rõ về web server*,

<https://topdev.vn/blog/web-server/>

[12]. Smart factory(2021). *Giao thức MQTT trong IoT là gì ? Những ứng dụng của MQTT như thế nào,*

<https://smartfactoryvn.com/technology/internet-of-things/giao-thuc-mqtt-la-gi-nhung-ung-dung-cua-mqtt-nhu-the-nao/>

[13]. Huphaco(2020). *MQTT là gì?*,

<https://huphaco.vn/mqtt-la-gi/>

[14]. Huphaco(2020). *Modbus Là Gì ? Modbus Hoạt Động Như Thế Nào,*

<https://huphaco.vn/modbus-rtu-la-gi/>

[15]. ARDUINO KIT(2020). *Lập trình Arduino là gì? Nền tảng lập trình đơn giản nhất hiện nay,*

<https://arduinokit.vn/arduino-la-gi/>

[16]. VISUALCPP(2020). *Visual Studio là gì ? Giới thiệu các tính năng phần mềm Visual Studio,*

<http://visualcpp.net/visual-studio-la-gi-gioi-thieu-phan-mem-visual-studio/>

[17]. Điện tử hello(2020). *GIỚI THIỆU VỀ STM32F103C8T6,*

<https://www.dientuhello.com/bai-1-gioi-thieu-ve-stm32f103c8t6/>

[18]. VINA FE(2021). *ESP8266 là gì,*

<https://dientutuonglai.com/esp8266-la-gi.html>

[19]. Iotmaker(2008). *NODEMCU v1.0 Lua - ESP8266 ESP12E,*

<https://iotmaker.vn/nodemcu.html>

[20]. Nguyễn Quốc(2020). *Nguyên lý hoạt động của relay,*

<https://bff-tech.com/relayla-gi-nguyen-ly-hoat-dong-cua-relay/>

[21]. Linhkienviet(2018). *Module hạ áp mini DC-DC MP1584,*

<http://linhkienviet.vn/module-ha-ap-mini-dc-dc-mp1584>

[22]. Arduinokit(2019). *Đọc nhiệt độ, độ ẩm (DHT11) sử dụng Arduino Uno,*

<https://arduinokit.vn/doc-cam-bien-nhiet-do-do-am-dht11-arduino/>

[23]. Hshop(2020). *Cảm Biến Độ Ẩm, Nhiệt Độ DHT11,*

<https://nshopvn.com/product/module-cam-bien-do-am-nhiet-do-dht11/>

[24]. Hshop(2020). *Cảm Biến Điện Áp AC Voltage Sensor ZMPT101B ,*

<https://hshop.vn/products/mach-do-dien-ac-voltage-sensor-zmpt101b>

- [25].Dienmayhoangngan(2018). *Nguồn Adapter 5V-2A 5.5x2.1mm* ,
<https://dienmayhoangngan.com/adapter-5v-2a-5-5x2-1mm-nti10056109.html>
- [26].Hshop(2020). *Cảm Biến Dòng Điện ACS712 (Hall Effect Current Sensor)* ,
<https://hshop.vn/products/cam-bien-dong-dien-hall-acs712-30a>
- [27]. Longvan(2018). *Web server là gì? Những lưu ý khi sử dụng Web server*,
<https://longvan.net/web-server.html>
- [28]. GiangPHT(2018). *Web Server là gì? Tìm hiểu cơ chế vận hành của web server*,
<https://bizflycloud.vn/tin-tuc/tat-tat-kien-thuc-co-ban-ve-web-server-ban-phai-biet-20180515115521302.htm>

PHỤ LỤC

Code khối Gateway:

```
#ifndef ESP8266
#include <ESP8266WiFi.h> // Pins for board ESP8266 Wemos-NodeMCU
#else
#include <WiFi.h>
#endif
#include <PubSubClient.h>
#include <SoftwareSerial.h>
#include <WiFiClientSecure.h>
const char* ssid = "DESKTOP";
const char* password = "12345678";
const char* mqtt_server = "91f3d37a8fd64ec8be753f1b3b279991.s2.eu.hivemq.cloud"; //
replace with your broker url
const char* mqtt_username = "HoangNhuCuong";
const char* mqtt_password = "Cuong12345";
const uint16_t mqtt_port = 8883;

WiFiClientSecure espClient;
PubSubClient client(espClient);

SoftwareSerial mySerial(12, 14); // RX, TX
char data;
//char data1;
int len ;
int i = 0 ;
char buffer[30];

String txtMsg = ""; //Chuỗi lưu giá trị gửi từ máy tính
int lastStringLength = txtMsg.length();

uint8_t Modbus_Rtu[8];
uint8_t Modbus_Rtu_Cevap[9];
uint16_t CRC_kontrol;
uint32_t Volt_1;
float Faz_1;

void setup()
{
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }
  mySerial.begin(9600);
  mySerial.setTimeout(100);
}
```

```
setup_wifi();
#ifdef ESP8266
espClient.setInsecure();
#else // for the ESP32
  espClient.setCACert(root_ca); // enable this line and the the "certificate" code for secure
connection
#endif
client.setServer(mqtt_server, mqtt_port);
client.setCallback(callback);
}
// Hàm kết nối wifi
void setup_wifi()
{
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
// Hàm call back để nhận dữ liệu
void callback(char* topic, byte* payload, unsigned int length)
{
  for (int i = 0; i < length; i++) {
    char data1 = (char)payload[i];

    mySerial.write(data1);

    Serial.print(data1);
  }

  Serial.println();
}
// Hàm reconnect thực hiện kết nối lại khi mất kết nối với MQTT Broker
void reconnect()
{
  while (!client.connected()) // Chờ tới khi kết nối
  {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP8266Client-"; // Create a random client ID
```

```
clientId += String(random(0xffff), HEX);
// Thực hiện kết nối với mqtt user và pass
if (client.connect(clientId.c_str(), mqtt_username, mqtt_password)) //kết nối vào broker
{
  //Serial.println("Đã kết nối:");
  client.subscribe("Dieu_khien"); //đăng kí nhận dữ liệu từ topic IoT47_MQTT_Test
}
else
{
  Serial.print("Lỗi:, rc=");
  Serial.print(client.state());
  Serial.println(" try again in 5 seconds");
  // Đợi 5s
  delay(5000);
}
}
}
unsigned long t;
void loop()
{
  //callback(char* topic, byte* payload, unsigned int length);

  // data1=data;
  Serial.print(data);
  while (mySerial.available()>0)
  {
    txtMsg = mySerial.readString();
    //txtMsg += inChar;
    Serial.println(txtMsg);
  }
  if (!client.connected())// Kiểm tra kết nối
    reconnect();
  client.loop();
  if (millis() - t > 50) //nếu 500 mili giây trôi qua
  {
    t = millis();
    //Serial.print("Gui tin nhan \"Xin chao\"Test");
    // client.publish("Test", "Xin chao !"); // gửi dữ liệu lên topic IoT47_MQTT_Test
    client.publish("Test", (txtMsg).c_str());
    txtMsg = "";
  }

  if (Serial.available()) {
    mySerial.write(Serial.read());
  }

  // CRC_kontrol = ModRTU_CRC(Modbus_Rtu, 6);
  // Modbus_Rtu[7] = (CRC_kontrol >> 8) & 0xff;
  // Modbus_Rtu[6] = CRC_kontrol & 0xff;
```

```
// //HAL_UART_Transmit(&huart1, Modbus_Rtu, 8, 500);
// //HAL_Delay(300);
// Volt_1 = (Modbus_Rtu_Cevap[3] << 8) | Modbus_Rtu_Cevap[4];
}

uint16_t ModRTU_CRC(uint8_t buf[8], int len)
{
    uint16_t crc = 0xFFFF;

    for (int pos = 0; pos < len; pos++) {
        crc ^= (uint16_t)buf[pos];      // XOR byte into least sig. byte of crc

        for (int i = 8; i != 0; i--) { // Loop over each bit
            if ((crc & 0x0001) != 0) { // If the LSB is set
                crc >>= 1;           // Shift right and XOR 0xA001
                crc ^= 0xA001;
            }
            else // Else LSB is not set
                crc >>= 1;           // Just shift right
        }
    }
    // Note, this number has low and high bytes swapped, so use it accordingly (or swap bytes)
    return crc;
}
//}
```

Code khối Node:

```
/* USER CODE BEGIN Header */
/**
```

```
*****
***
```

```
* @file      : main.c
* @brief     : Main program body
```

```
*****
***
```

```
* @attention
*
* <h2><center>&copy; Copyright (c) 2022 STMicroelectronics.
* All rights reserved.</center></h2>
*
* This software component is licensed by ST under BSD 3-Clause license,
* the "License"; You may not use this file except in compliance with the
* License. You may obtain a copy of the License at:
*      opensource.org/licenses/BSD-3-Clause
*
```

```
*****
***
*/
/* USER CODE END Header */
/* Includes -----*/
#include "main.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include <stdio.h>
#include <math.h>
#include <stdint.h>
#include <string.h>
/* USER CODE END Includes */

/* Private typedef -----*/
/* USER CODE BEGIN PTD */
uint16_t analog_Read[2];
//ZMPT101B
double sensorValue = 0;
int Veff;
int max_Vol[100];
int min_Vol[100];
int kz, x;
int ky;
int w;
//acs712
double Voltage = 0;
double VRMS = 0;
double AmpsRMS = 0;
int mVperAmp = 230; // this the 5A version of the ACS712 -use 100 for 20A Module and 66
for 30A Module
int readValue;
int max_Adc[100];
int min_Adc[100];
//UART
uint8_t buffer[2];
//
uint8_t TT1;
uint8_t TT2;
/* USER CODE END PTD */

/* Private define -----*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */
```

```
/* USER CODE END PM */

/* Private variables -----*/
ADC_HandleTypeDef hadc1;
DMA_HandleTypeDef hdma_adc1;

TIM_HandleTypeDef htim1;

UART_HandleTypeDef huart1;

/* USER CODE BEGIN PV */
uint8_t u8_Status1;
uint8_t u8_Status2;
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
    if (GPIO_Pin == GPIO_PIN_4) {
        while (HAL_GPIO_ReadPin(Button1_GPIO_Port, Button1_Pin) == 0) {
        }
        u8_Status1 = ~u8_Status1;
        HAL_GPIO_WritePin(RL1_GPIO_Port, RL1_Pin, u8_Status1);
        TT1 = HAL_GPIO_ReadPin(RL1_GPIO_Port, RL1_Pin);
    }
    else if (GPIO_Pin == GPIO_PIN_5) {
        while (HAL_GPIO_ReadPin(Button2_GPIO_Port, Button2_Pin) == 0) {
        }
        u8_Status2 = ~u8_Status2;
        HAL_GPIO_WritePin(RL2_GPIO_Port, RL2_Pin, u8_Status2);
        TT2 = HAL_GPIO_ReadPin(RL2_GPIO_Port, RL2_Pin);
    }
}
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_ADC1_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_TIM1_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */

//code_begin_zmpt101b
int max_dien_ap(int A[], int n) {
    int max_dien_ap = A[0];
    for (int i = 1; i < n; i++)
        if (max_dien_ap < A[i])
```

```
        max_dien_ap = A[i];
    return max_dien_ap;
}
int min_dien_ap(int A[], int n) {
    int min_dien_ap = A[0];
    for (int i = 1; i < n; i++)
        if (min_dien_ap > A[i])
            min_dien_ap = A[i];
    return min_dien_ap;
}

void Trung_binh_vol() {
    for (int e = 0; e < 100; e++) {
        HAL_ADC_Start_DMA(&hadc1, (uint32_t*) analog_Read, 2);
        sensorValue = analog_Read[1];
        max_Vol[e] = sensorValue;
        HAL_Delay(10);
    }
    kz = max_dien_ap(max_Vol, 100);
    for (int e = 0; e < 100; e++) {
        HAL_ADC_Start_DMA(&hadc1, (uint32_t*) analog_Read, 2);
        sensorValue = analog_Read[1];
        min_Vol[e] = sensorValue;
        HAL_Delay(10);
    }
    ky = min_dien_ap(min_Vol, 100);
    w = (kz - ky);
}

void read_ZMP101B() {
    Trung_binh_vol();
    x = w / 4.098754;
    Veff = x / 1.4;
    if (x < 200)
        Veff = 0;
    if (Veff > 200)
        Veff = 220;
}
//code_end_zmpt101b=====
=====

//code_begin_acs712
int max_dong(int B[], int K) {
    int max_dong = B[0];
    for (int I = 1; I < K; I++)
        if (max_dong < B[I])
            max_dong = B[I];
    return max_dong;
}
int min_dong(int B[], int K) {
    int min_dong = B[0];
```

```
    for (int I = 1; I < K; I++)
        if (min_dong > B[I])
            min_dong = B[I];
    return min_dong;
}
//float C, B, E, K;
float A_MAX, A_MIN, Trung_binh_dong, E;
void trung_binh() {

    for (int m = 0; m < 100; m++) {
        HAL_ADC_Start_DMA(&hadc1, (uint32_t*) analog_Read, 2);
        readValue = analog_Read[0];
        max_Adc[m] = readValue;
        HAL_Delay(10);
    }

    A_MAX = max_dong(max_Adc, 50);

    for (int m = 0; m < 100; m++) {
        HAL_ADC_Start_DMA(&hadc1, (uint32_t*) analog_Read, 2);
        readValue = analog_Read[0];
        min_Adc[m] = readValue;
        HAL_Delay(10);
    }

    A_MIN = min_dong(min_Adc, 50);
    Trung_binh_dong = (A_MAX - A_MIN);
    E = ((A_MAX - A_MIN) * 5.0) / 4096.0;
    //result = ((maxValue - minValue) * 5.0) / 1024.0;

}
void read_acs712() {
    trung_binh();
    Voltage = E;
    VRMS = (Voltage / 2.42100) * 0.72; //root 2 is 0.707
    AmpsRMS = (VRMS * 1000) / mVperAmp;
    if (AmpsRMS <= 0.3) {
        AmpsRMS = 0;
    }
    //Watt = (AmpsRMS * 240 / 1.3); // 1.3 is an empirical calibration factor
}
//code_end_acs712=====

//code_begin_dht11=====
#define DHT11_PORT GPIOB
#define DHT11_PIN GPIO_PIN_9
uint8_t RHI, RHD, TCI, TCD, SUM;
uint32_t pMillis, cMillis;
float tCelsius = 0;
float tFahrenheit = 0;
```



```
float RH = 0;
```

```
void microDelay(uint16_t delay) {  
    __HAL_TIM_SET_COUNTER(&htim1, 0);  
    while (__HAL_TIM_GET_COUNTER(&htim1) < delay)  
        ;  
}
```

```
uint8_t DHT11_Start(void) {  
    uint8_t Response = 0;  
    GPIO_InitTypeDef GPIO_InitStructPrivate = { 0 };  
    GPIO_InitStructPrivate.Pin = DHT11_PIN;  
    GPIO_InitStructPrivate.Mode = GPIO_MODE_OUTPUT_PP;  
    GPIO_InitStructPrivate.Speed = GPIO_SPEED_FREQ_LOW;  
    GPIO_InitStructPrivate.Pull = GPIO_NOPULL;  
    HAL_GPIO_Init(DHT11_PORT, &GPIO_InitStructPrivate); // set the pin as output  
    HAL_GPIO_WritePin(DHT11_PORT, DHT11_PIN, 0); // pull the pin low  
    HAL_Delay(20); // wait for 20ms  
    HAL_GPIO_WritePin(DHT11_PORT, DHT11_PIN, 1); // pull the pin high  
    microDelay(30); // wait for 30us  
    GPIO_InitStructPrivate.Mode = GPIO_MODE_INPUT;  
    GPIO_InitStructPrivate.Pull = GPIO_PULLUP;  
    HAL_GPIO_Init(DHT11_PORT, &GPIO_InitStructPrivate); // set the pin as input  
    microDelay(40);  
    if (!(HAL_GPIO_ReadPin(DHT11_PORT, DHT11_PIN))) {  
        microDelay(80);  
        if ((HAL_GPIO_ReadPin(DHT11_PORT, DHT11_PIN)))  
            Response = 1;  
    }  
    pMillis = HAL_GetTick();  
    cMillis = HAL_GetTick();  
    while ((HAL_GPIO_ReadPin(DHT11_PORT, DHT11_PIN)) && pMillis + 2 >  
cMillis) {  
        cMillis = HAL_GetTick();  
    }  
    return Response;  
}
```

```
uint8_t DHT11_Read(void) {  
    uint8_t a, b;  
    for (a = 0; a < 8; a++) {  
        pMillis = HAL_GetTick();  
        cMillis = HAL_GetTick();  
        while (!(HAL_GPIO_ReadPin(DHT11_PORT, DHT11_PIN))  
            && pMillis + 2 > cMillis) { // wait for the pin to go high  
            cMillis = HAL_GetTick();  
        }  
        microDelay(40); // wait for 40 us  
        if (!(HAL_GPIO_ReadPin(DHT11_PORT, DHT11_PIN))) // if the pin is low  
            b &= ~(1 << (7 - a));  
    }  
}
```

```
        else
            b |= (1 << (7 - a));
        pMillis = HAL_GetTick();
        cMillis = HAL_GetTick();
        while ((HAL_GPIO_ReadPin(DHT11_PORT, DHT11_PIN))
                && pMillis + 2 > cMillis) { // wait for the pin to go low
            cMillis = HAL_GetTick();
        }
    }
    return b;
}
//code_end_dht11=====

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart) {
    if (huart->Instance == huart1.Instance) {
        dieu_khien();
        // HAL_UART_Receive_IT(&huart1, buffer, 1);
    }
    HAL_UART_Receive_IT(&huart1, buffer, 1);
}
//=====
char NHIET_DO[18];
char DONG[17];
char DO_AM[15];
char AP[17];
char data[100];
void Transmit_data() {
    TT1 = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_12);
    TT2 = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_13);
    read_ZMP101B();
    DHT11_Start();
    read_acs712();
    sprintf(NHIET_DO, "Nhiệt độ:%d", TCI);
    sprintf(DO_AM, "Độ am:%d", RHI);
    sprintf(AP, "Điện áp:%d", (int) Veff);
    sprintf(DONG, "Dòng điện:%d", (int) (AmpsRMS * 100));
    sprintf(data, " ND:%d DA:%d AP:%d D:%d TT1:%d TT2:%d ", TCI, RHI,
              (int) Veff, (int) (AmpsRMS * 100), TT1, TT2);

    HAL_UART_Transmit_IT(&huart1, data, 50);
    HAL_Delay(10);
}
//=====
void dieu_khien() {
    if (buffer[0] == '1') {
        HAL_GPIO_WritePin(RL1_GPIO_Port, RL1_Pin, 0); //OFF
        buffer[0] = ' ';
    }
    if (buffer[0] == '2') {
        HAL_GPIO_WritePin(RL1_GPIO_Port, RL1_Pin, 1); //ON
    }
}
```

```
        buffer[0] = ' ';
    }
    if (buffer[0] == '3') {
        HAL_GPIO_WritePin(RL2_GPIO_Port, RL2_Pin, 0); //OFF
        buffer[0] = ' ';
    }
    if (buffer[0] == '4') {
        HAL_GPIO_WritePin(RL2_GPIO_Port, RL2_Pin, 1); //ON
        buffer[0] = ' ';
    }
}
//=====
/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_ADC1_Init();
    MX_USART1_UART_Init();
    MX_TIM1_Init();
    /* USER CODE BEGIN 2 */
    //    Transmit_data();
    HAL_TIM_Base_Start(&htim1);
    HAL_UART_Receive_IT(&huart1, buffer, 1);
}
```

```
    HAL_ADC_Start_DMA(&hadc1, (uint32_t*) analog_Read, 2);
    HAL_Delay(100);
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
    while (1) {
        dieu_khien();
        Transmit_data();
        if (DHT11_Start()) {
            RHI = DHT11_Read(); // Tích phân độ ẩm tương ứng
            RHD = DHT11_Read(); // Độ ẩm tương đối thập phân
            TCI = DHT11_Read(); // Tích phân độ C
            TCD = DHT11_Read(); // Thập phân độ C
        }
        /// if(TCI>=30){
        //     HAL_GPIO_WritePin(Speaker_GPIO_Port, Speaker_Pin, 1);
        //     HAL_Delay(1000);
        /// }
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
    }
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

    /** Initializes the RCC Oscillators according to the specified parameters
     * in the RCC_OscInitTypeDef structure.
     */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
    RCC_OscInitStruct.HSIState = RCC_HSI_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }
}
```

```
}
/** Initializes the CPU, AHB and APB buses clocks
*/
RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
        |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
{
    Error_Handler();
}
PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_ADC;
PeriphClkInit.AdcClockSelection = RCC_ADCCLK2_DIV6;
if (HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit) != HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief ADC1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_ADC1_Init(void)
{
    /* USER CODE BEGIN ADC1_Init 0 */

    /* USER CODE END ADC1_Init 0 */

    ADC_ChannelConfTypeDef sConfig = {0};

    /* USER CODE BEGIN ADC1_Init 1 */

    /* USER CODE END ADC1_Init 1 */
    /** Common config
    */
    hadc1.Instance = ADC1;
    hadc1.Init.ScanConvMode = ADC_SCAN_ENABLE;
    hadc1.Init.ContinuousConvMode = DISABLE;
    hadc1.Init.DiscontinuousConvMode = DISABLE;
    hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
    hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
    hadc1.Init.NbrOfConversion = 2;
    if (HAL_ADC_Init(&hadc1) != HAL_OK)

```

```
{
    Error_Handler();
}
/** Configure Regular Channel
 */
sConfig.Channel = ADC_CHANNEL_0;
sConfig.Rank = ADC_REGULAR_RANK_1;
sConfig.SamplingTime = ADC_SAMPLETIME_239CYCLES_5;
if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
{
    Error_Handler();
}
/** Configure Regular Channel
 */
sConfig.Channel = ADC_CHANNEL_1;
sConfig.Rank = ADC_REGULAR_RANK_2;
if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN ADC1_Init 2 */

/* USER CODE END ADC1_Init 2 */

}

/**
 * @brief TIM1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM1_Init(void)
{
    /* USER CODE BEGIN TIM1_Init 0 */

    /* USER CODE END TIM1_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

    /* USER CODE BEGIN TIM1_Init 1 */

    /* USER CODE END TIM1_Init 1 */
    htim1.Instance = TIM1;
    htim1.Init.Prescaler = 71;
    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim1.Init.Period = 65535;
    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    htim1.Init.RepetitionCounter = 0;
}
```

```
htim1.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
if (HAL_TIM_Base_Init(&htim1) != HAL_OK)
{
    Error_Handler();
}
sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
if (HAL_TIM_ConfigClockSource(&htim1, &sClockSourceConfig) != HAL_OK)
{
    Error_Handler();
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM1_Init 2 */

/* USER CODE END TIM1_Init 2 */

}

/**
 * @brief USART1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_USART1_UART_Init(void)
{
    /* USER CODE BEGIN USART1_Init 0 */

    /* USER CODE END USART1_Init 0 */

    /* USER CODE BEGIN USART1_Init 1 */

    /* USER CODE END USART1_Init 1 */
    huart1.Instance = USART1;
    huart1.Init.BaudRate = 9600;
    huart1.Init.WordLength = UART_WORDLENGTH_8B;
    huart1.Init.StopBits = UART_STOPBITS_1;
    huart1.Init.Parity = UART_PARITY_NONE;
    huart1.Init.Mode = UART_MODE_TX_RX;
    huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart1.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart1) != HAL_OK)
    {
        Error_Handler();
    }
    /* USER CODE BEGIN USART1_Init 2 */
```

```
/* USER CODE END USART1_Init 2 */  
  
}  
  
/**  
 * Enable DMA controller clock  
 */  
static void MX_DMA_Init(void)  
{  
  
/* DMA controller clock enable */  
__HAL_RCC_DMA1_CLK_ENABLE();  
  
/* DMA interrupt init */  
/* DMA1_Channel1_IRQn interrupt configuration */  
HAL_NVIC_SetPriority(DMA1_Channel1_IRQn, 0, 0);  
HAL_NVIC_EnableIRQ(DMA1_Channel1_IRQn);  
  
}  
  
/**  
 * @brief GPIO Initialization Function  
 * @param None  
 * @retval None  
 */  
static void MX_GPIO_Init(void)  
{  
GPIO_InitTypeDef GPIO_InitStructure = {0};  
  
/* GPIO Ports Clock Enable */  
__HAL_RCC_GPIOD_CLK_ENABLE();  
__HAL_RCC_GPIOA_CLK_ENABLE();  
__HAL_RCC_GPIOB_CLK_ENABLE();  
  
/*Configure GPIO pin Output Level */  
HAL_GPIO_WritePin(GPIOB, RL1_Pin|RL2_Pin|Speaker_Pin|GPIO_PIN_15  
|GPIO_PIN_9, GPIO_PIN_RESET);  
  
/*Configure GPIO pins : Button1_Pin Button2_Pin */  
GPIO_InitStructure.Pin = Button1_Pin|Button2_Pin;  
GPIO_InitStructure.Mode = GPIO_MODE_IT_FALLING;  
GPIO_InitStructure.Pull = GPIO_PULLUP;  
HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);  
  
/*Configure GPIO pins : RL1_Pin RL2_Pin Speaker_Pin PB15  
PB9 */  
GPIO_InitStructure.Pin = RL1_Pin|RL2_Pin|Speaker_Pin|GPIO_PIN_15  
|GPIO_PIN_9;  
GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
```



```
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/* EXTI interrupt init*/
HAL_NVIC_SetPriority(EXTI4_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(EXTI4_IRQn);

HAL_NVIC_SetPriority(EXTI9_5_IRQn, 0, 0);
HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);

}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
/* USER CODE BEGIN Error_Handler_Debug */
/* User can add his own implementation to report the HAL error return state */
__disable_irq();
while (1) {
}
/* USER CODE END Error_Handler_Debug */
}
```