

## TÓM TẮT

Tên đề tài:

*Thiết kế tích hợp module truyền thông IoT công nghiệp cho thiết bị Shinko WCL-13A*

Sinh viên thực hiện:

Lê Hoài Nam

MSSV: 105160185

Lớp: 16TDH

Nguyễn Đình Dũng

MSSV: 105160185

Lớp: 16TDH

Thiết bị điều khiển nhiệt độ WCL-13A của công ty Shinko Technos. WCL-13A là bộ điều khiển nhiệt độ cho nhiều loại máy khác nhau, như máy ép phun, máy đùn, máy đóng gói, máy sấy và lò điện dựa trên nhiệt độ ngõ vào. Các loại đầu vào có thể là cặp nhiệt điện, đầu vào RTD, đầu vào dòng, đầu vào điện áp. Đầu ra điều khiển có thể là dòng/áp hoặc điều khiển trực tiếp dùng relay hoặc SSR. WCL-13A hỗ trợ thiết lập các chức năng sản phẩm từ máy tính bằng cáp chuyên dụng của hãng thông qua các chuẩn truyền thông công nghiệp Modbus protocol, Shinko protocol.

Tuy nhiên, việc cấu hình thiết bị phải được thao tác trực tiếp trên thiết bị và yêu cầu người sử dụng có chuyên môn về kỹ thuật. Ngoài ra, không thể quan sát hết thông số của thiết bị vì màn hình LCD của thiết bị hiển thị rất hạn chế.

Từ những vấn đề hiện tại và yêu cầu của đề tài, nhóm đưa ra ý tưởng thiết kế một *module truyền thông IoT công nghiệp*, kết nối với thiết bị qua chuẩn truyền thông Modbus-RTU. Từ đó có thể cấu hình thiết bị từ laptop, điện thoại thông qua web server nhúng trên Module. Module tích hợp thẻ nhớ và kết nối internet, sao lưu dữ liệu đọc được từ thiết bị vào thẻ nhớ và gửi lên Server.

## **LỜI NÓI ĐẦU**

Quá trình thực tập và làm đồ án tốt nghiệp tại công ty Shinko Technos cho em cơ hội tổng hợp và hệ thống hóa lại những kiến thức đã học, đồng thời kết hợp với thực tế để nâng cao kiến thức chuyên môn. Trong quá trình làm đồ án doanh nghiệp, em đã được mở rộng tầm nhìn và tiếp thu rất nhiều kiến thức thực tế.

Từ đó em nhận thấy, việc cọ xát thực tế là vô cùng quan trọng – nó giúp sinh viên xây dựng nền tảng lý thuyết được học ở trường vững chắc hơn. Trong quá trình làm đồ án, từ chỗ còn bỡ ngỡ cho đến thiếu kinh nghiệm, em đã gặp phải rất nhiều khó khăn nhưng với sự giúp đỡ tận tình của quý thầy cô Khoa Điện và sự nhiệt tình của các anh chị trong Công ty SHINKO TECHNOS đã giúp em có được những kinh nghiệm quý báu để hoàn thành tốt đồ án tốt nghiệp này. Em xin chân thành cảm ơn.

Lời cảm ơn đầu tiên em xin gửi đến ban lãnh đạo cùng các phòng ban, các cô chú, anh chị trong Công ty SHINKO TECHNOS – đơn vị đã tiếp nhận và nhiệt tình tạo mọi điều kiện thuận lợi cho em trong quá trình thực tập cũng như làm đồ án.

Em cũng xin gửi lời cảm ơn chân thành đến Ban giám hiệu trường Đại Học Bách Khoa – Đại Học Đà Nẵng, quý thầy cô Khoa Điện đã tận tâm giảng dạy và truyền đạt những kiến thức, kinh nghiệm quý báu cho em. Đặc biệt, em xin cảm ơn Thầy Ngô Đình Thanh, người đã tận tình hướng dẫn em hoàn thành bài báo cáo này.

Vì thời gian và kiến thức còn hạn hẹp nên bài báo cáo không thể tránh khỏi những thiếu sót, rất mong sự góp ý của Công ty, quý thầy cô và các bạn, để em rút kinh nghiệm và hoàn thành tốt hơn. Em xin chân thành cảm ơn!

## **CAM ĐOAN**

“Em xin cam đoan đề tài: “Thiết kế tích hợp module truyền thông IoT công nghiệp cho thiết bị Shinko WCL-13A” là một công trình nghiên cứu độc lập dưới sự hướng dẫn của giáo viên hướng dẫn: TS Ngô Đình Thanh. Ngoài ra không có bất cứ sự sao chép của người khác. Đề tài, nội dung báo cáo thực tập là sản phẩm mà em đã nỗ lực nghiên cứu trong quá trình học tập tại trường cũng như tham gia thực tập tại công ty Shinko Technos. Các số liệu, kết quả trình bày trong báo cáo là hoàn toàn trung thực, em xin chịu hoàn toàn trách nhiệm, kỷ luật của bộ môn và nhà trường đề ra nếu như có vấn đề xảy ra.”

Sinh viên thực hiện

Lê Hoài Nam

## MỤC LỤC

MỞ ĐẦU .....	1
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.....	2
1.1    Phân tích đề tài.....	2
1.1.1.    Số liệu và dữ liệu ban đầu.....	2
1.1.2.    Yêu cầu công nghệ.....	2
1.2    Nội dung nghiên cứu.....	3
1.2.1.    Tổng quan thiết kế .....	3
1.2.2.    Sơ đồ khối tổng quan .....	4
1.3    Kế hoạch thực hiện .....	4
1.3.1.    Phân chia nhiệm vụ.....	4
1.3.2.    Dòng thời gian .....	5
CHƯƠNG 2: Thiết kế phần cứng.....	7
2.1    Tìm hiểu thiết bị điều khiển nhiệt độ BCS2 .....	7
2.2    Tìm hiểu công nghệ .....	9
2.2.1.    Modbus .....	9
2.2.2.    Công nghệ Internet.....	14
2.3    Phương án thiết kế .....	15
2.3.1.    Bộ xử lý trung tâm .....	15
2.3.2.    Tính chọn ic kết nối mạng LAN .....	21
2.3.3.    Module 4G .....	24
2.3.4.    Bộ chuyển đổi RS485-TTL .....	26
2.3.5.    Hệ thống thời gian thực Realtime System .....	27
2.3.6.    Thẻ nhớ SD Card .....	29
2.3.7.    Khởi nút nhấn điều khiển.....	29
2.3.8.    Khởi hiển thị .....	30
2.3.9.    Khởi nguồn .....	32
2.3.10.    Cảm biến nhiệt độ K .....	35

2.4	Sơ đồ khối nguyên lý chi tiết .....	35
2.5	Thi công phần cứng.....	37
2.5.1.	Sơ đồ mạch nguyên lý .....	37
2.5.2.	Thiết kế phần cứng .....	39
2.5.3.	Lắp ráp và kiểm tra .....	40
CHƯƠNG 3: Thiết kế phần mềm và đánh giá kết quả .....		43
3.1	Tính năng thiết bị .....	43
3.2	Tìm hiểu phần mềm .....	43
3.2.1.	Đọc ghi dữ liệu với thiết bị điều khiển nhiệt độ BCS2 .....	43
3.2.2.	Webserver .....	45
3.3	Lưu đồ thuật toán .....	49
3.3.1.	Chương trình tổng quan .....	49
3.3.2.	Chương trình khi xử lý yêu cầu từ máy khách .....	51
3.3.3.	Chương trình xử lý yêu cầu từ máy khách kèm tham số.....	52
3.4	Chạy thử phần mềm .....	54
3.4.1.	Chức năng cấu hình .....	55
3.4.2.	Chức năng lưu file vào thẻ nhớ.....	55
3.4.3.	Chức năng tải file lên Server .....	56
3.4.4.	Chức năng tải file từ Server .....	57
3.4.5.	Chức năng OTA Update .....	58
3.4.6.	Chức năng hiển thị giá trị PV .....	59
3.5	Kết quả, đánh giá .....	60
3.5.1.	Kết quả.....	60
3.5.2.	Đánh giá.....	61
TÀI LIỆU THAM KHẢO .....		63

## DANH SÁCH CÁC BẢNG, HÌNH VẼ

Hình 1.1 Mô hình tổng quan.....	4
Hình 2.1 Dòng sản phẩm BCx2.....	7
Hình 2.2 Ví dụ về điều khiển chương trình theo nhiệt.....	8
Hình 2.3 Kích thước bảng điều khiển của BCS2 .....	8
Hình 2.4 Sơ đồ kết nối BCS2 và PC.....	9
Hình 2.5 Mô hình Modbus-RTU theo nguyên lý master-slave .....	10
Hình 2.6 Sơ đồ nối dây BCS2 .....	10
Hình 2.7 Cấu trúc khung truyền Modbus-RTU .....	11
Hình 2.8 Ví dụ lệnh ghi giá trị cho SV .....	11
Hình 2.9 Chương trình tính mã CRC-16. ....	12
Hình 2.10 Truyền thông modbus với thiết bị BCS2.....	13
Hình 2.11 Phần mềm Hercules .....	13
Hình 2.12 Cổng mạng LAN RJ45 .....	14
Hình 2.13 Mô hình mạng LAN. ....	14
Hình 2.14 Sơ đồ khối tổng quan.....	15
Hình 2.15 Chip ESP32-WROOM-32 .....	16
Hình 2.16 Sơ đồ nguyên lý mạch nguồn nuôi .....	17
Hình 2.17 Sơ đồ nguyên lý mainboard ESP32.....	18
Hình 2.18 Switch bounce trên nút nhấn .....	18
Hình 2.19 Sơ đồ nguyên lý mạch nạp .....	19
Hình 2.20 Sơ đồ chân ESP32 DEVKIT V1 - DOIT.....	20
Hình 2.21 Sơ đồ mạch nguyên lý LAN8720 .....	21
Hình 2.22 Mạch dao động thạch anh cho LAN8720.....	21
Hình 2.23 Nguồn nuôi cho LAN8720 .....	22
Hình 2.24 Module LAN8720 .....	23
Hình 2.25 Cấu trúc liên kết vi điều khiển và module mạng LAN.....	23
Hình 2.26 Thử nghiệm kết nối mạng LAN với LAN8720.....	24
Hình 2.27 Dải băng tần mạng 4G tại Việt Nam .....	24
Hình 2.28 Module SIM7600.....	25
Hình 2.29 Mô hình kết nối ESP32,SIM, Server .....	26
Hình 2.30 Module chuyển đổi RS485-TTL .....	26
Hình 2.31 Chức năng chuyển đổi mức logic 3.3v-5v.....	26
Hình 2.32 Sơ đồ nối dây mạch chuyển giao tiếp TTL-RS485 và ESP32 .....	27
Hình 2.33 Sơ đồ nguyên lý DS3231 .....	27

Hình 2.34 Module thời gian thực DS3231 .....	28
Hình 2.35 Module thẻ nhớ SD Card.....	29
Hình 2.36 Mạch nguyên lý nút nhấn điều khiển .....	30
Hình 2.37 Nút nhấn vật lý .....	30
Hình 2.38 Màn hình LCD.....	31
Hình 2.39 Module I2C.....	31
Hình 2.40 Sơ đồ kết nối LCD và Chip I2C PCF8574 .....	32
Hình 2.41 Công suất tiêu thụ trên module RS485-TTL .....	34
Hình 2.42 Module nguồn HLK-PM01 .....	34
Hình 2.43 Cảm biến nhiệt độ K.....	35
Hình 2.44 Sơ đồ khối phương án thiết kế chi tiết.....	36
Hình 2.45 Sơ đồ mạch nguyên lý .....	37
Hình 2.46 Hộp nhựa PLC .....	39
Hình 2.47 Mạch in pdf.....	40
Hình 2.48 Mạch in 1 lớp trên bo đồng .....	40
Hình 2.49 Mạch thiết bị sau khi gắn linh kiện .....	41
Hình 2.50 Thiết bị sau khi đóng hộp .....	42
Hình 3.1 Tin nhắn yêu cầu đọc giá trị PV .....	43
Hình 3.2 Tin nhắn phản hồi giá trị PV .....	43
Hình 3.3 Gói tin phản hồi ghi giá trị SV .....	44
Hình 3.4 Nguyên lý hoạt động của Web Server.....	45
Hình 3.5 Phần mềm Arduino IDE .....	46
Hình 3.6 Mô hình hoạt động của DNS .....	47
Hình 3.7 Lưu đồ thuật toán tổng quan.....	49
Hình 3.8 Lưu đồ thuật toán tổng quan xử lý request.....	51
Hình 3.9 Giao diện trang chủ .....	52
Hình 3.10 Giao diện cấu hình modbus.html.....	53
Hình 3.11 Đường dẫn gồm các tham giá cấu hình .....	54
Hình 3.12 Test board thử nghiệm chương trình .....	54
Hình 3.13 Chương trình thử nghiệm kết nối phân cứng .....	55
Hình 3.14 File 420ma.txt trong thẻ nhớ .....	56
Hình 3.15 Sử dụng phần mềm WinSCP truy cập FTP Server .....	57
Hình 3.16 Chức năng tải file từ Server.....	58
Hình 3.17 Giao diện chức năng OTA Update .....	58
Hình 3.18 Chức năng hiển thị giá trị PV .....	59

Hình 3.19 Sản phẩm hoàn thiện .....	60
Hình 3.20 Kết nối Module với BCS2 .....	61
Bảng 2.1 Công suất tiêu thụ năng lượng trên ESP32 .....	32
Bảng 2.2 Công suất tiêu thụ của LAN8720.....	33
Bảng 2.3 Công suất tiêu thụ trên DS3231 .....	33
Bảng 2.4 Bảng nối dây ESP32 .....	39



## DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
SV	Desired value
PV	Process variable
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
UART	universal asynchronous receiver-transmitter
TCP	Transmission Control Protocol
HTTP	Hypertext Transfer Protocol
FTP	File Transfer Protocol
LCD	Liquid Crystal Display
RTS	Real-time system

## MỞ ĐẦU

Công nghệ đã trở thành một phần không thể thiếu trong cuộc sống hiện đại. Internet of Thing, hiện đã và đang phát triển mạnh mẽ trong những năm gần đây, nó đơn giản là một hệ thống bao gồm tất cả mọi thứ xung quanh chúng ta được kết nối với nhau thông qua Internet. Xu hướng này làm cho mọi thứ trở nên dễ dàng hơn bao giờ hết, khi mà sự tương tác giữa con người và thiết bị được tối ưu hóa nhất có thể. Điều khiển, giám sát thiết bị từ xa là một trong những ứng dụng phổ biến nhất của xu hướng này.

Với mục đích tiếp cận xu hướng trên và mong muốn nắm bắt những công nghệ mới nên nhóm chúng tôi quyết định thực hiện đề tài “*Thiết kế tích hợp module truyền thông IoT công nghiệp cho thiết bị Shinko WCL-13A*”.

Đề tài này sẽ là một hệ thống hoàn thiện bao gồm: phần cứng, phần mềm, có thể đáp ứng được yêu cầu của doanh nghiệp và ứng dụng sản phẩm trong thực tế.

Mục đích thực hiện đề tài: tổng hợp và hệ thống hóa lại những kiến thức đã học, đồng thời kết hợp với thực tế để nâng cao kiến thức chuyên môn. Làm quen với quy trình vận hành của doanh nghiệp, các bước để thiết kế một sản phẩm từ khâu lên ý tưởng, thiết kế phần cứng, thiết kế phần mềm, thử nghiệm sản phẩm.

Nội dung chính của đề tài:

- Tìm hiểu thiết bị điều khiển nhiệt độ WCL-13A, cách đọc thông số và cấu hình thiết bị qua giao thức truyền thông Modbus-RTU.
- Thiết kế phần cứng Module truyền thông cho thiết bị WCL-13A.
- Xây dựng Web Server nhúng trên Module cho phép cài đặt thiết bị WCL-13A.
- Chạy thử thiết bị và đánh giá kết quả.

## CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

### 1.1 Phân tích đề tài

#### 1.1.1. Số liệu và dữ liệu ban đầu

Đề tài: “Thiết kế tích hợp module truyền thông IoT công nghiệp cho thiết bị Shinko BCS2”

- Điện áp ngõ vào 220VAC
- Kết nối Modbus-RTU thiết bị BCS2
- Chức năng log dữ liệu 4-20mA tại chỗ vào thẻ nhớ qua file 420mA \*.txt
  - o Thời gian lưu dữ liệu vào file txt cài đặt được 1 phút, 5 phút, 10 phút, 15 phút hoặc 20 phút/lần.
- Giao tiếp truyền nhận dữ liệu qua LAN/WIFI/4G (Dùng IC, Chip)
- Cấu hình thông số thiết bị qua webserver
- Chức năng gửi file dữ liệu 4-20mA \*.txt qua giao thức FTP
  - o Truyền dữ liệu theo phương thức FTP tới địa chỉ máy chủ FTP bằng tài khoản và địa chỉ FTP do Shinko cung cấp
  - o Dữ liệu được truyền về VPS có khoảng thời gian truyền cài đặt được 1 phút, 5 phút, 10,15 phút hoặc 20 phút/lần.
  - o Cấu hình 2 tài khoản FTP
- Đọc File \*.txt từ VPS

#### 1.1.2. Yêu cầu công nghệ

- Nguồn cung cấp: Điện áp ngõ vào là điện áp xoay chiều 220VAC/50~60Hz. Điện áp phổ biến sử dụng cho các thiết bị điện tử là điện áp một chiều, cần sử dụng bộ chuyển đổi điện áp để chuyển dòng xoay chiều thành một chiều với mức điện áp phù hợp. Đáp ứng được các yêu cầu: nguồn chống có độ bền cao, ổn định, chống nhiễu tốt để đảm bảo các thiết bị hoạt động lâu dài
- Kết nối Modbus-RTU: Chuẩn truyền thông Modbus là một chuẩn truyền thông công nghiệp với các ưu điểm như độ tin cậy cao, khả năng truyền xa, băng thông thấp. Để sử dụng được truyền thông ta cần sử dụng lớp vật lý rs485 và IC chuyên dụng để mã hoá/giải mã gói tin trong truyền thông giữa BCS2 và bộ xử lý trung tâm.
- Giao tiếp truyền nhận dữ liệu qua LAN/WIFI: Lựa chọn vi điều khiển bộ xử lý trung tâm hỗ trợ các chuẩn giao tiếp của mạng LAN, WIFI.

- Kết nối 4G: Sử dụng module SIM hỗ trợ mạng 4G và 3G để đảm bảo tín hiệu ổn định, dự phòng trường hợp mất kết nối Internet.

## **1.2 Nội dung nghiên cứu**

### **1.2.1. Tổng quan thiết kế**

*Nguồn cung cấp:*

Phương án 1: sử dụng nguồn tuyến tính

Ưu điểm: cấu tạo đơn giản, chi phí thấp, độ gợn sóng điện áp đầu ra nhỏ, ít ảnh hưởng bởi nhiễu

Nhược điểm: Hiệu suất chuyển đổi không cao, toả nhiệt làm giảm tuổi thọ của thiết bị.

Nguồn tuyến tính phù hợp với các sản phẩm nhỏ, yêu cầu độ ổn định của nguồn như trong các thiết bị đo lường.

Phương án 2: Sử dụng nguồn xung

Ưu điểm: hiệu suất cao, ít tỏa nhiệt và kích thước nhỏ hơn nhiều so với nguồn tuyến tính có cùng công suất.

Nhược điểm: Độ gợn sóng điện áp đầu ra không ổn định như nguồn tuyến tính, cấu tạo linh kiện phức tạp hơn.

=> Giải pháp: Sử dụng nguồn xung để chuyển đổi điện áp xoay chiều thành điện áp một chiều, đảm bảo hiệu suất chuyển đổi, tiết kiệm năng lượng. Kết hợp các IC nguồn tuyến tính như LM7805, AMS1117 để tạo ra các cấp điện áp phù hợp với từng thiết bị ngoại vi.

*Kết nối Internet:*

Sử dụng mạng Ethernet, cáp quang tốc độ 10/100BASE-TX (Tốc độ từ 10 đến 100Mb/s), kết nối ổn định, tốc độ truyền cao.

Bên cạnh đó, Xây dựng thêm các phương án dự phòng cho việc nối mạng Internet khi không có cáp quang như kết nối qua WIFI và 4G.

#### **1.2.1.1. Lưu trữ dữ liệu**

Có rất nhiều giải pháp lưu trữ dữ liệu như EEPROM, bộ nhớ flash, HDD, SSD, SD Card.

Đối với đề tài này nhóm lựa chọn giải pháp lưu trữ dữ liệu sử dụng thẻ nhớ SD Card vì thẻ nhớ SD Card có chuẩn giao tiếp đơn giản, bộ nhớ lớn, kích thước nhỏ gọn, dễ dàng thay thế.

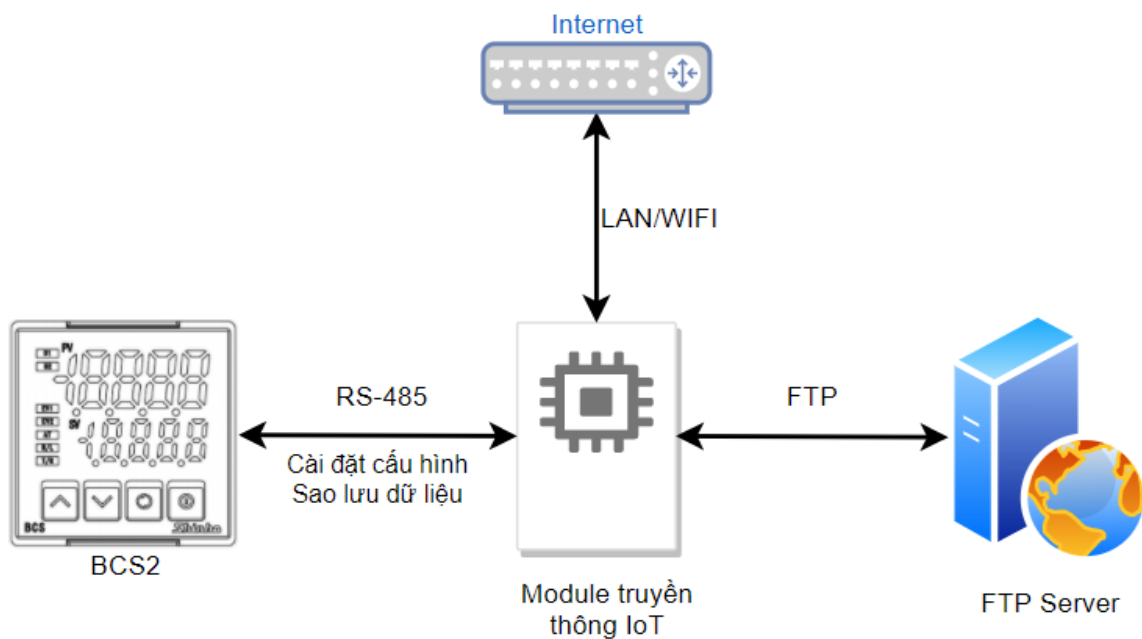
#### 1.2.1.2. Bộ xử lý trung tâm

Bộ xử lý trung tâm sử dụng vi điều khiển hỗ trợ nhiều kết nối ngoại vi. Tương thích với các chuẩn truyền thông Ethernet và có thể hoạt động như một Web server.

#### 1.2.2. Sơ đồ khối tổng quan

Trong quá trình thực hiện đề tài, nhóm sử dụng thiết bị điều khiển nhiệt độ Shinko BCS2 thay thế cho thiết bị Shinko WCL-13A, hai thiết bị có chức năng tương đương nhau.

*Ý tưởng thiết kế:* Từ những yêu cầu đề tài từ công ty và quá trình tìm hiểu đề tài. Nhóm đưa ra ý tưởng thiết kế Module truyền thông IoT kết nối với thiết bị BCS2 theo giao thức truyền thông MODBUS-RTU, chuẩn giao tiếp vật lý RS485 để cấu hình thiết bị BCS2. Module sẽ kết nối đến mạng nội bộ thông qua dây cáp mạng LAN RJ45 hoặc kết nối router wifi, từ đó các thiết bị trong cùng mạng LAN có thể truy cập vào Web Server nhúng trên thiết bị và cấu hình thông qua giao diện Web. Module IoT có các chức năng chuyển đổi truyền thông MODBUS-RTU, đọc/ghi dữ liệu với thiết bị BCS2, chức năng sao lưu dữ liệu vào thẻ nhớ và gửi dữ liệu sao lưu lên Server.



Hình 1.1 Mô hình tổng quan

### 1.3 Kế hoạch thực hiện

#### 1.3.1. Phân chia nhiệm vụ

Nhóm gồm 2 Thành viên:

Thành viên 1: Lê Hoài Nam

MSSV: 105160185

thiết kế phần mềm, tìm hiểu về các giao thức truyền thông, kết nối internet, xây dựng web server và viết chương trình.

Thành viên 2: Nguyễn Đình Dũng                      MSSV: 1051

Đảm nhận nhiệm vụ thiết kế phần cứng, tính toán công suất, lựa chọn nguồn, thiết kế mạch.

Giai đoạn 1 (31/8 -> 15/9):: Lên ý tưởng

- Task 1: Phân tích đề tài, lựa chọn giải pháp, Xây dựng sơ đồ tổng quan
- Task 2: Lựa chọn linh kiện

Giai đoạn 2 (15/9 -> 15/10):: Thiết kế sản phẩm dạng prototype (module)

- Task 1: Lựa chọn linh kiện, xây dựng sản phẩm prototype.
- Task 2: Thử nghiệm sản phẩm, viết chương trình test.

Giai đoạn 3 (15/10 -> 15/12):: Hoàn thiện prototype

- Task 1: hoàn thiện mạch prototype, layout mạch công nghiệp
- Task 2: Hoàn thiện chương trình

Giai đoạn 4 (1/12 -> 23/12): Hoàn thiện sản phẩm, báo cáo hoàn chỉnh

### 1.3.2. Dòng thời gian

Tuần/ngày	Nội dung
Tuần 1	Họp cùng giáo viên hướng dẫn và cán bộ hướng dẫn tại doanh nghiệp để trao đổi về đề tài tốt nghiệp
Tuần 2	Tìm hiểu các kiến thức liên quan
Tuần 3	Xây dựng giải pháp cho đề tài
Tuần 4	Báo cáo lại giải pháp, thống nhất phương án thiết kế
Tuần 5	Lên kế hoạch, chia giai đoạn và mục tiêu cần hoàn thành
Tuần 6	Lựa chọn thiết bị ngoại vi, xây dựng bomlist, báo cáo kết quả cho giáo viên hướng dẫn và cán bộ hướng dẫn tại doanh nghiệp
Tuần 7,8	Đặt mua linh kiện viết chương trình thử nghiệm phần cứng

Tuần 8	Chạy thử chương trình
Tuần 9,10	Tính toán thiết kế mạch Tìm hiểu lập trình web
Tuần 11,12	Thiết kế mạch Viết chương trình chính
Tuần 13	Thiết kế mạch Thiết kế web
Tuần 14	Thi công mô hình Chạy thử sản phẩm
Tuần 15,16	Hoàn thiện mạch, kiểm tra hoạt động của hệ thống Viết báo cáo
Tuần 17	Hoàn thiện báo cáo và mô hình sản phẩm

-Mỗi tuần làm việc, nghiên cứu tại công ty Shinko Technos các ngày thứ 2,4,6 theo giờ hành chính.

- Báo cáo tiến độ cho giáo viên hướng dẫn và cán bộ hướng dẫn doanh nghiệp theo mỗi giai đoạn.

## CHƯƠNG 2: THIẾT KẾ PHẦN CỨNG

### 2.1 Tìm hiểu thiết bị điều khiển nhiệt độ BCS2

Trong quá trình thực tập và thử nghiệm sản phẩm nhóm sử dụng thiết bị điều khiển nhiệt độ BCS2. BCS2 là dòng sản phẩm điều khiển nhiệt độ có chức năng tương tự WCL-13A và được sự cho phép của công ty Shinko sử dụng BCS2 làm đối tượng nghiên cứu thay thế tương đương trong đề tài.



Hình 2.1 Dòng sản phẩm BCx2

BCS2 là dòng điều khiển nhiệt độ thế hệ mới, kế thừa và hướng tới nâng cao các tính năng hiện có của dòng JC như tính năng thao tác, tính năng quan sát, tính năng điều khiển, giảm thiểu không gian lắp đặt. Nhóm đã được tìm hiểu và thao tác với sản phẩm BCS2 trong quá trình thực tập tại công ty.

#### Các tính năng của sản phẩm:

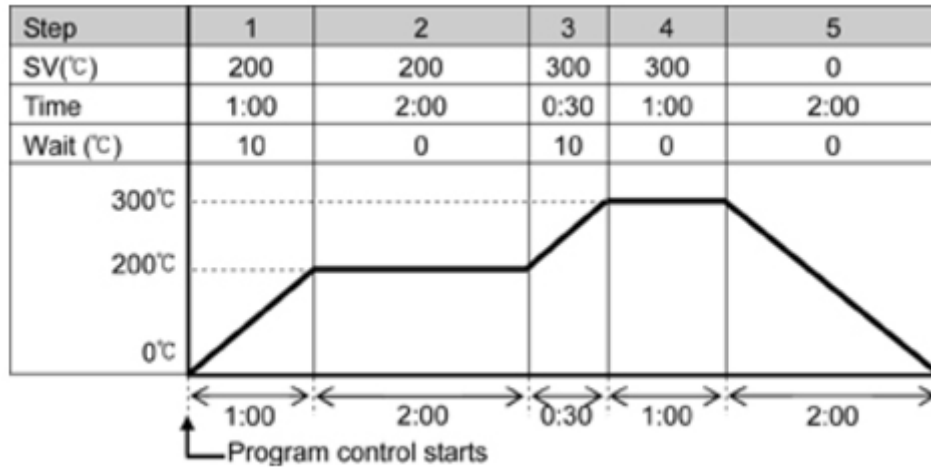
- *Giao thức cài đặt ban đầu thuận tiện*
- *Chức Năng Tự Động Dò Các Tham Số Điều Khiển Qua Đặc Tính Khởi Động Của Đối Tượng (At On Startup)*

Chức năng này sẽ tính toán các tham số PID khi chức năng AT không thể thực hiện được do sự biến động của nhiệt độ.

- *Chức Năng Điều Khiển Chương Trình Đơn Giản*

Chức năng điều khiển theo chương trình với 9 bước nhiệt.





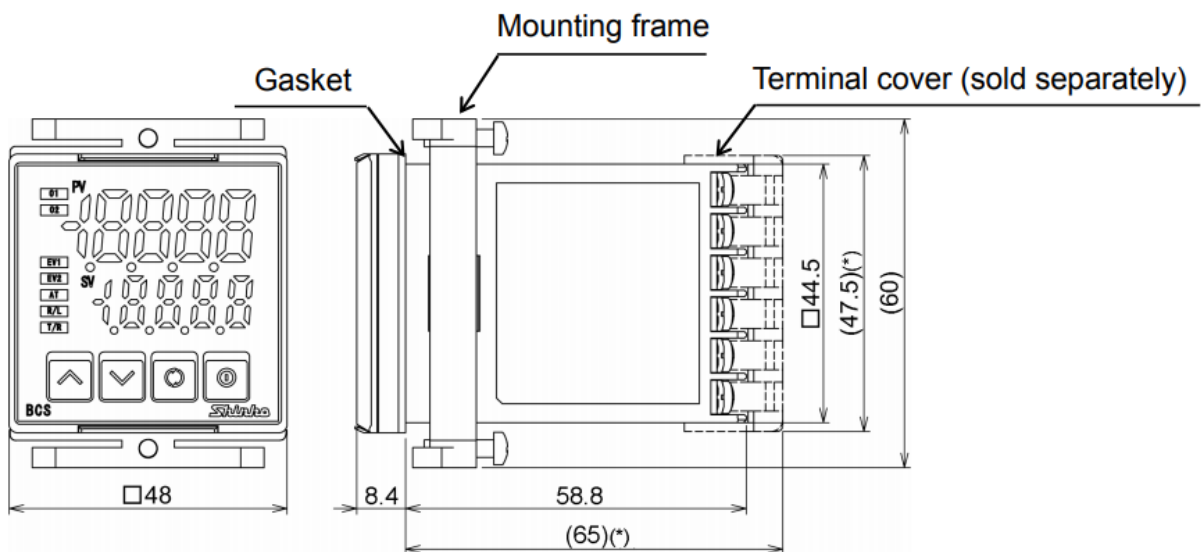
Hình 2.2 Ví dụ về điều khiển chương trình theo nhiệt

- Chức Năng Đầu Vào/Ra Event

Có tối đa 2 điểm đầu vào Event (tùy chọn) và 2 điểm đầu ra Event ( 1 điểm là tiêu chuẩn). Có thể lựa chọn từ nhiều lựa chọn đầu vào/ra : Đầu ra cảnh báo, đầu ra cảnh báo đứt dây nóng (tùy chọn), đầu ra cảnh báo Loop break, đầu ra tín hiệu thời gian, đầu ra kết thúc chương trình nhiệt.v.v...

- Màn Hình Hiển Thị Pv, Sv 5 Chữ Số, Lớn, Dễ Quan Sát
- Kích Thước Nhỏ Gọn

Với chiều sâu lắp đặt trong bảng điều khiển chỉ 60 mm, dòng BCS2 lý tưởng cho việc lắp đặt trong tất cả các bảng điều khiển



Hình 2.3 Kích thước bảng điều khiển của BCS2

- Thiết Lập Chức Năng Đơn Giản Từ Máy Tính

Kết nối BCS2 với máy tính bằng cáp chuyên dụng (bán riêng), sử dụng phần mềm chuyên dụng để thiết lập các chức năng sản phẩm từ máy tính.

Không cần thiết cung cấp nguồn điện cho BCS2 khi kết nối với máy tính, nguồn điện BCx2 được cung cấp bởi máy tính thông qua cổng kết nối USB.



Hình 2.4 Sơ đồ kết nối BCS2 và PC

Phần mềm giám sát dòng BCS2 SWC-BCx01M (phần mềm chuyên dụng từ trang web SHINKO TECHNOS R&D.)

- *Đáp Ứng Tiêu Chuẩn Chống Nước/Bụi Ip66*

Đáp ứng tiêu chuẩn IP66 (đối với mặt trước) cho phép sử dụng sản phẩm trong môi trường khắc nghiệt phải tiếp xúc với bụi và nước.

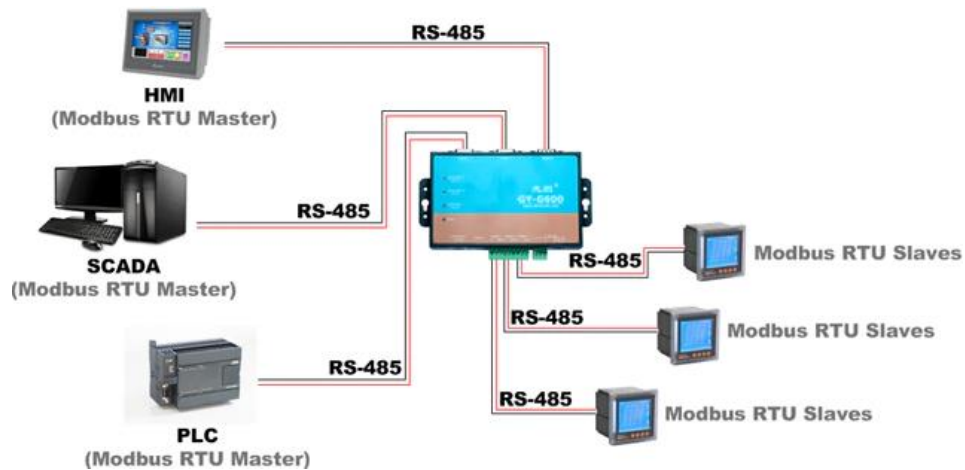
## 2.2 Tìm hiểu công nghệ

### 2.2.1. Modbus

#### 2.2.1.1. Định nghĩa

Mod bus là một giao thức truyền thông nối tiếp được sử dụng phổ biến trong công nghiệp với các ưu điểm dễ triển khai, bảo trì độ ổn định cao và ít nhiễu.

Modbus RTU hoạt động dựa trên nguyên lý Master – Slave, bên đóng vai trò nhận (Master) và một bên đóng vai trò truyền tín hiệu (Slave) thông qua địa chỉ thanh ghi. Các phương thức truyền của giao thức Modbus RTU bằng đường truyền vật lý RS232, Modbus RTU RS485.

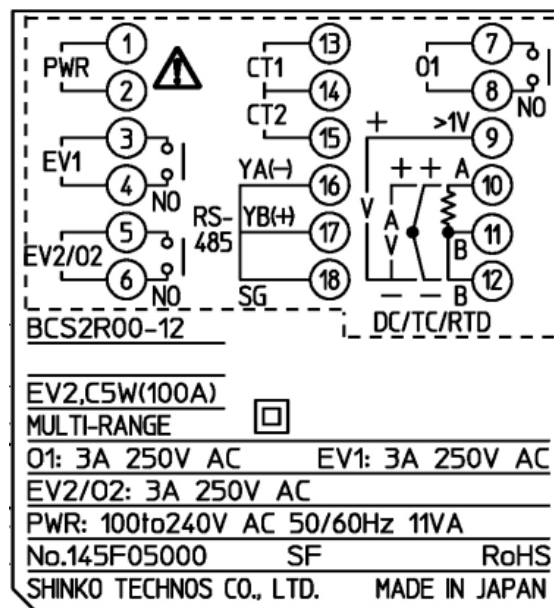


Hình 2.5 Mô hình Modbus-RTU theo nguyên lý master-slave

### 2.2.1.2. Kết nối phân cứng

Bộ điều khiển nhiệt độ BCS2 sử dụng đường truyền vật lý với 3 dây YA(-), YB(+) và chân SG là chân nối chung mass giữa các thiết bị.

Tác dụng của chân SG là làm điện áp tham chiếu cho hiện điện áp giữa hai dây YA và YB, giảm sai số điện áp giữa hai đầu thiết bị.



Hình 2.6 Sơ đồ nối dây BCS2

- Khung truyền: Dữ liệu nhị phân được truyền mỗi 8 bit.
  - Định dạng dữ liệu:
  - Start bit: 1 bit
  - Data bit: 8 bits
  - Parity: No parity (Chẵn, lẻ)

- Stop bit: 1 bit (2 bits)
- Error detection: CRC-16.

Để thuận tiện cho người sử dụng, nhóm sử dụng khung truyền mặc định của thiết bị cho module truyền thông IoT.

Cấu trúc khung truyền:

3.5 idle characters	Slave address	Function code	Data	Error check CRC-16	3.5 idle characters
---------------------	---------------	---------------	------	--------------------	---------------------

Hình 2.7 Cấu trúc khung truyền Modbus-RTU

- Slave address (1 bit): Được đặt riêng cho mỗi thiết bị trong mạng, các thiết bị được đánh số từ 0 đến 95, tuy nhiên địa chỉ 0 slave sẽ không phản hồi ( quy chuẩn của Modbus RTU) nên địa chỉ của thiết bị sẽ đánh số từ 1 đến 95 (01H đến 5FH).
- Function Code (1 bit): BCS2 cung cấp nhiều mã function code để thực hiện từng nhiệm vụ riêng biệt như đọc/ghi dữ liệu, đọc thông tin nhận dạng thiết bị.

Trong phạm vi đề tài: nhóm sử dụng 2 mã function chính là đọc/ghi dữ liệu

- 03(03H): Đọc dữ liệu từ thiết bị.
- 06(06H): Ghi giá trị vào thiết bị.
- Data (8bit): là dữ liệu cần cung cấp tùy theo function code.

Ví dụ lệnh Ghi giá trị SV=600°:

3.5 idle characters	Slave address (01H)	Function code (06H)	Data item (0001H)	Data (0258H)	Error check CRC-16 (D890H)	3.5 idle characters
	1	1	2	2	2	

Hình 2.8 Ví dụ lệnh ghi giá trị cho SV

Giá trị muốn cài đặt SV chuyển đổi qua mã thập lục phân và đưa vào vị trí tương ứng trong khung truyền ( 600 = 0258H)

Error check CRC-16: mã CRC-16 được tính theo giá trị của data phía trước nó. Mỗi giá trị data khác nhau sẽ cho ra mã CRC-16 khác nhau nhằm đảm bảo độ tin cậy của gói tin, tránh sai sót trong quá trình truyền nhận.

Cách tính mã CRC-16:

1. Khởi tạo dữ liệu CRC-16 (giả sử là X) (FFFFH).
2. Tính OR (XOR) của X(FFFFH) với byte đầu tiên của data . Gán giá trị mới cho X.
3. Dịch 1 bit của X sang phải. Gán giá trị mới cho X.

4. Nếu bit có trọng số thấp của X (LSB), thực hiện phép XOR X và giá trị cố định (A001H). Gán giá trị mới cho X. Nếu LSB chưa có giá trị 1, chuyển sang bước 5.
5. Lặp lại các bước 3 và 4 cho đến khi dịch phải bit của X được 8 lần.
6. Tính XOR giá trị của X với bit data tiếp theo, gán giá trị tính được cho X
7. Lặp lại các bước từ 3 đến 5.
8. Lặp lại các bước từ 3 đến 5 cho đến bit data cuối cùng.
9. Giá trị X tính được là mã CRC-16 của data.

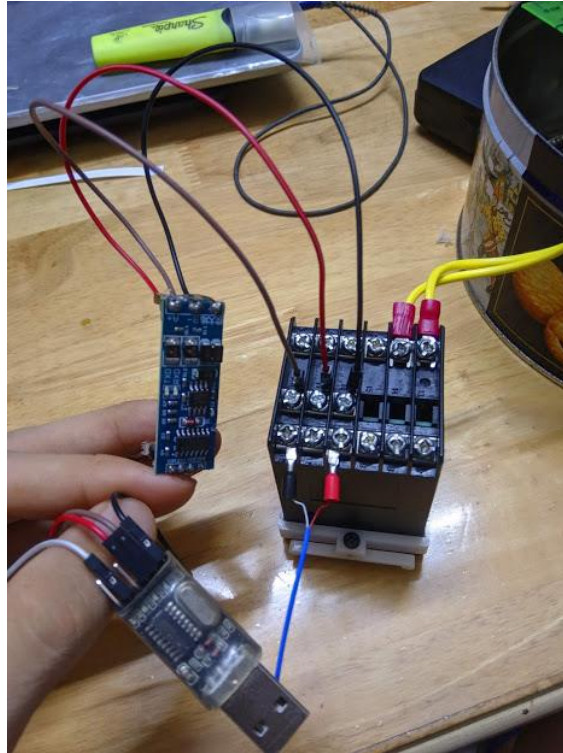
Các bước tính toán được thể hiện dưới dạng chương trình C:

```
//Calc the raw_msg_data_byte CRC code
uint16_t crc = 0xFFFF;
String crc_string = "";
for (int pos = 0; pos < raw_msg_data.length() / 2; pos++) {
    crc ^= (uint16_t)raw_msg_data_byte[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)

//Become crc byte to a capital letter String
crc_string = String(crc, HEX);
```

*Hình 2.9 Chương trình tính mã CRC-16.*

Thử nghiệm thực tế bằng USB chuyển đổi USB-TTL và module TTL-UART RS485:



Hình 2.10 Truyền thông modbus với thiết bị BCS2

Sử dụng phần mềm Hercules để thử nghiệm truyền nhận gói tin giữa thiết bị BCS2 và máy tính.



Hình 2.11 Phần mềm Hercules

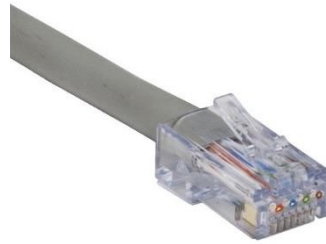
Kết quả đạt được:

- 2 thiết bị phải cấu hình đồng bộ về baudrate, độ dài data bit, parity để có thể truyền thông với nhau.
- Mã CRC-16 của mỗi gói tin phải tính chính xác để BCS2 tiếp nhận gói tin.
- Gửi gói tin đúng theo cấu trúc của Modbus và đúng function được cấu hình trong thiết bị BCS2, thiết bị sẽ gửi gói tin phản hồi.
- Gói tin gửi và nhận đều được gửi dưới dạng mã thập lục phân (HEX).
- Gửi gói tin sai cú pháp không có phản hồi.

### 2.2.2. Công nghệ Internet

**LAN** là viết tắt của Local Area Network tạm dịch là mạng máy tính nội bộ, giao tiếp này cho phép các máy tính kết nối với nhau để cùng làm việc và chia sẻ dữ liệu. Kết nối này được thực hiện thông qua sợi cáp LAN hoặc Wifi (không dây) trong không gian cố định.

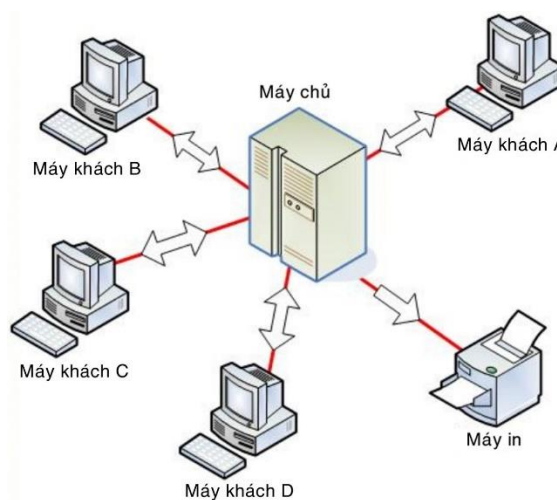
Cổng kết nối mạng LAN phổ biến hiện nay là cổng RJ45, được sử dụng phổ biến trên hầu hết các thiết bị từ công nghiệp đến dân dụng



Hình 2.12 Cổng mạng LAN RJ45

**Wifi** (Wireless Fidelity) là thuật ngữ dùng chung để chỉ tiêu chuẩn IEEE802.11 cho mạng cục bộ không dây (Wireless Local Networks) hoặc WLANs. Dữ liệu được truyền qua sóng vô tuyến cho phép các thiết bị truyền nhận dữ liệu ở tốc độ cao trong phạm vi của mạng Wifi.

Để tạo được mạng LAN nội bộ thì cần có một thiết bị làm máy chủ (server), một số thiết bị hỗ trợ kết nối và cuối cùng là các máy khách. Trước khi tạo mạng LAN cần chắc chắn rằng tất cả các thiết bị đều được tích hợp sẵn card mạng NIC (Network Interface Card).



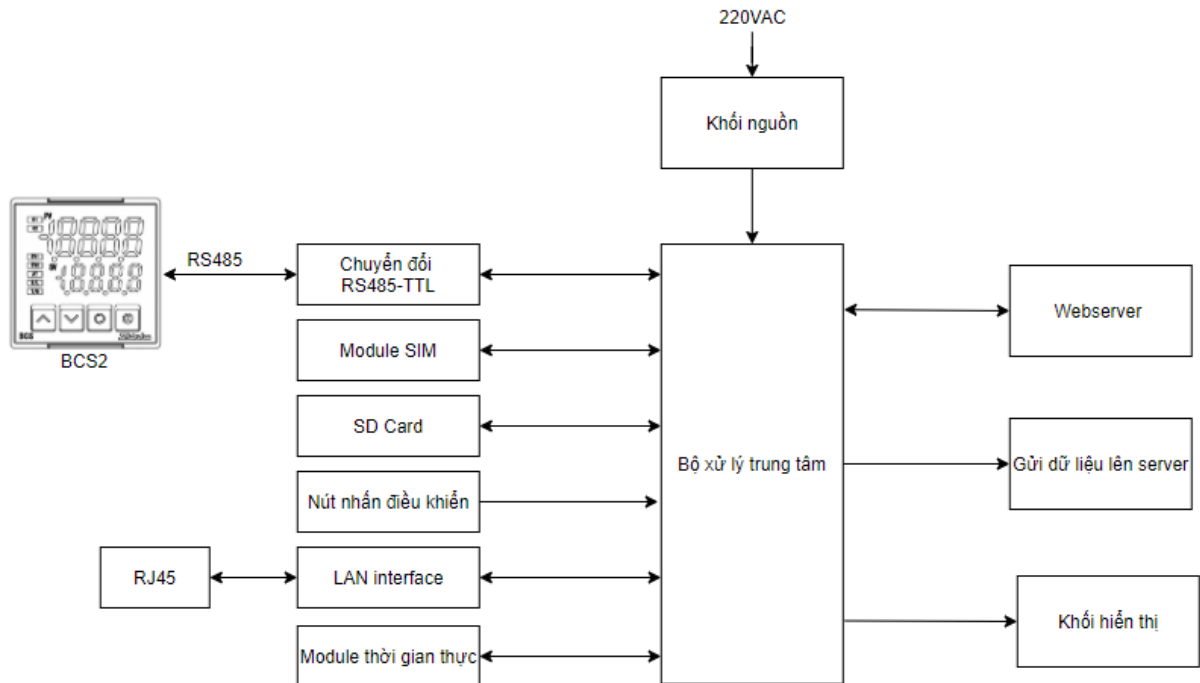
Hình 2.13 Mô hình mạng LAN.



Như vậy để một thiết bị có thể hoạt động như một web server thiết bị đó cần hỗ trợ kết nối internet Bộ xử lý trung tâm cần hỗ trợ kết nối wifi và kết nối mạng LAN.

### 2.3 Phương án thiết kế

Sơ đồ khối tổng quan của hệ thống sau quá trình tìm hiểu:



Hình 2.14 Sơ đồ khối tổng quan

#### 2.3.1. Bộ xử lý trung tâm

Lựa chọn vi điều khiển ESP32 làm bộ xử lý trung tâm vì ESP32 được tích hợp công nghệ Wifi, sử dụng giao thức 802.11 b/g/n và tần số 2.4GHz, dải tần số được sử dụng phổ biến cho mạng wifi hiện nay. Đồng thời hỗ trợ chuẩn RMII để giao tiếp với IC PHY, lớp vật lý hỗ trợ chuẩn IEEE 802.3/802.3u 10/100 Mbps.

Một số thông số cơ bản của ESP32:

- CPU: Xtensa Dual-Core LX6 microprocessor.
- Chạy hệ 32 bit
- Tốc độ xử lý: 160MHZ up to 240 MHz
- Tốc độ xung nhịp đọc flash chip: 40mhz --> 80mhz (tùy chỉnh khi lập trình)
- RAM: 520 KByte SRAM
- Số cổng GPIOs : 34
- Cập nhập firmware từ xa (OTA)

Giao tiếp không dây:

- Wi-Fi: 802.11 b/g/n/e/i



- Bluetooth: v4.2 BR/EDR and BLE

Hỗ trợ nhiều loại giao tiếp:

- 8-bit DACs( digital to analog): 2 cổng
- Analog(ADC) 12-bit: 16 cổng.
- I2C: 2 cổng
- UART: 3 cổng
- SPI: 3 cổng
- I2S: 2 cổng
- SD card /SDIO/MMC host
- Slave (SDIO/SPI)
- Ethernet MAC interface
- CAN bus 2.0
- IR (TX/RX)
- Băm xung PWM (tất cả các chân )

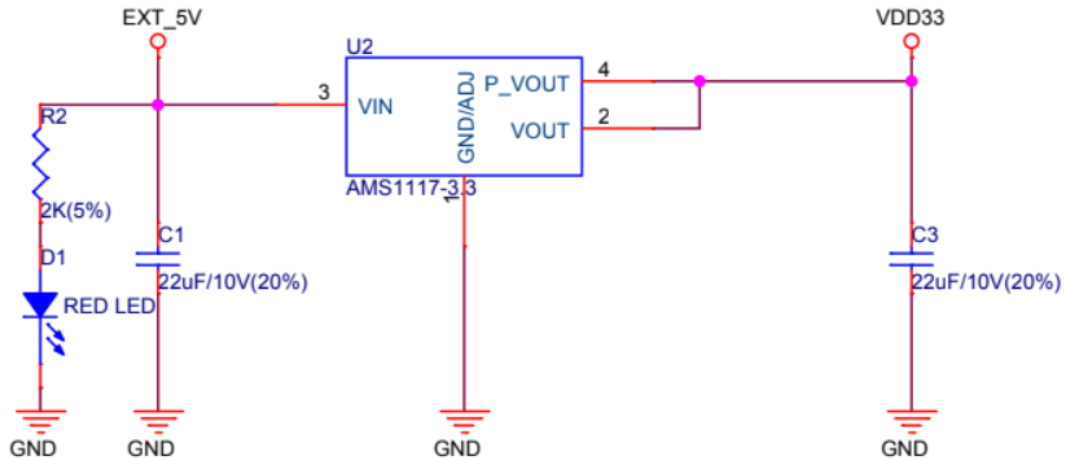
Nguồn điện hoạt động:

- Nhiệt độ hoạt động -40 + 85C
- Điện áp hoạt động: 2.2-3.6V



Hình 2.15 Chip ESP32-WROOM-32

### 2.3.1.1. Tính toán nguồn nuôi cho bộ xử lý trung tâm esp32-wroom32



Hình 2.16 Sơ đồ nguyên lý mạch nguồn nuôi

Dựa trên yêu cầu nguồn nuôi của ESP32 Wroom32 là  $V_{cc} = 3.3V$ . Từ đó chọn ra ASM1117-3.3 là IC ổn áp để cấp nguồn nuôi cho bộ xử lý trung tâm ESP32.

Một số thông số kỹ thuật ASM1117-3.3:

- Điện áp ngõ ra 3.3V
- Dòng tối đa 1A.
- Áp ngõ vào :  $1.5V < V_{in} - V_{out} < 12V$
- Nhiệt độ hoạt động -40 đến 125 độ C
- Dạng chân TO252
- Số chân 4

Lựa chọn linh kiện:

Với điện áp đầu vào là 5VDC, Điện áp ra là 3.3VDC, dòng tải lên tới 0.5A.

Đèn Led D1 là đèn báo nguồn đầu vào:

Chọn  $R2 = 2k$ , khi đó:

$$I_{led} = V_{in}/R = 5V/2k = 0.0025A = 2.5mA$$

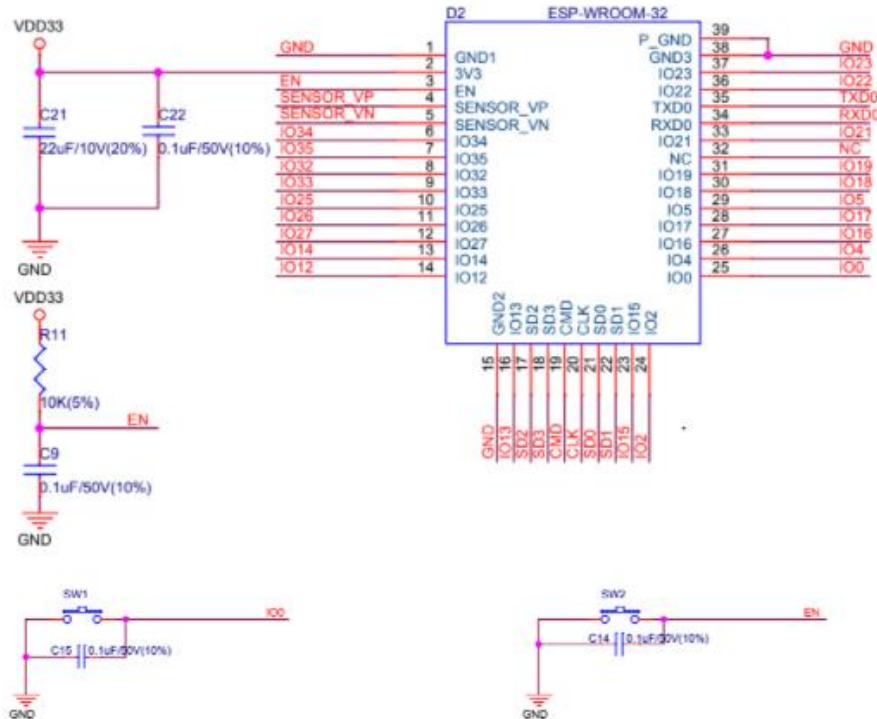
với  $R2$  là điện trở giới hạn dòng đặt lên Led D1.

Trong quá trình đo đạc và tìm hiểu manual của nhà phát hành IC ổn áp ASM1117-3.3 dựa trên các thông số điện áp nhiều đầu ra tiến hành chọn tụ lọc  $C1$  và  $C3$  với giá trị:

$$C1 = C3 = 22\mu F$$

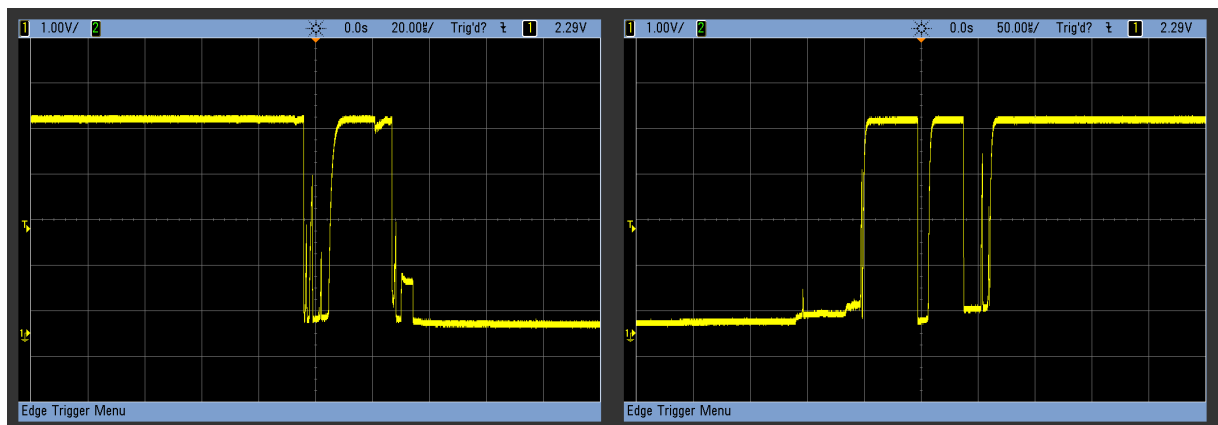
Tụ C1 có tác dụng lọc nhiễu nguồn đầu vào 5V, tụ C3 có tác dụng lọc nhiễu đầu ra đảm bảo sự ổn định của điện áp đầu ra cung cấp cho bộ xử lý trung tâm.

### 2.3.1.2. Tính toán bộ xử lý trung tâm esp32-wroom32



Hình 2.17 Sơ đồ nguyên lý mainboard ESP32

Tụ C9, C15, C14 với các giá trị  $C9 = C15 = C14 = 0.1\mu\text{F}/50\text{V}$  là các tụ có tác dụng chống nhiễu, Debounce cho nút nhấn, giải quyết hiện tượng SWITCH BOUNCE nhiều khi nhấn nút.



Hình 2.18 Switch bounce trên nút nhấn

R11 là điện trở pullup chân ENABLE (EN) có tác dụng hạn chế dòng bảo vệ chân EN. R11 được tính như sau:

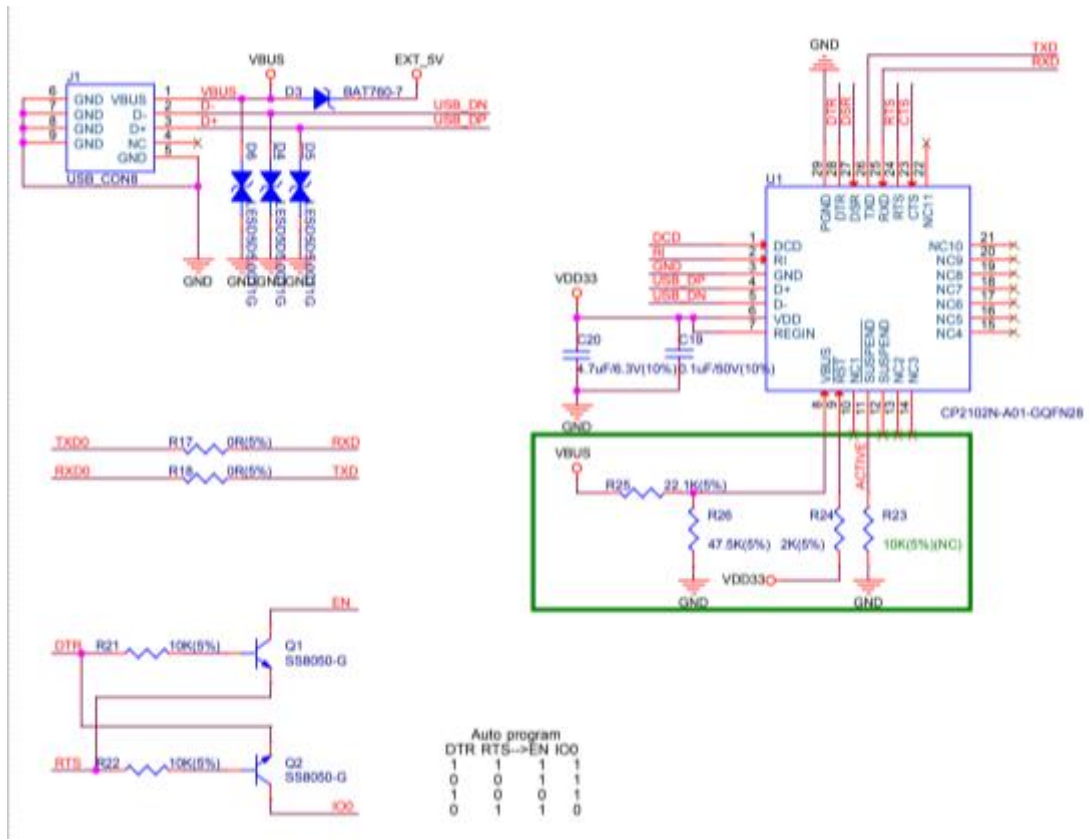
Chọn  $R11 = 10k$ , khi đó:  $I_{en} = V_{dd} / R11 = 3.3V/10K = 0.00033A = 0.33mA$

Với  $I_{en}$  ( dòng đặt lên pin EN) có giá trị 0.33mA Thì R11 đáp ứng được yêu cầu hạn chế dòng bảo vệ được chân EN.Với tụ sau lọc có công thức:

$$C = \frac{(5xI)}{(V_{dc} * f)}$$

Vì C21 mắc song song với C22, giả sử chọn  $C_1 = 22\mu F$ , áp dụng công thức ta tính được  $C_2 = 0.1\mu F$

Tính toán mạch nạp:



Hình 2.19 Sơ đồ nguyên lý mạch nạp

Dựa trên kiến nghị của nhà phát hành về diode ESD protection thì nhóm lựa chọn linh kiện diode LESD5D5.0CT1G cho ESD protection cho bản vẽ với một số thông số kỹ thuật như sau:

- LESD5D5.0CT1G được thiết kế để bảo vệ các thành phần nhạy cảm với điện áp khỏi ESD và điện áp quá độ .
- Khả năng kẹp tuyệt vời, rò rỉ thấp và thời gian phản hồi nhanh, làm cho linh kiện này trở nên lý tưởng cho ESD protection.

R21 và R22 có tác dụng tương tự như R11 được tính toán ở trên, Ở đây R21 R22 đảm nhiệm vai trò hạn chế dòng kích mở transistor có giá trị  $R21 = R22 = 10k \text{ ohm}$ .

Với tụ sau lọc có công thức

$$C = \frac{(5xI)}{(V_{dc} * f)}$$

Vì C19 mắc song song với C20, giả sử chọn  $C_1 = 4.7\mu F$ , áp dụng công thức ta tính được  $C_2 = 0.1\mu F$ .

R25 và R26 là mạch phân áp với giá trị  $V_{bus} = 5V$  điện áp phân áp đầu ra là  $3.3V$  Khi đó ta có:

$$V_{bus}/(R26 + R25) = V_{out}/R26$$

$$5V/(R26 + R25) = 3V/R26$$

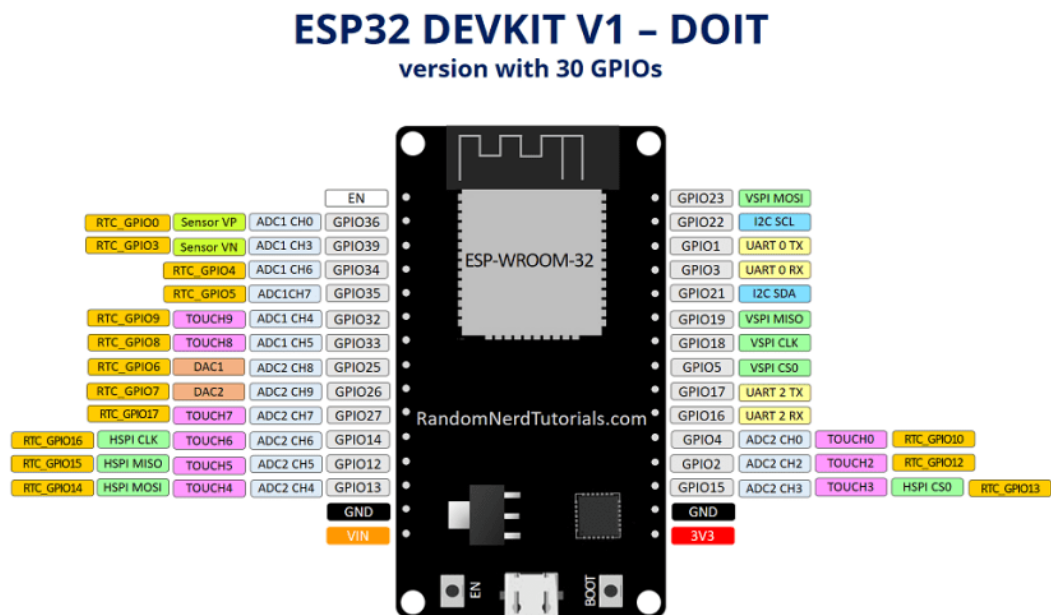
$$\Rightarrow R26 = 2 R25$$

Chọn  $R25 = 22K \Rightarrow R26 = 44k \approx 47k$

### 2.3.1.3. Lựa chọn bộ xử lý trung tâm

Từ các thiết kế và tính toán trên nhóm lựa chọn chip ESP32 DEVKIT V1 DOIT để sử dụng làm prototype cho dự án vì đáp ứng được yêu cầu thiết kế, dễ phát triển ứng dụng.

Module ESP32 DEVKIT V1 – DOIT là module được phát triển dựa trên ESP32 kèm theo ic nguồn, chip nạp chương trình CP2102 thuận lợi cho việc lập trình và thử nghiệm và xây dựng ứng dụng IoT prototype một cách nhanh chóng.



Hình 2.20 Sơ đồ chân ESP32 DEVKIT V1 - DOIT

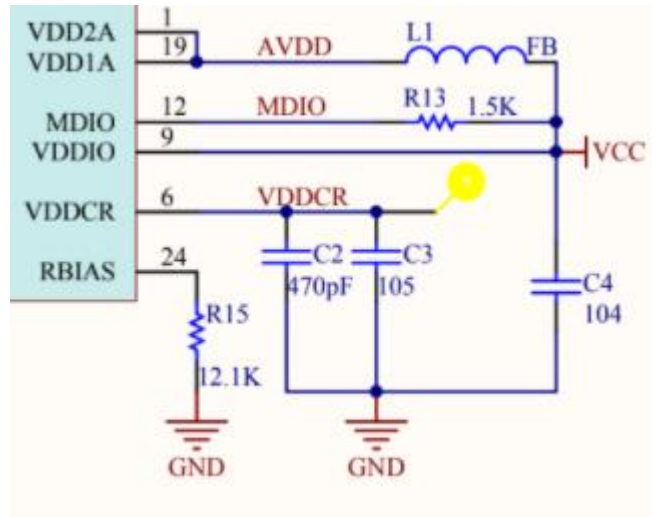


$R6 = R8 = 330 \text{ Ohm}$

Lúc này dòng chạy qua Led sẽ là:

$$I = 3.3V/330\text{Ohm} = 10\text{mA}$$

Đảm bảo được dòng qua Led không quá lớn giúp bảo vệ Led.



Hình 2.23 Nguồn nuôi cho LAN8720

$$V_{in} = V_{cc} = 3.3V$$

Với nguồn trong dự án nhóm sử dụng là Single Power Supply, Nên theo kiến nghị của manual nhà phát hành cần sử dụng một cuộn cảm FB( ferrite bead) Có tác dụng lọc nguồn đầu vào.

Các tụ C2, C3, C4 là các tụ lọc nhiễu nguồn đầu vào dựa trên số liệu đo từ nhà phát hành và trên thực tế, Chọn  $C2 = 470\text{pF}$ ,  $C3 = 0.1\mu\text{F}$ ,  $C4 = 1\mu\text{F}$

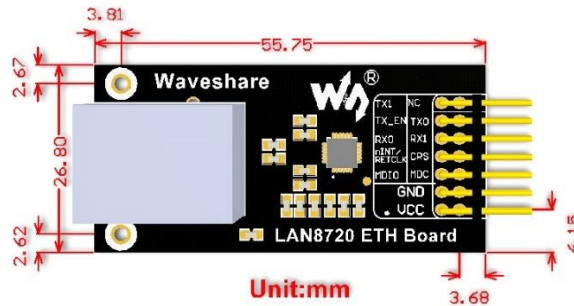
Điện trở pulldown R15 có giá trị  $R15 = 12\text{k}\Omega$  có tác dụng hạn chế dòng chạy qua chân RBIAS.

Các điện trở R1, R2, R3, R4 có tác dụng hạn chế dòng điện đặt lên các chân TXP, TXN, RXP, RXN

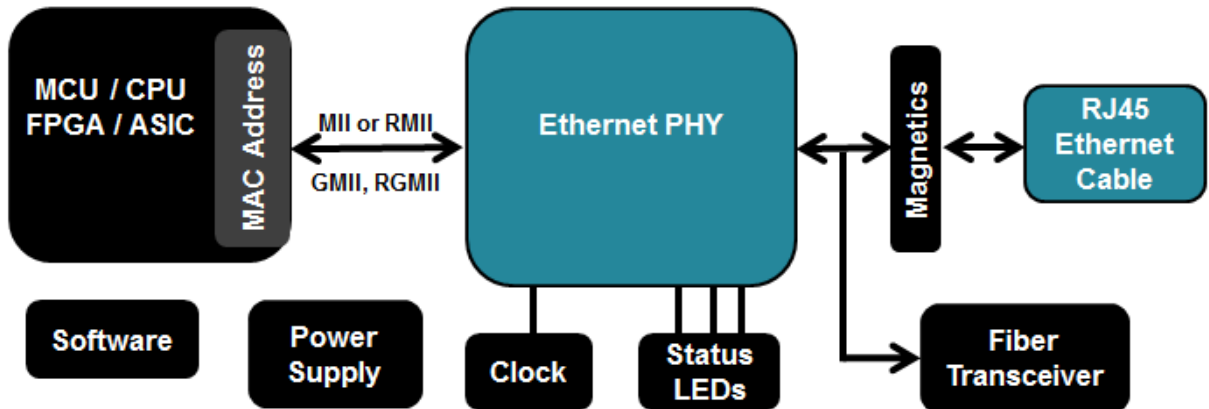
### 2.3.2.2. Module LAN8720

Dựa trên kết quả tính toán lựa chọn module mạng LAN8720 hỗ trợ chuẩn RMII để kết nối thiết bị với mạng ethernet





Hình 2.24 Module LAN8720



Hình 2.25 Cấu trúc liên kết vi điều khiển và module mạng LAN

Thông số cơ bản

- Bộ thu phát lớp vật lý (PHY) tốc độ cao 10/100 Mbps
- Nguồn cung cấp 3.3V
- Hỗ trợ giao diện RMII
- Dải điện áp I/O: +1.6V đến +3.6V
- Nhiệt độ hoạt động 0°C đến 85°C





Hình 2.26 Thử nghiệm kết nối mạng LAN với LAN8720


### 2.3.3. Module 4G

Hiện nay ở nước ta, các nhà mạng đang sử dụng dải băng tần B3 (1800MHz) cho mạng 4G và đến 01/01/2022 sẽ cắt sóng 2G trên toàn quốc. Vì vậy để thiết bị hoạt động lâu dài cần lựa chọn module hỗ trợ mạng 3G/4G sử dụng được ở Việt Nam.

**Legend:** in use partially in use not in use other bands

**Note:** This list of network deployments does not imply any widespread deployment or national coverage.

#### Commercial deployments [\[edit\]](#)

Country or Territory	Operator	VoLTE	B1 2100 MHz	B3 1800 MHz	B5 850 MHz	B7 2600 MHz	B8 900 MHz	B20 800 MHz	B28 700 MHz
 Vietnam	VNPT Vinaphone		15 MHz, UMTS	20 MHz Nov 2016	N/A	-	8,4 MHz, GSM	-	
	Viettel		5 MHz, 2019 10 MHz, UMTS	20 MHz Apr 2017	N/A	-	8,2 MHz, GSM	-	

Hình 2.27 Dải băng tần mạng 4G tại Việt Nam

Sử dụng Module SIM7600 đáp ứng yêu cầu đề tài (hỗ trợ 4G, FTP), module được thiết kế theo tiêu chuẩn công nghiệp, lập trình dễ dàng bằng tập lệnh AT thông qua UART.

Thông tin sản phẩm Module SIM7600:

- Lập trình qua tập lệnh AT
- Dải tần số:
  - TDD-LTE B38/B40
  - FDD-LTE B1/B3/B5/B7/B8/B20
  - GSM/GPRS/EDGE 900/1800 MHz
- Điện áp nguồn: 3.4V ~ 4.2V, Typ: 3.8V
- Nhiệt độ hoạt động: -40°C to +85°C
- Truyền dữ liệu: LTE CAT1: Uplink đến 5Mbps, Downlink 10Mbps
- Hỗ trợ TCP/IP/IPV4/IPV6/MultiPDP/FTP/FTPS/HTTP/HTTPS/DNS



Hình 2.28 Module SIM7600

Tập lệnh AT gửi file lên FTP Server:

AT+CFTPPORT=21 //Cài đặt Port FTP server

AT+CFTPMODE=1

AT+CFTPTYPE=A // File type A, định dạng txt

AT+CFTPSERV="103.97.125.251" //Địa chỉ server

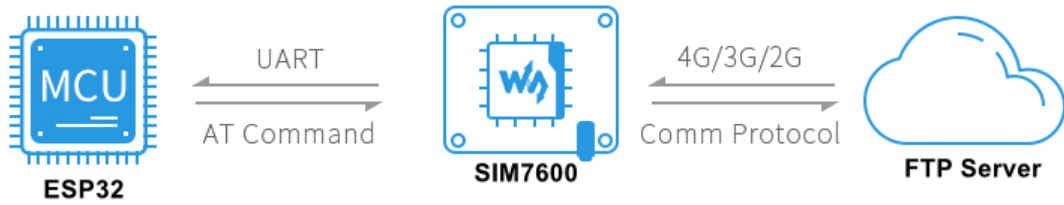
AT+CFTPUN="nguyendinhdung@lhnam.net" //Tài khoản

AT+CFTPPW="nguyendinhdung" //mật khẩu

AT+CFTPPUT="/uploadbysim.txt",0 //Tên file

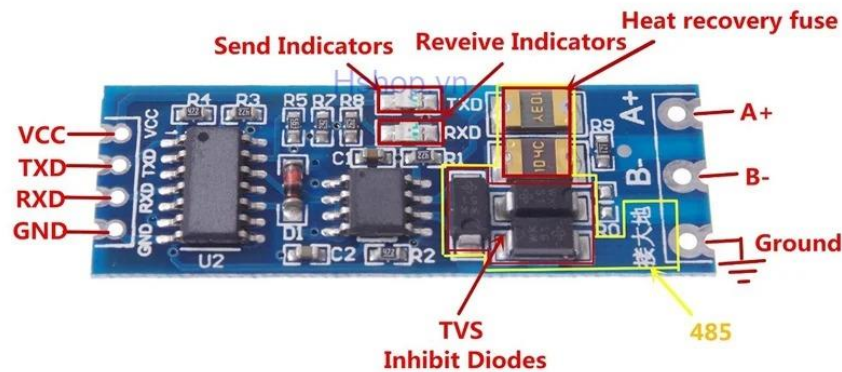
Nội dung file: truyền từ bộ xử lý trung tâm ESP32 đến module SIM qua UART

Gửi kí tự 1A(HEX), kết thúc lệnh tạo file và gửi file lên server đã cấu hình.



Hình 2.29 Mô hình kết nối ESP32, SIM, Server

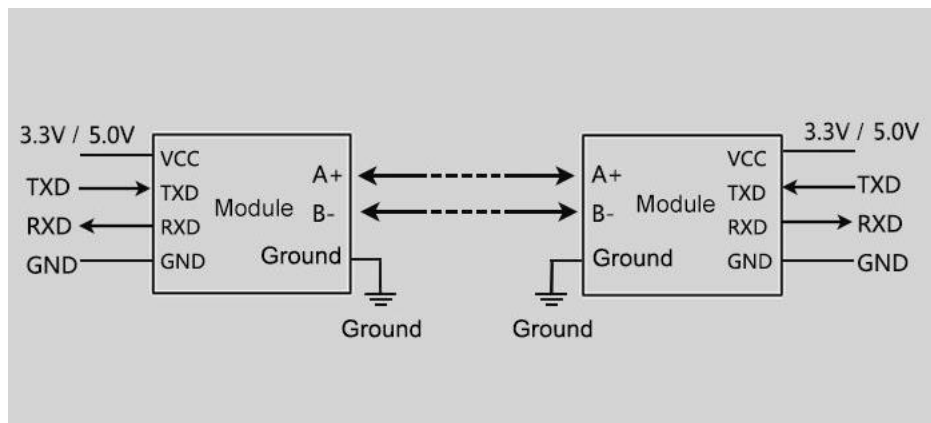
#### 2.3.4. Bộ chuyển đổi RS485-TTL



Hình 2.30 Module chuyển đổi RS485-TTL

Mạch chuyển giao tiếp UART TTL - RS485 được thiết kế để có thể chuyển giao tiếp từ chuẩn giao tiếp UART TTL (Vi điều khiển, máy tính nhúng,...) sang chuẩn giao tiếp RS485 và ngược lại.

Mạch chuyển giao tiếp UART TTL - RS485 được thiết kế với khả năng chống nhiễu cao, tích hợp các bộ đệm, Cầu chì tự phục hồi, Diode chống nhiễu giúp hệ thống chạy ổn định, an toàn hơn và không làm cháy board điều khiển trung tâm. Phần chân giao tiếp RS485 trên mạch có chân Mass, nếu hệ thống có đường dây mass tiếp đất thì có thể sử dụng để nối vào chân Mass này giúp tăng khả năng chống nhiễu và chống sét.



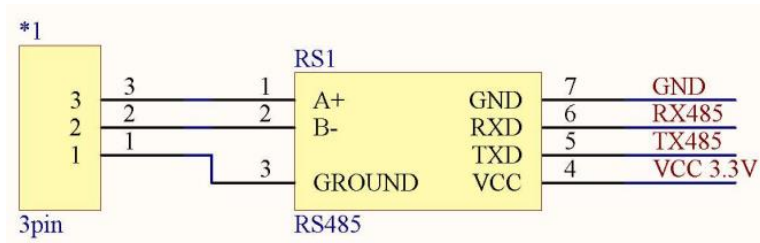
Hình 2.31 Chức năng chuyển đổi mức logic 3.3v-5v

Module sử dụng mạch đệm 74HC14 làm trung gian chuyển mức giao tiếp logic từ 5v sang 3.3V và ngược lại.

Các diode TVS được thiết kế nối đất giúp đạt khả năng chống sét và chống nhiễu.

Thông số kỹ thuật:

- Điện áp hoạt động: 3 - 5VDC.
- Điện áp giao tiếp TTL: 3 - 5VDC.
- Khoảng cách truyền RS485 có thể lên đến 1km, khuyến nghị sử dụng dưới 800m và dây bus chuyên dụng cho RS485.
- Chuẩn chân cắm TTL 2.54mm.
- Có đèn led thông báo trạng thái truyền nhận RX và TX.

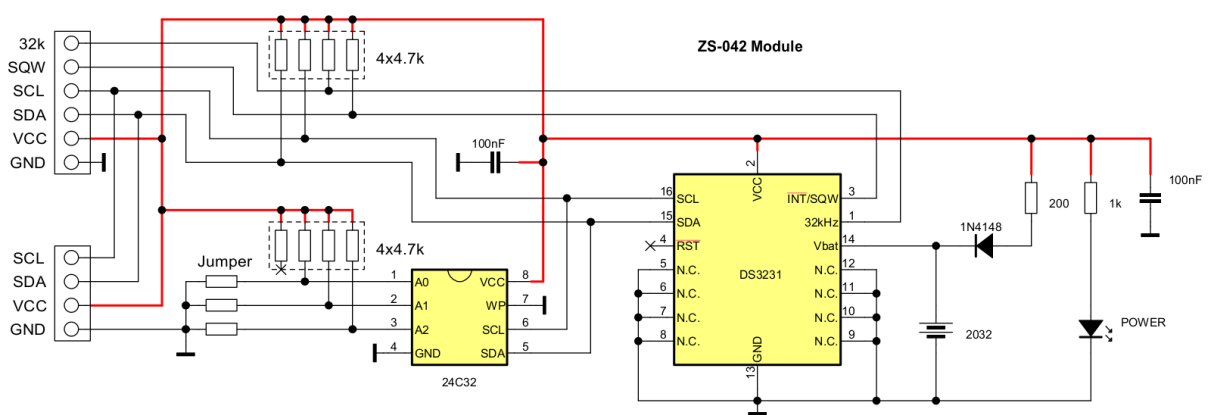


Hình 2.32 Sơ đồ nối dây mạch chuyển giao tiếp TTL-RS485 và ESP32

### 2.3.5. Hệ thống thời gian thực Realtime System

#### 2.3.5.1. Tính toán linh kiện

Sử dụng IC thời gian thực DS3231 để xây dựng hệ thống RTS.



Hình 2.33 Sơ đồ nguyên lý DS3231

Tính toán thiết kế mạch sạc pin DS3231

Với nguồn cấp là  $V_{cc} = 3.3V$

Chọn pin sạc là loại pin 2032, mạch sạc được kết hợp với diode 1N4148 nối tiếp điện trở  $R = 200\ \Omega$ . Với Nguồn nuôi là 3.3V trong quá trình sạc, Điện áp rớt trên diode 1N4148 ít nhất là 0.8V và điện trở giúp hạn chế dòng sạc:

$$I = (V_{cc} - V_{diode}) / R$$

Với:  $I$  là dòng điện sạc cho pin 2032

$$V_{cc} = 3.3V$$

$$V_{diode} \geq 0.8V$$

$$R = 200\ \Omega$$

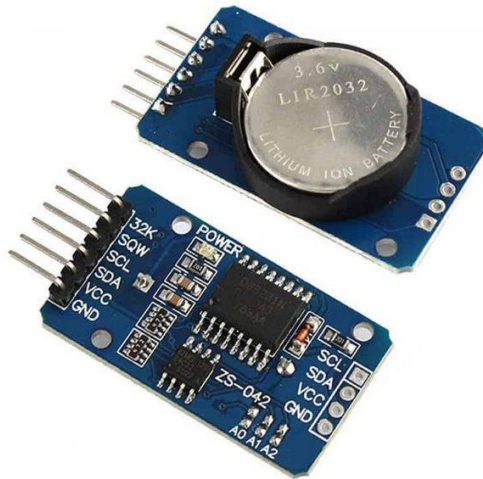
Với  $V_{cc} = 3.3V$  thì pin khi được sạc đầy sẽ rơi vào mức cao nhất là  $3.3V - 0.8V = 2.5V$  Ở 2.5V mạch RTC vẫn hoạt động bình thường khi điện áp của nó lớn hơn 2.3V. Như vậy sẽ đảm bảo được an toàn và độ tin cậy cho pin.

Mạch đèn báo nguồn với  $R = 1k\ \Omega$  giúp hạn chế dòng, bảo vệ Led.

Các tụ  $C$  với giá trị  $C = 100nF$  là các tụ lọc nguồn đầu vào được chọn dựa trên thực nghiệm.

Các bộ điện trở  $R = 7k$  pullup là các điện trở nối pullup hạn chế dòng đầu vào các chân.

#### 2.3.5.2. Lựa chọn linh kiện cho sản phẩm prototype



Hình 2.34 Module thời gian thực DS3231

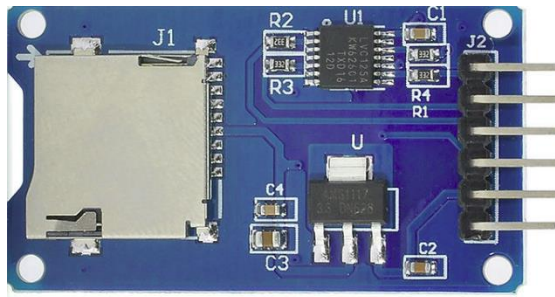
Module thời gian thực DS3231 phù hợp với thiết kế đã tính toán. Module sử dụng IC DS3121 là IC thời gian thực có độ chính xác cao với thạch anh tích hợp sẵn có khả năng điều chỉnh nhiệt. IC có đầu vào cho pin riêng, tách biệt khỏi nguồn chính đảm bảo cho việc giữ thời gian chính xác. Thạch anh tích hợp sẵn giúp tăng độ chính xác trong thời gian dài hoạt động và giảm số lượng linh kiện cần thiết khi làm board.

Thời gian trong IC được giữ ở dạng: giờ, phút, giây, ngày, thứ, tháng, năm. Giao tiếp với IC được thực hiện thông qua I2C bus.

Thông số kỹ thuật:

- IC chính: RTC DS3231 + EEPROM AT24C32
- Nguồn cung cấp: 3.3~5VDC.
- Giao tiếp: I2C
- Lưu trữ và cung cấp các thông tin thời gian thực: ngày, tháng, năm, giờ, phút, giây,...
- Có pin backup duy trì thời gian trong trường hợp không cấp nguồn.
- Kích thước: 38 x 22 x 14mm

### 2.3.6. Thẻ nhớ SD Card



Hình 2.35 Module thẻ nhớ SD Card

Module Micro SD card là module đọc/ghi thẻ nhớ micro SD sử dụng giao tiếp SPI. Có thể đọc/ghi dữ liệu từ micro SD. Cho phép thực hiện các dự án lưu trữ dữ liệu (data logging), phát nhạc MP3...

Thông tin sản phẩm:

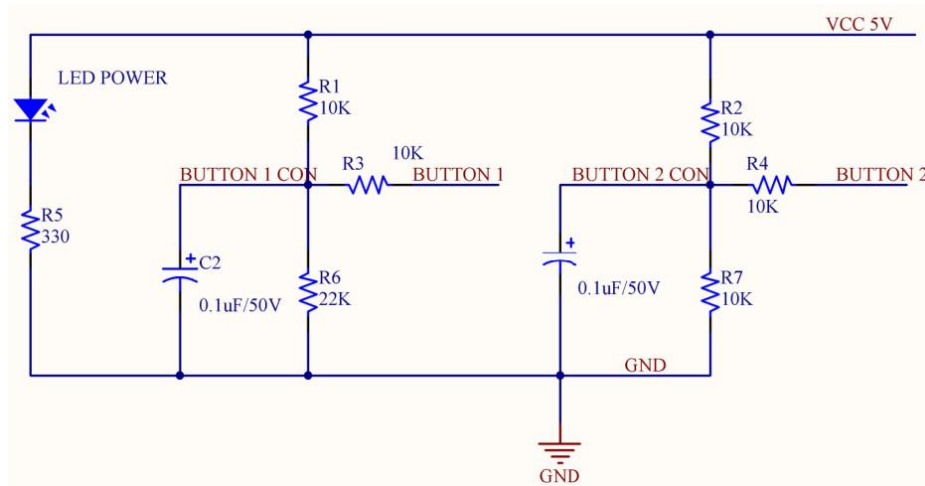
- Hỗ trợ thẻ nhớ micro SD, micro SDHC
- Giao thức: SPI
- Thẻ nhớ Micro SD hỗ trợ định dạng FAT16 và FAT32
- Hỗ trợ việc recording và playback cho lượng âm thanh lớn
- Điện áp cung cấp: 5VDC
- Kích thước dài x rộng x cao: 42mm X 24mm X 12mm

Ghi chú về trình bày mỗi trang của đồ án

### 2.3.7. Khởi nút nhấn điều khiển

Gồm 3 nút nhấn điều khiển dạng pull down dùng để lựa chọn chức năng cho sản phẩm.





Hình 2.36 Mạch nguyên lý nút nhấn điều khiển

Mạch nút nhấn điều khiển sử dụng tụ 0.1uF/50V chống nhiễu phần cứng dựa trên kết quả tính toán thu được khi thiết kế nút nhấn cho bộ xử lý trung tâm.

Lựa chọn nút nhấn tròn loại nhấn nhả kích thước 16mm, gắn ở mặt trên hộp PLC:



Hình 2.37 Nút nhấn vật lý

### 2.3.8. **Khởi hiển thị**

Chức năng hiển thị trạng thái kết nối của các ngoại vi được kết nối với bộ xử lý trung tâm:

A: Active, E: Error

SD: thẻ nhớ SD Card

RTC: Module thời gian thực

SIM: Module SIM7600

WIFI: Kết nối wifi

S: Web Server



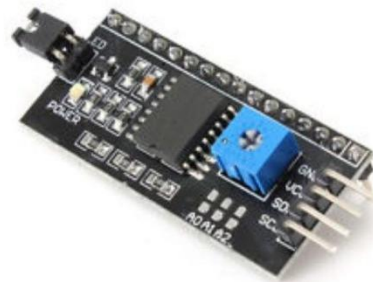
Hình 2.38 Màn hình LCD

LCD là chữ viết tắt của Liquid Crystal Display, là màn hình tinh thể lỏng, đây là loại thiết bị để hiển thị nội dung, cấu tạo bởi các tế bào (cũng là các điểm ảnh) chứa các tinh thể lỏng (liquid crystal) có khả năng thay đổi tính phân cực của ánh sáng và do đó thay đổi cường độ ánh sáng truyền qua khi kết hợp với các kính lọc phân cực.

LCD có ưu điểm là phẳng, cho hình ảnh sáng, chân thật và tiết kiệm năng lượng.

LCD có 16 chân để kết nối với chức năng của mỗi chân.

Để đơn giản kết nối, tiết kiệm chân, LCD thường được sử dụng kèm module I2C:



Hình 2.39 Module I2C

I2C là một chuẩn giao tiếp theo mô hình master/slave. I2C sử dụng 2 dây giao tiếp là SCL và SDA. Dây SCL (Serial Clock Data) là dây truyền xung clock phát từ thiết bị master đồng bộ với việc truyền dữ liệu, dây SDA (Serial Data) là dây truyền dữ liệu.





32

Bảng Công suất tiêu thụ của LAN8720:

Power Pin Group		VDDA3.3 Power PinS(mA)	VDDCR Power pin(mA)	VDDIO power pin(mA)	Total Current (mA)	Total Power (mW)
100BASE-TX /w traffic	Max	28	21	0.6	49	159
	Typical	26	19	0.5	45	148
	Min	23	18	0.3	41	96 Note 5-8
10BASE-T /w traffic	Max	9.7	13	0.6	24	77
	Typical	8.9	12	0.5	22	70
	Min	8.3	12	0.3	20	42 Note 5-8
Energy Detect Power Down	Max	4.2	3.0	0.2	7.4	25
	Typical	4.1	1.9	0.2	6.2	21
	Min	3.9	1.9	0	5.8	16 Note 5-8
General Power Down	Max	0.4	2.8	0.2	3.4	11.2
	Typical	0.3	1.8	0.2	2.3	7.6
	Min	0.3	1.7	0	2	3.0 Note 5-8

Bảng 2.2 Công suất tiêu thụ của LAN8720

$I_{LANmax} = 49mA$ .

Công suất tiêu thụ trên DS3231

### Electrical Characteristics

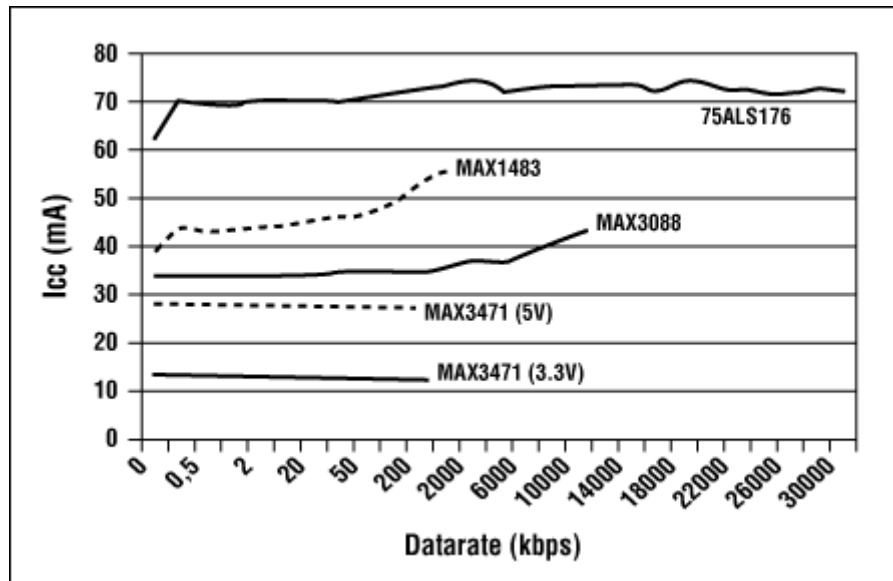
( $V_{CC} = 0V$ ,  $V_{BAT} = 2.3V$  to  $5.5V$ ,  $T_A = T_{MIN}$  to  $T_{MAX}$ , unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Battery Current	$I_{BATA}$	$\overline{E}OSC = 0$ , BBSQW = 0, SCL = 400kHz (Note 5)			70	$\mu A$
					150	
Timekeeping Battery Current	$I_{BATT}$	$\overline{E}OSC = 0$ , BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = $V_{BAT}$ (Note 5)	$V_{BAT} = 3.63V$	0.84	3.0	$\mu A$
			$V_{BAT} = 5.5V$	1.0	3.5	
Temperature Conversion Current	$I_{BATTC}$	$\overline{E}OSC = 0$ , BBSQW = 0, SCL = SDA = 0V or SCL = SDA = $V_{BAT}$	$V_{BAT} = 3.63V$		575	$\mu A$
			$V_{BAT} = 5.5V$		650	
Data-Retention Current	$I_{BATDR}$	$\overline{E}OSC = 1$ , SCL = SDA = 0V, +25°C			100	nA

Bảng 2.3 Công suất tiêu thụ trên DS3231

$I_{DS3231max} = 3\mu A$ .

Công suất tiêu thụ trên RS485:



Hình 2.41 Công suất tiêu thụ trên module RS485-TTL

$I_{RS485-TTL \max} = 30 \text{ mA}$ .

⇒ Tổng dòng điện tiêu thụ:

$$I_{\max} = I_{ESP32} + I_{LAN} + I_{RS485} + I_{DS3231 \max} = 0.147 \text{ A}$$

### 2.3.9.2. Lựa chọn module nguồn

Từ công suất tiêu thụ tính toán, lựa chọn module nguồn HLK-PM01 có dòng trung bình 0.6A làm nguồn cấp cho thiết bị.



Hình 2.42 Module nguồn HLK-PM01

Module nguồn AC-DC Hi-Link HLK-PM01 5VDC 3W có thiết kế nhỏ gọn với vỏ bọc nhựa an toàn, chuyên nghiệp, được sử dụng để chuyển nguồn xoay chiều AC sang 5VDC công suất tối đa 3W cấp cho thiết bị, module được sản xuất bởi hãng Hi-Link chuyên về các module nguồn được sử dụng trong công nghiệp với độ bền, chống nhiễu tốt và độ an toàn cao.

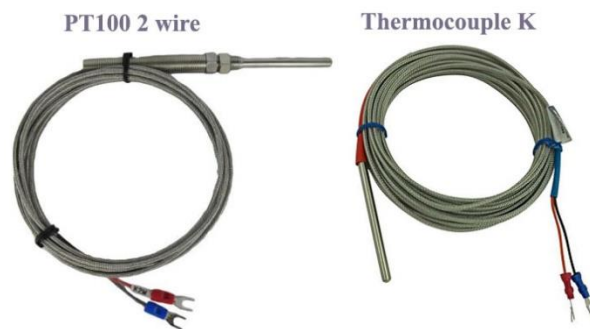
Thông số kỹ thuật:

- Điện áp vào : 100V ~ 240VAC / 50~60Hz

- Điện áp ngõ ra : 5VDC
- Công suất trung bình: 3W
- Đáp ứng yêu cầu UL, CE.
- Độ gọn sóng thấp, ít nhiễu
- Bảo vệ quá tải, ngắn mạch
- Hiệu suất chuyển đổi cao
- Sản phẩm thiết kế đáp ứng tiêu chuẩn EMC (hoạt động được trong môi trường có từ trường)
- Tiêu hao khi không tải  $<0,1W$
- Nhiệt độ hoạt động:  $-20^{\circ}C \rightarrow 80^{\circ}C$

### 2.3.10. Cảm biến nhiệt độ K

Trong phạm vi đề tài, nhóm sử dụng cảm biến nhiệt độ K 2 dây kết nối với đầu vào cảm biến của BCS2 để thử nghiệm và thu thập dữ liệu.



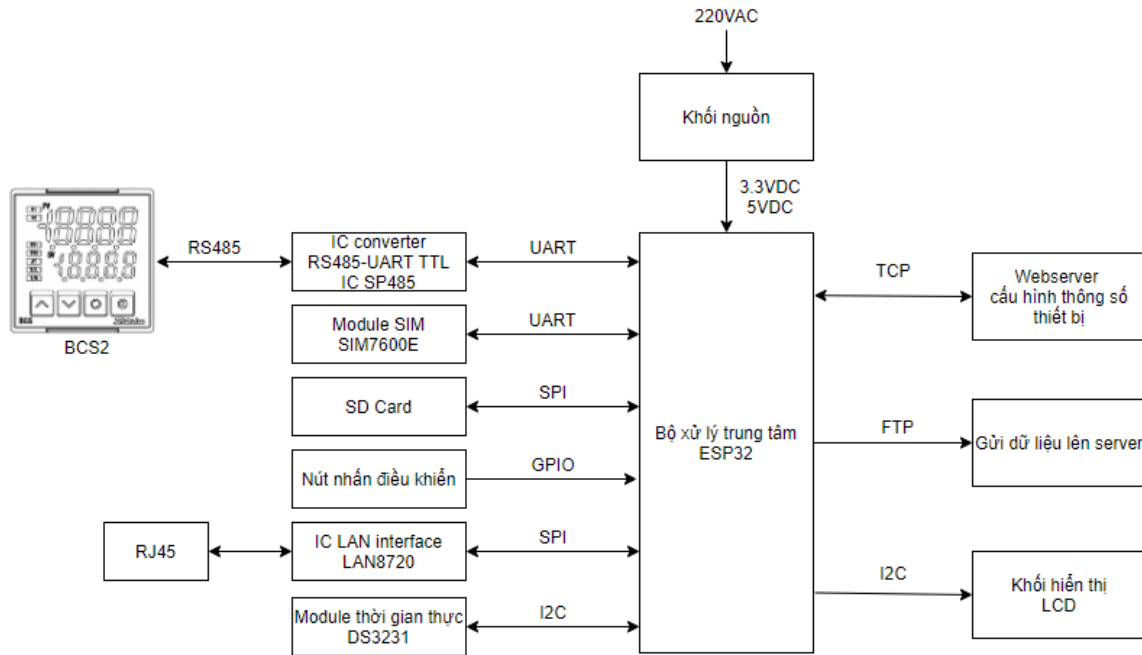
Hình 2.43 Cảm biến nhiệt độ K

Cảm biến nhiệt độ có cấu tạo chính là 2 dây kim loại khác nhau được gắn vào đầu nóng và đầu lạnh. Cảm biến nhiệt hoạt động dựa trên cơ sở là sự thay đổi điện trở của kim loại so với sự thay đổi nhiệt độ môi trường.

Cụ thể, khi có sự chênh lệch nhiệt độ giữa đầu nóng và đầu lạnh thì sẽ có một sức điện động V được phát sinh tại đầu lạnh. Dựa vào loại cảm biến sử dụng ta lựa chọn loại cảm biến tương ứng khi cấu hình thiết bị điều khiển nhiệt độ BCS2 để đo được giá trị chính xác nhất.

## 2.4 Sơ đồ khối nguyên lý chi tiết

Phương án thiết kế chi tiết của nhóm sau quá trình lựa chọn phần cứng:

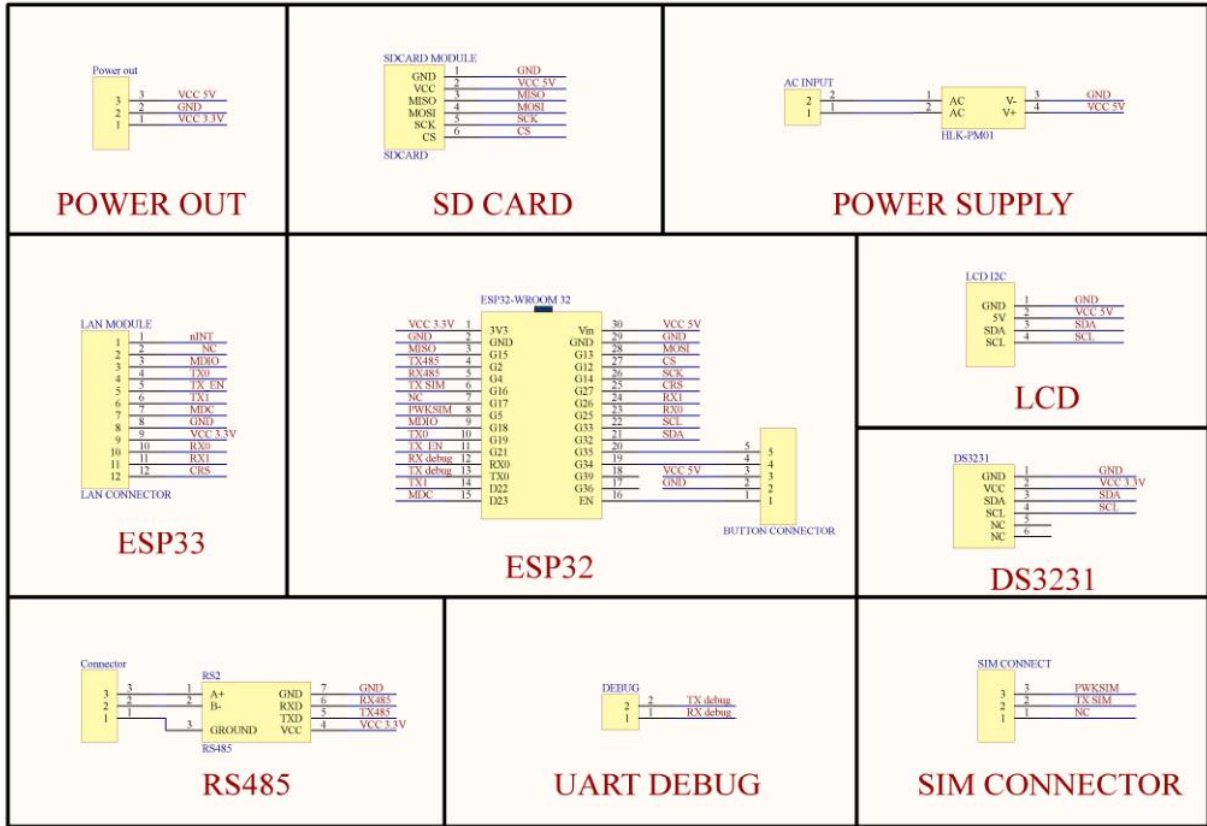


Hình 2.44 Sơ đồ khối phương án thiết kế chi tiết

- Khối nguồn: Cung cấp nguồn cho bộ xử lý trung tâm và các thiết bị ngoại vi.
- Khối xử lý trung tâm: Tín nhận tín hiệu đầu vào từ các thiết bị ngoại vi, xử lý logic chương trình và gửi tín hiệu điều khiển đến các thiết bị đầu ra.
- Khối chuyển đổi RS485-TTL: Chuyển đổi giữa giao thức truyền thông Module và chuẩn TTL giao tiếp giữa BCS2 và bộ xử lý trung tâm.
- Khối Module SIM: Kết nối mạng 4G để gửi dữ liệu lên server
- Khối SD Card: Lưu trữ dữ liệu đọc được từ BCS2 và module thời gian thực.
- Nút nhấn điều khiển: Chức năng thay đổi giữa các Mode LAN/WIFI và 4G, khởi động lại bộ xử lý trung tâm
- Giao diện mạng LAN: Trung gian giao tiếp giữa mạng Ethernet và bộ xử lý trung tâm.
- Module thời gian thực: Lưu giá trị thời gian thực cùng với giá trị đọc được từ BSC2 để ghi vào thẻ nhớ.
- Web server: Cung cấp giao diện cấu hình và hiển thị thông số của thiết bị và bộ BCS2.
- Gửi dữ liệu lên server: Dữ liệu được gửi lên server theo giao thức FTP.
- Khối hiển thị: Hiển thị trạng thái của các thiết bị ngoại vi.

## 2.5 Thi công phần cứng

### 2.5.1. Sơ đồ mạch nguyên lý



Hình 2.45 Sơ đồ mạch nguyên lý

Sản phẩm thiết kế dạng prototype sử dụng các module có sẵn nên cách đi dây phụ thuộc vào cấu tạo chân ra của các Module này.

*Khối nguồn:* Điện áp xoay chiều đưa vào domino đôi được nối với 2 chân vào của Module nguồn HLK-PM01, 2 chân ra của HLK là GND và VCC 5V sử dụng để cấp nguồn cho các khối còn lại trong mạch.

- Khối chuyển đổi truyền thông RS-485: 3 chân YA YB GND kết nối với các chân tương tự của thiết bị BCS2, 4 chân bên còn lại là GND, RX, TX, VCC kết nối với bộ xử lý trung tâm ESP32 theo kết nối truyền thông UART.
- Khối SD CARD: kết nối chip ESP32 theo giao thức SPI, sử dụng 4 chân của vi điều khiển.
- Khối LCD và DS3231: kết nối với ESP32 theo giao thức I2C, mỗi module có 1 địa chỉ riêng, master là ESP32 khi cần điều khiển module nào sẽ gửi lệnh điều khiển kèm địa chỉ của module đó.
- Module LAN8720: module sử dụng nhiều chân và có kích thước lớn nên nhóm sử dụng mạch in để thay đổi chiều cắm module.

- UART DEBUG: Chân ra domino RX TX của chip điều khiển ESP32 , sử dụng để debug, sửa lỗi chương trình.
- SIM CONNECTOR: ra chân domino RX2, TX2, VCC, GND để kết nối với module 4G, sử dụng chức năng 4G Mode, gửi file lên server khi không có kết nối Internet.

Sơ đồ mạch nguyên lý được thiết kế trên phần mềm Altium với bảng nối dây:

GPIO (ESP32)	Thiết bị ngoại vi
0	nINT/REFCLK (50MHz) - 4k7 Pullup (LAN8720)
1	UART0 - debug, upload code
2	txPin UART1 -> txPin module rs485
3	UART0 - debug, upload code
4	rxPin UART1 -> rxPin module rs485
5	Powerkey SIM
12	MISO (SD Card)
13	MOSI (SD Card)
14	SCK (SD Card)
15	CS (SD Card)
16	rxPin uart2 -> txPin SIM7600
17	NC - Osc. Enable - 4k7 Pulldown, txPin uart2 -> rxPin SIM7600
18	MDIO (LAN8720)
19	TX0 (LAN8720)
21	TX_EN (LAN8720)
22	TX1 (LAN8720)
23	MDC (LAN8720)
25	RX0 (LAN8720)
26	RX1 (LAN8720)
27	CRS (LAN8720)
32	SDA (RS3231, I2C LCD)

33	SCL (RS3231, I2C LCD)
34	button1
35	button2

Bảng 2.4 Bảng nối dây ESP32

### 2.5.2. Thiết kế phần cứng

#### 2.5.2.1. Hộp sản phẩm

Mạch in và linh kiện được thiết kế vừa với hộp PLC có sẵn



Hình 2.46 Hộp nhựa PLC

Kích thước hộp: 155x110x60mm

Chất liệu: nhựa

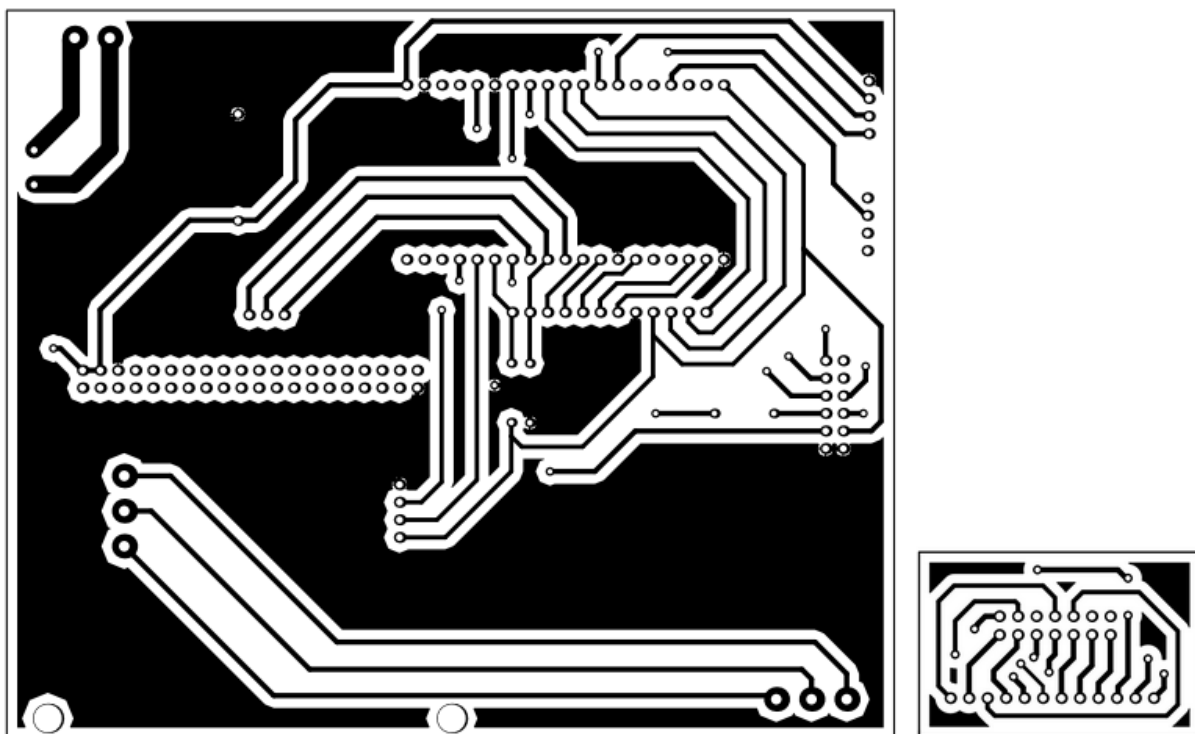
#### 2.5.2.2. Mạch in

Mạch in được xây dựng dựa trên mạch nguyên lý, linh kiện được sắp xếp sao cho cổng LAN RJ45 và cổng kết nối RS485 được đưa ra ngoài phía khe hộp.

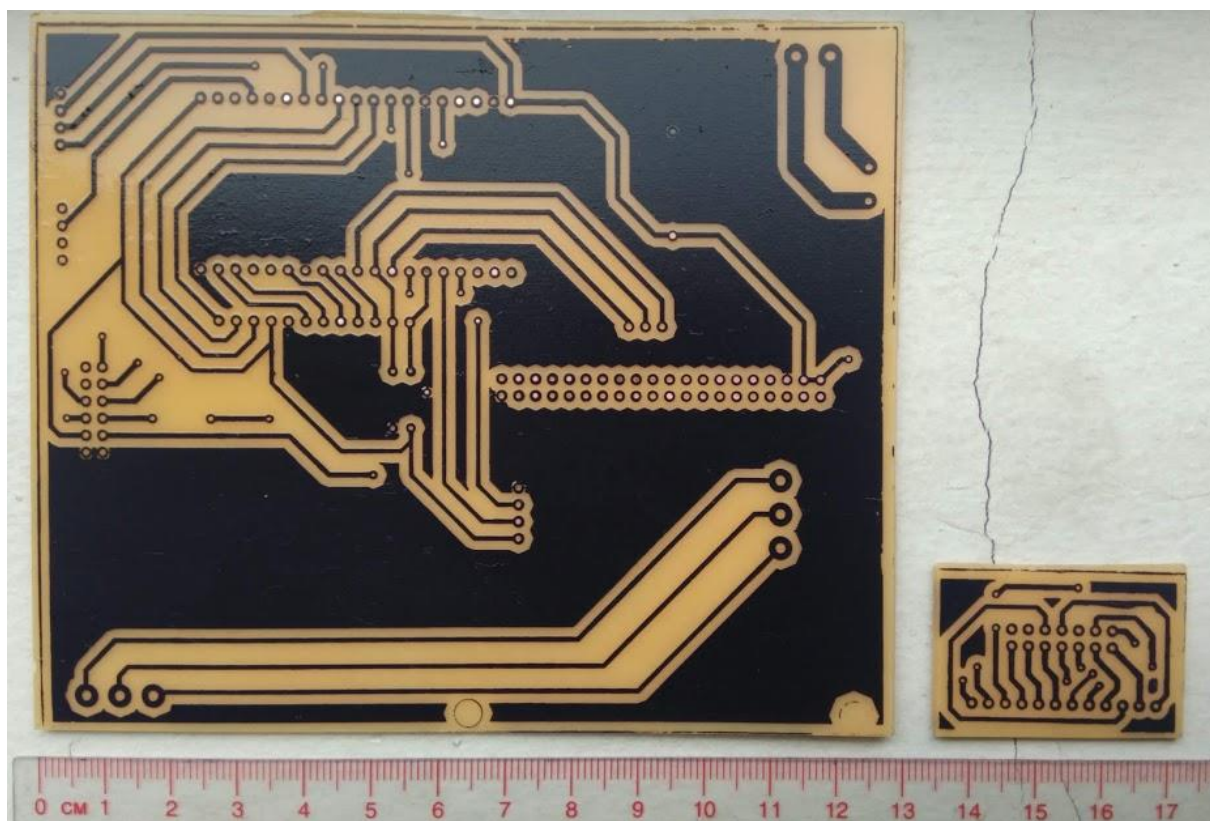
Mạch in được chia làm 1 mạch chính và 2 mạch phụ:

- 1 mạch chính gắn các module ESP32, DS3231, LAN, RS485-TTL.
- 1 Mạch phụ sử dụng cho nút nhấn và đèn báo nguồn.
- 1 mạch phụ sử dụng để ra chân LAN8720 và kết nối với mạch chính.





Hình 2.47 Mạch in pdf



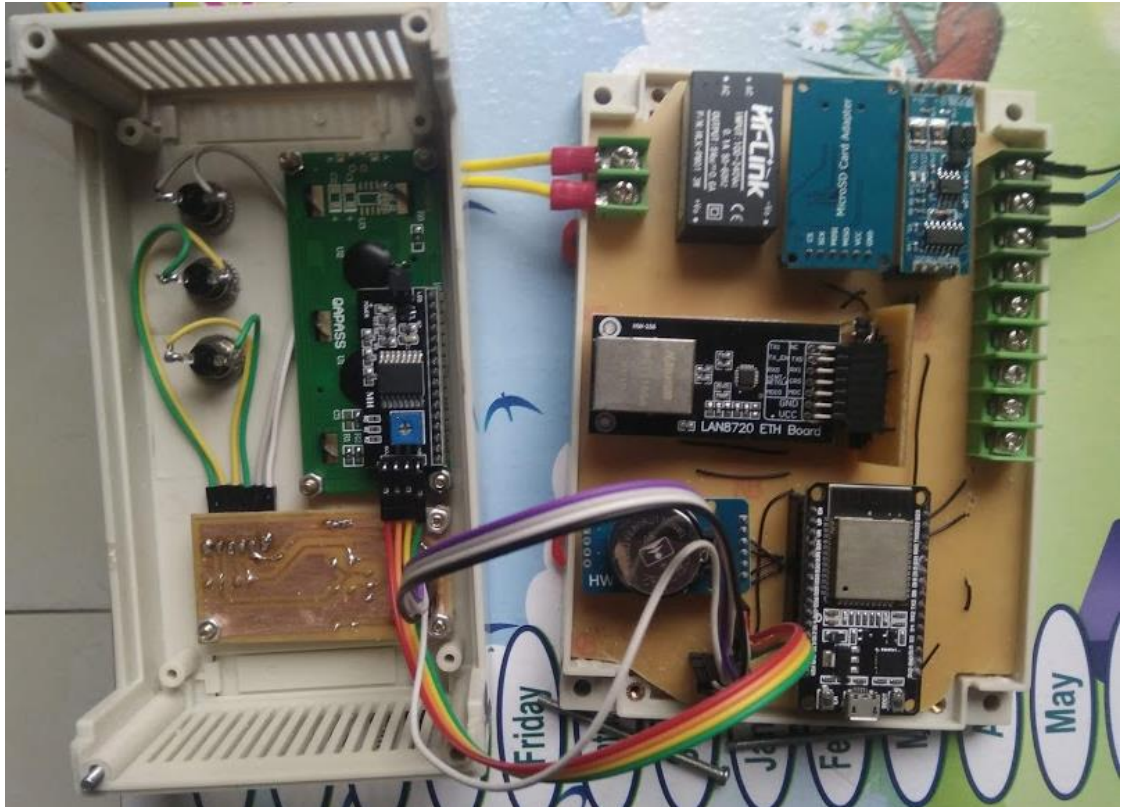
Hình 2.48 Mạch in 1 lớp trên bo đồng

### 2.5.3. Lắp ráp và kiểm tra

Sau khi mạch in hoàn thành, tiến hành khoan lỗ và hàn linh kiện.

Các bước tiến hành:

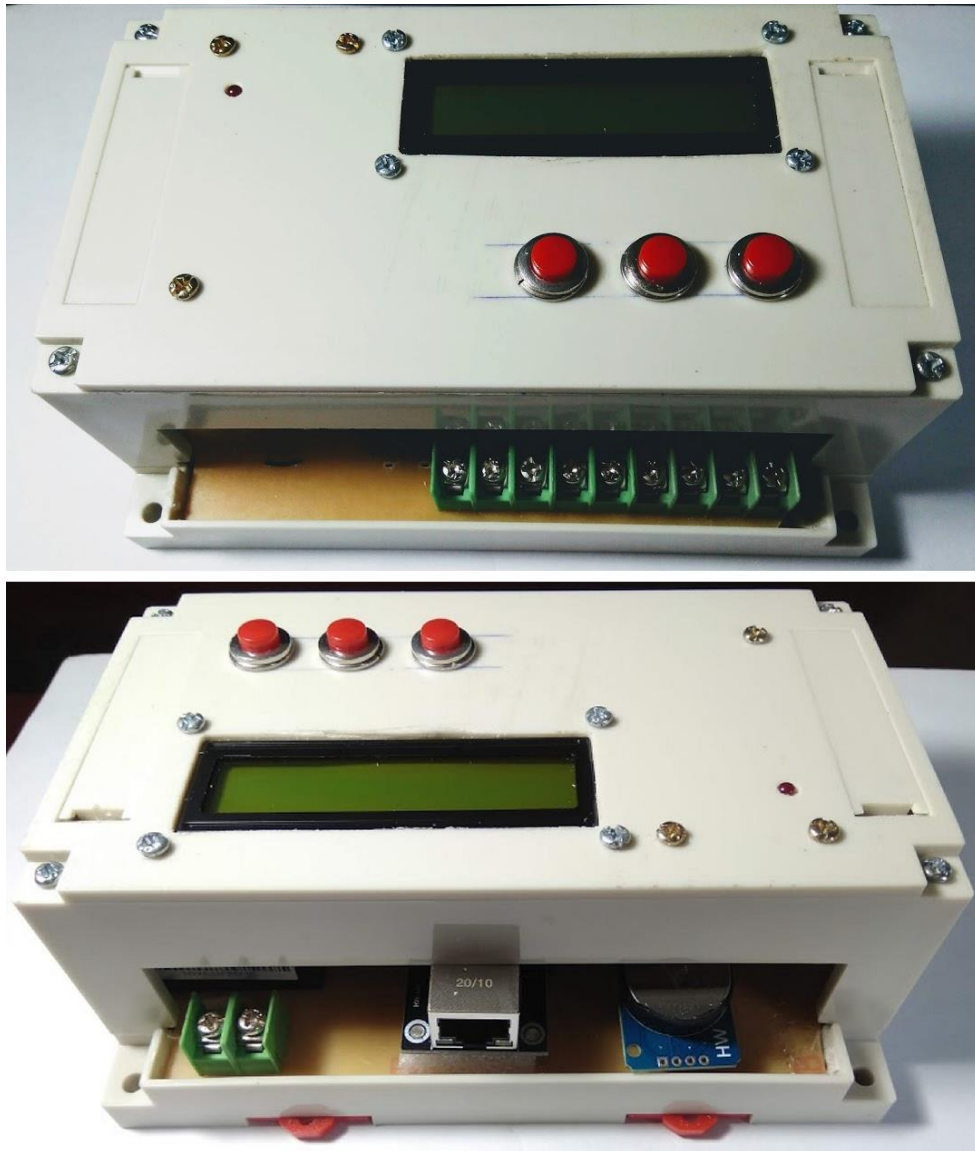
- Sau khi mạch in hoàn thành, kiểm tra lại mạch in so với sơ đồ nguyên lý.
- tiến hành khoan lỗ.
- Hàn linh kiện.
- Dùng đồng hồ VOM kiểm tra ngắn mạch, hở mạch.
- Gắn nguồn, kiểm tra điện áp nguồn, điện áp DC sau nguồn.
- Nạp chương trình và chạy thử thiết bị.



Hình 2.49 Mạch thiết bị sau khi gắn linh kiện

Kết quả đạt được:

- Đáp ứng yêu cầu đề tài sử dụng điện áp 220VAC.
- Mạch in vừa với hộp PLC.
- Thiết bị hoạt động ổn định.



*Hình 2.50 Thiết bị sau khi đóng hộp*

## CHƯƠNG 3: THIẾT KẾ PHẦN MỀM VÀ ĐÁNH GIÁ KẾT QUẢ

### 3.1 Tính năng thiết bị

Thiết bị điều khiển nhiệt độ BCS2 sau khi được gắn cảm biến nhiệt độ K và cấp nguồn sẽ được kết nối với Module IoT mở rộng qua dây RS485.

- Cắm nguồn cho module mở rộng và kết nối dây mạng. Module sẽ nạp lại cấu hình trước đó trong bộ nhớ flash.
- Kiểm tra kết nối LAN, nếu không khả dụng chuyển sang kết nối wifi sử dụng tên wifi và mật khẩu trong file cấu hình trước đó. Nếu Wifi không khả dụng chuyển sang kết nối 4G. Ở chế độ 4G Module sẽ gửi file lên server theo các thông số cài đặt sẵn.
- Khi có kết nối LAN/Wifi, khởi động web server. Web server được truy cập qua tên miền trong mạng nội bộ http://esp32.local hoặc địa chỉ ip dạng 192.168.x.x.
- Điều khiển thiết bị BCS2 bằng gói tin chứa lệnh điều khiển thông qua giao thức truyền thông Modbus-RTU và đọc thông số từ thiết bị.
- Màn hình LCD hiển thị trạng thái thiết bị ngoại vi: kết nối LAN/Wifi, kết nối thẻ nhớ SD Card, Module thời gian thực DS3231.

### 3.2 Tìm hiểu phần mềm

#### 3.2.1. Đọc ghi dữ liệu với thiết bị điều khiển nhiệt độ BCS2

##### 3.2.1.1. Đọc giá trị SV, PV

Gói tin được gửi dưới dạng mã thập lục phân, bao gồm ID của thiết bị BCS2, Mã chức năng, dữ liệu và mã crc-16

3.5 idle characters	Slave address (01H)	Function code (03H)	Data item (0100H)	Amount of data (0001H)	Error check CRC-16 (85F6H)	3.5 idle characters
	1	1	2	2	2	

Hình 3.1 Tin nhắn yêu cầu đọc giá trị PV

Mã chức năng đọc chung là 03H

Data item: 0100H, đọc giá trị hiện tại của PV.

Khi gói tin yêu cầu đọc được gửi đúng cú pháp, BCS2 sẽ gửi gói tin phản hồi chứa dữ liệu mà module mở rộng đã yêu cầu:

3.5 idle characters	Slave address (01H)	Function code (03H)	Response byte count (02H)	Data (0258H)	Error check CRC-16 (B8DEH)	3.5 idle characters
	1	1	1	2	2	

Hình 3.2 Tin nhắn phản hồi giá trị PV



Chương trình đọc giá trị SV,PV:

Bước 1: Gửi tin nhắn yêu cầu đọc giá trị của PV

Gói tin được lưu vào một mảng byte array. Sau đó sử dụng vòng lặp for lặp qua từng giá trị và lệnh Serial.write() để truyền giá trị đến cổng uart kết nối module rs485-ttl.

Gói tin có dạng: byte msg\_sv[] = {0x01, 0x03, 0x01, 0x03, 0x00, 0x01, 0x75, 0xF6};

Bước 2: Chờ phản hồi và ghi giá trị phản hồi vào mảng mới.

Sau khi gửi yêu cầu đọc giá trị PV, BCS2 sẽ phản hồi lại gói tin chứa giá trị PV hiện tại. Sử dụng vòng lặp while() để lưu từng byte phản hồi từ cổng uart vào mảng.

Bước 3: Xử lý gói tin.

Gói tin phản hồi có định dạng được quy định sẵn trong tài liệu của thiết bị. Giá trị của SV là byte thứ 4 và 5 trong gói tin phản hồi. Tách 2 byte này và chuyển đổi về chuyển đổi sang mã thập phân, ta có được giá trị PV theo nhiệt độ Celsius.

### 3.2.1.2. Ghi giá trị SV

Tương tự như đọc giá trị PV, ta cần thay đổi function code và data item phù hợp với yêu cầu.

Yêu cầu đặt giá trị SV = 600°C

3.5 idle characters	Slave address (01H)	Function code (06H)	Data item (0001H)	Data (0258H)	Error check CRC-16 (D890H)	3.5 idle characters
	1	1	2	2	2	

Giá trị SV đổi sang hệ thập lục phân 2 byte và gửi gói tin qua cổng uart.

Gói tin phản hồi:

3.5 idle characters	Slave address (01H)	Function code (06H)	Data item (0001H)	Data (0258H)	Error check CRC-16 (D890H)	3.5 idle characters
	1	1	2	2	2	

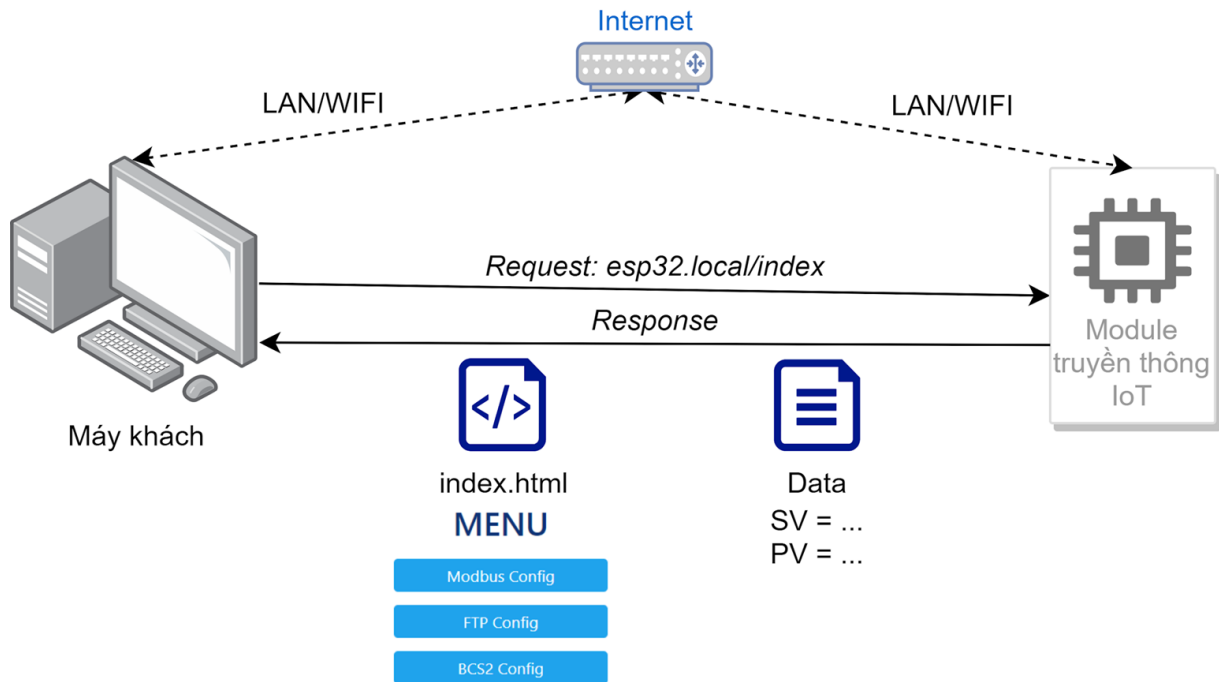
3.5 idle characters	Slave address (01H)	Function code (86H)	Exception code (03H)	Error check CRC-16 (0261H)	3.5 idle characters
	1	1	1	2	

Hình 3.3 Gói tin phản hồi ghi giá trị SV

Gói tin phản chứa giá trị SV đã đặt hoặc mã lỗi nếu không đặt giá trị thành công.

### 3.2.2. Webserver

Webserver được nhúng trong bộ nhớ flash của bộ xử lý trung tâm ESP32, bao gồm các file html, css. Vì vậy web server không chiếm dụng bộ nhớ của chương trình chính.



Hình 3.4 Nguyên lý hoạt động của Web Server

Yêu cầu máy tính hoặc điện thoại kết nối đến cùng mạng cục bộ LAN với module truyền thông IoT và Web Server trên Module đã hoạt động. Khi có request gửi từ trình duyệt máy tính tới Web Server nhúng trên Module thông của tên miền hoặc địa chỉ IP của module trong mạng, giao diện web sẽ được phản hồi và hiển thị lại trên trình duyệt web, tùy theo phần đường dẫn sau địa chỉ (/index hoặc /modbus) mà Web Server sẽ phản hồi lại các giao diện tương ứng.

#### 3.2.2.1. Phần mềm lập trình vi điều khiển trung tâm

Hiện nay có rất nhiều phần mềm hỗ trợ lập trình vi điều khiển ESP32:

- Phần mềm ESP-IDF (Espressif IoT Development Framework): công cụ lập trình của hãng Espressif, hãng sản xuất chip ESP8266 và ESP32. Công cụ cho phép can thiệp sâu vào phần cứng của chip.
- Platform.IO: extension sử dụng trên phần mềm VS Code dùng để lập trình cho các dòng chip nhúng. Ưu điểm của PlatformIO là tính năng quản lý thư mục, gợi ý lệnh, sử dụng ngôn ngữ C để lập trình.
- MicroPython: công cụ lập trình cho ESP32 sử dụng ngôn ngữ C.
- Arduino IDE: Môi trường phát triển được sử dụng phổ biến cho các bo mạch Arduino như Arduino UNO R3, Arduino MEGA, Arduino Pro Mini

Lựa chọn phần mềm lập trình Arduino với những ưu điểm sau:

- Chức năng viết chương trình, biên dịch, nạp chương trình tích hợp trong một ứng dụng.
- Sử dụng ngôn ngữ C và C++ quan thuộc với sinh viên kỹ thuật.
- Cộng đồng hỗ trợ lớn, kho thư viện đa dạng phong phú giúp xây dựng dự án nhanh chóng mà không cần phải đi sâu tìm hiểu thiết bị.



Hình 3.5 Phần mềm Arduino IDE

#### 3.2.2.2. Thư viện xây dựng Webserver

Để có thể nhúng Webserver vào chip ESP32 ta cần một thư viện xây dựng máy chủ web HTTP, xử lý các yêu cầu từ client. Client có thể là laptop, điện thoại kết nối cùng mạng với ESP32.

Thư viện Webserver bất đồng bộ ESPAsyncWebServer:

ESPAsyncWebServer là thư viện sử dụng cho ESP32, Sử dụng máy chủ web không đồng bộ hiệu suất cao. Được khuyên dùng thay thế cho máy chủ web WebServer mặc định trên ESP32 và ESP8266.

Ưu điểm của Webserver sử dụng thư viện ESPAsyncWebServer:

- Sử dụng mạng không đồng bộ có thể xử lý nhiều kết nối cùng một lúc.
- Bạn được gọi khi yêu cầu đã sẵn sàng và được trả về.
- Tốc độ là nhanh hơn, đem lại trải nghiệm tốt hơn cho người sử dụng.
- Dễ dàng sử dụng API, HTTP, để tùy biến lại Webserver
- Dễ dàng mở rộng để xử lý bất kỳ loại nội dung.
- Hỗ trợ lên tới 100 yêu cầu cùng lúc.

Lưu ý:

- Đây là máy chủ không đồng bộ hoàn toàn, không chạy trên chuỗi vòng lặp `loop()`.
- Không thể sử dụng hàm `delay` hoặc bất kỳ hàm chức năng nào sử dụng chúng trong yêu cầu phản hồi về máy khách.
- Máy chủ không thể gửi nhiều hơn một phản hồi cho một yêu cầu.

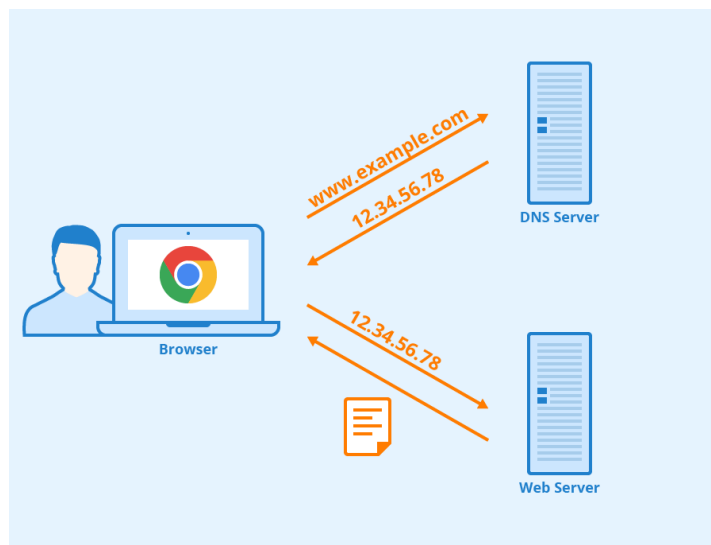
### 3.2.2.3. Thư viện mDNS

Khi tên miền (domain) chưa xuất hiện, người ta phải truy cập các trang web bằng cách gõ địa chỉ IP lên trình duyệt. Mà địa chỉ IP thì chẳng dễ nhớ chút nào. Đó là lý do tại sao tên miền sử dụng.

Tuy nhiên, để gửi yêu cầu đến một trang web, máy tính cần phải biết địa chỉ IP của nó, DNS ra đời để giải quyết vấn đề này. DNS là viết tắt của Domain Name System, nơi để ánh xạ tên miền của trang web sang địa chỉ IP của nó. Trên Internet, có rất nhiều máy chủ DNS. Mỗi máy chủ DNS có một danh sách dài các tên miền và địa chỉ IP tương ứng của chúng.

DNS như danh bạ điện thoại, thay vì phải nhớ số giờ thì chỉ cần tra cứu tên người cần liên lạc.

Khi bạn truy cập một trang web trong trình duyệt, nó sẽ gửi một yêu cầu tới máy chủ DNS. Sau khi máy chủ DNS tra cứu và phản hồi lại địa chỉ IP, trình duyệt sẽ gửi yêu cầu đến đúng địa chỉ IP.



Hình 3.6 Mô hình hoạt động của DNS

Tuy nhiên trong mạng nội bộ (LAN) không thể sử dụng máy chủ DNS của riêng mình. Nghĩa là không thể dùng tên miền để truy vấn đến các thiết bị trong mạng nội bộ.

Để khắc phục vấn đề này nhóm sử dụng giao thức mDNS(Multicast Domain Name System) phân giải tên máy chủ thành địa chỉ IP trong mạng nội bộ.

mDNS sử dụng tên miền với đuôi là .local (http://esp32.local). Nếu máy tính cần gửi yêu cầu đến một tên miền có đuôi là .local, nó sẽ gửi một truy vấn multicast đến tất cả các thiết bị khác trên mạng LAN có hỗ trợ mDNS, thiết bị sẽ tự nhận dạng tên miền của chính mình. Sau đó thiết bị có tên phù hợp sẽ phản hồi bằng cách gửi một multicast khác



kèm địa chỉ IP của nó. Đến lúc này máy tính của biết địa chỉ IP của thiết bị và có thể gửi các yêu cầu bình thường.

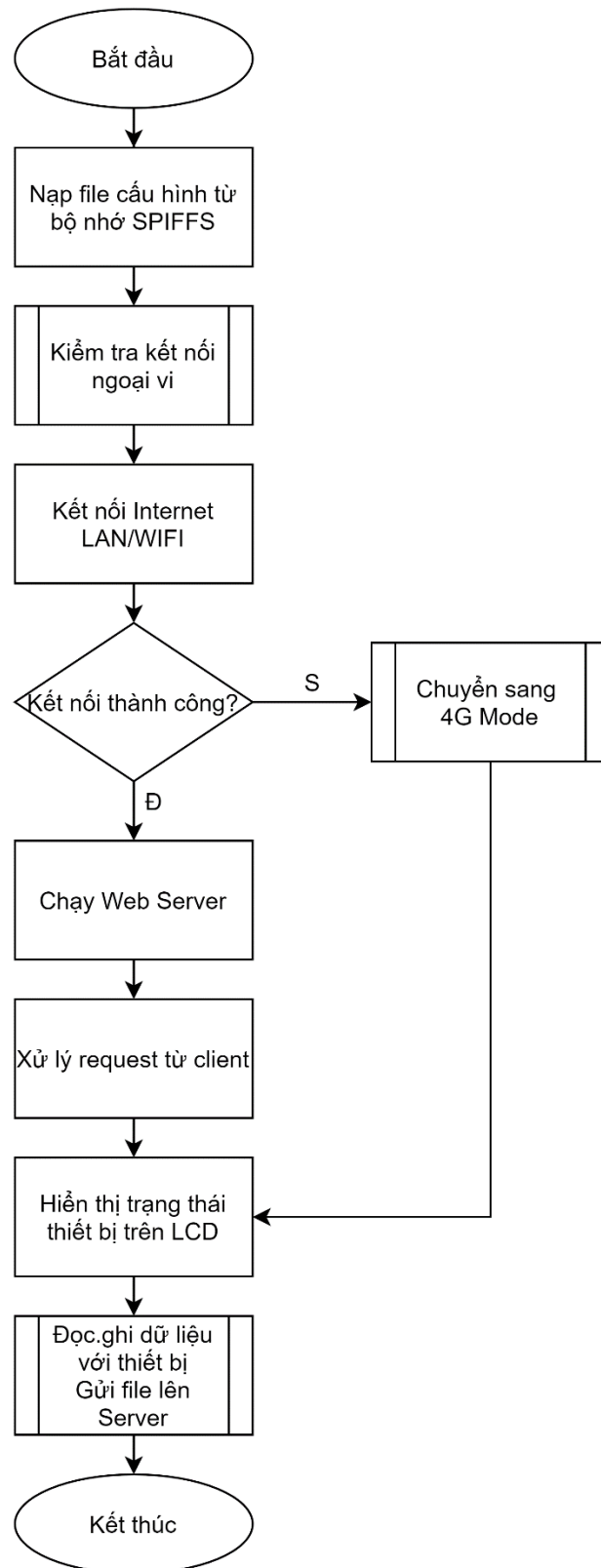
=> Việc sử dụng mDNS cho thiết bị giúp thiết bị có thể kết nối mạng LAN bất kỳ và người sử dụng chỉ cần truy cập vào đường dẫn mặc định (<http://esp32.local>) là có thể mở website cấu hình mà không cần truy cập qua địa chỉ IP(102.168.x.x).

Thư viện mDNS được thêm trực tiếp vào chương trình bằng lệnh

```
#include "ESPmDNS.h"
```

### 3.3 Lưu đồ thuật toán

#### 3.3.1. Chương trình tổng quan



Hình 3.7 Lưu đồ thuật toán tổng quan

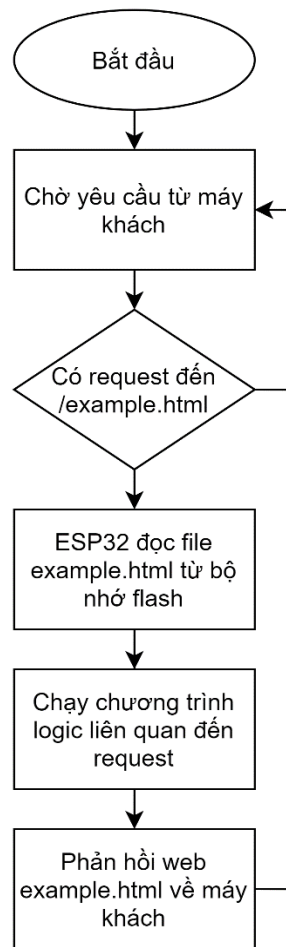
### Giải thích chương trình:

Đảm bảo bộ điều khiển nhiệt độ BCS2 đã kết nối nguồn và sử dụng cấu hình truyền thông Modbus (Modbus-RTU, Instrument number: 1), đầu vào cảm biến nhiệt độ K.

- Nạp file cấu hình từ bộ nhớ SPIFFS. Bao gồm các thông số về cấu hình truyền thông modbus-RTU, các thông tin về FTP server.
- Kiểm tra trạng thái hoạt động của các thiết bị ngoại vi, hiển thị trạng thái lên màn hình LCD.
- Kết nối Internet: Kiểm tra kết nối mạng LAN, sau 5s không có kết nối chuyển sang kết nối Wifi, sau 5s không có kết nối wifi chuyển sang chế độ 4G.
  - o Chế độ LAN/Wifi: Chạy Webserver, xử lý các yêu cầu từ máy khách.
  - o Chế độ 4G: sử dụng thông số cấu hình mặc định kết nối Modbus với thiết bị BCS2, đọc giá trị SV, PV và lưu dưới dạng file 420ma.txt vào thẻ nhớ, gửi file 420ma.txt lên Server.
- Khi máy khách gửi yêu cầu chạy thiết bị qua website. Chạy chương trình đọc giá trị SV, PV trên ESP32. Giá trị SV, PV sau khi đọc từ ESP32 sẽ được lưu vào bộ nhớ và phản hồi về máy khách khi có yêu cầu hiển thị.
- Gửi file dữ liệu lên server:
  - o Chế độ LAN/Wifi: file được gửi theo giao thức FTP lên server thông qua kết nối Internet.
  - o Chế độ 4G: Sử dụng tập lệnh AT lập trình module SIM gửi file lên FTP server qua kết nối 4G.

Chương trình đọc giá trị SV/PV.

### 3.3.2. Chương trình khi xử lý yêu cầu từ máy khách



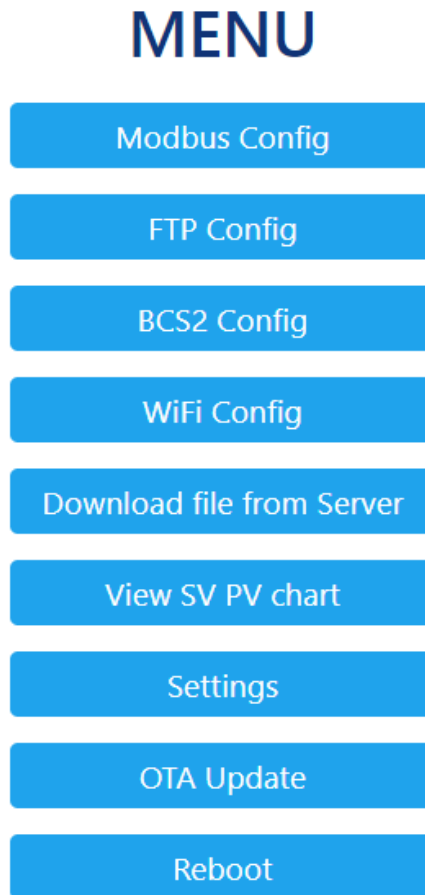
Hình 3.8 Lưu đồ thuật toán tổng quan xử lý request

Sau khi web server chạy, máy chủ sẽ luôn chờ yêu cầu từ máy khách.

mặc định khi máy khách truy cập vào tên miền hay địa chỉ IP của máy chủ có dạng <http://esp32.local>, máy chủ sẽ thực hiện yêu cầu theo lệnh lập trình trong vi điều khiển ESP32 và trả về website máy khách yêu cầu:

```
server.on("/", HTTP_GET, []) (AsyncWebServerRequest * request) {  
    request->send(SPIFFS, "/index.html");  
};
```

Khi có truy vấn theo giao thức HTTP\_GET đến đường dẫn <http://esp32.local> ESP32 đọc file “/index.html” từ bộ nhớ SPIFFS và phản hồi lại cho máy khách. Giao diện hiển thị ở trình duyệt của máy khách:



Hình 3.9 Giao diện trang chủ

Từ giao diện trang chủ, khi máy khách chọn chức năng bất kì sẽ được chuyển hướng đến đường link tương ứng.

Ví dụ: Khi chọn nút Reboot, máy khách gửi 1 HTTP\_GET đến máy chủ theo đường dẫn /reboot. Máy chủ chạy trên ESP32 tiếp nhận yêu cầu và thực hiện lệnh ESP.restart() khởi động lại chương trình.

```
server.on("/reboot", HTTP_GET, [] (AsyncWebServerRequest * request) {  
    ESP.restart();  
});
```

### 3.3.3. Chương trình xử lý yêu cầu từ máy khách kèm tham số

Ở giao diện trang chủ, khi máy khách chọn chức năng “Modbus Config”, giao diện web tương ứng đường dẫn /modbus.html được trả về.

**Config MODBUS-RTU protocol**

Variable	Current Value
Baudrate	9600
Data bit	8
Parity	parity
Stop bit	1
Instrument	1

List of Current Variable

Communication speed: 9600 bps ▼

Data bit: 8 bits ▼

Parity: No parity ▼

Stop bit: 1 bit ▼

Instrument number (1->94): 1

[Return to Home Page](#)

Hình 3.10 Giao diện cấu hình modbus.html

Trang Web trả về gồm các biến giữ chỗ %PLACE HOLDER% có dạng %baud%, %data\_bit%, Khi máy chủ biên dịch lại web modbus.html có những biến giữ chỗ này, giá trị trả về tương ứng với mỗi biến sẽ được hiển thị lên web, như ta thấy trong hình là các thông số cấu hình trong bộ nhớ SPIFFS ở cột “Current Value”.

```
String processor(const String &var)
{
  //Serial.println(var);
  if (var == "baud")
  {
    return readFile(SPIFFS, "/baud.txt");
  }
  if (var == "data_bit")
  {
    return readFile(SPIFFS, "/data_bit.txt");
  }
  if (var == "stop_bit")
  {
    return readFile(SPIFFS, "/stop_bit.txt");
  }
}
```

Ở Form Submit gồm các giá trị cấu hình mới, khi thay đổi những giá trị này và bấm **Submit**, một HTTP\_GET request được gửi đến đường dẫn đã lập trình.

Communication speed: 9600 bps  
Data bit: 8 bits  
Parity: No parity  
Stop bit: 1 bit  
Instrument number (1->94): 1  
Submit

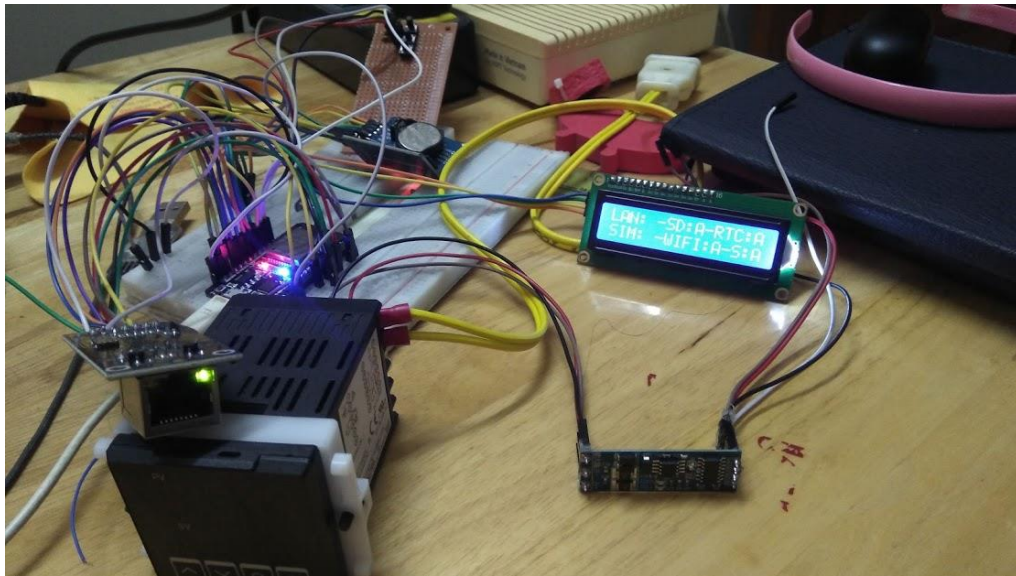
[http://esp32.local/modbus-config?baud=9600&data\\_bit=8&parity=no&stop\\_bit=1&instrument=1](http://esp32.local/modbus-config?baud=9600&data_bit=8&parity=no&stop_bit=1&instrument=1)

Hình 3.11 Đường dẫn gồm các tham giá cấu hình

Đường dẫn gồm các tham số này, sẽ được lưu vào bộ nhớ SPIFFS và áp dụng cho cấu hình truyền thông modbus sau khi khởi động lại thiết bị.

### 3.4 Chạy thử phần mềm




Chương trình được chạy thử trên board test trong quá trình phát triển:



Hình 3.12 Test board thử nghiệm chương trình

Các giai đoạn tiến hành viết chương trình cho module truyền thông IoT.

Giai đoạn 1: Viết chương trình thử nghiệm kết nối giữa các ngoại vi và bộ xử lý trung tâm

Tên	Chủ sở hữu	Sửa đổi lần...	↑
 esp32-ota-web-updater-code_ota	tôi	13 thg 10, 2020	tôi
 esp32-webserver-login	tôi	13 thg 10, 2020	tôi
 ESP32-web-template-sd	tôi	13 thg 10, 2020	tôi
 test_LAN_esp32	tôi	13 thg 10, 2020	tôi
 test-spiffs	tôi	13 thg 10, 2020	tôi
 test-uart	tôi	13 thg 10, 2020	tôi

*Hình 3.13 Chương trình thử nghiệm kết nối phần cứng*

Dựa trên thiết kế phần cứng, viết chương trình kết nối cho từng module, đảm bảo module hoạt động ổn định.

Giai đoạn 2: Kết nối phần cứng, ghép chương trình.

Giai đoạn 3: Xây dựng web server.

Giai đoạn 4: Chạy thử nghiệm thiết bị.

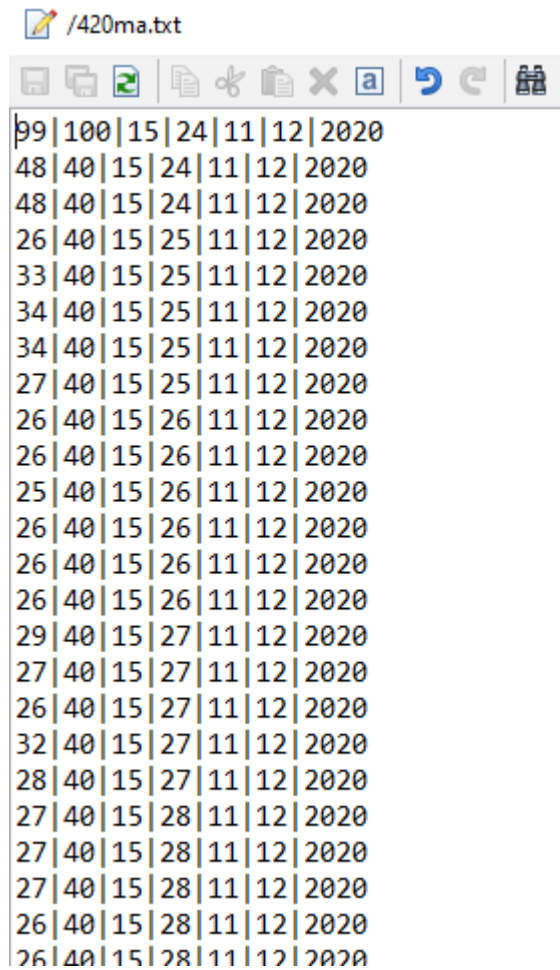
#### **3.4.1. Chức năng cấu hình**

Tại các mục cấu hình Modbus, FTP, Wifi, Device. Thông số cấu hình sau khi nhập trên web được gửi lên web server nhúng trong ESP32, thông số này được lưu lại trong bộ nhớ SPIFFS và được sử dụng để cấu hình lại thiết bị trong lần khởi động sau.

#### **3.4.2. Chức năng lưu file vào thẻ nhớ.**

Sau một thời gian chạy thử, giá trị nhiệt độ độ ẩm lưu vào thẻ nhớ theo thời gian cố định hoạt động ổn định. Sử dụng đầu đọc thẻ nhớ đọc lại file lưu cho kết quả như hình:



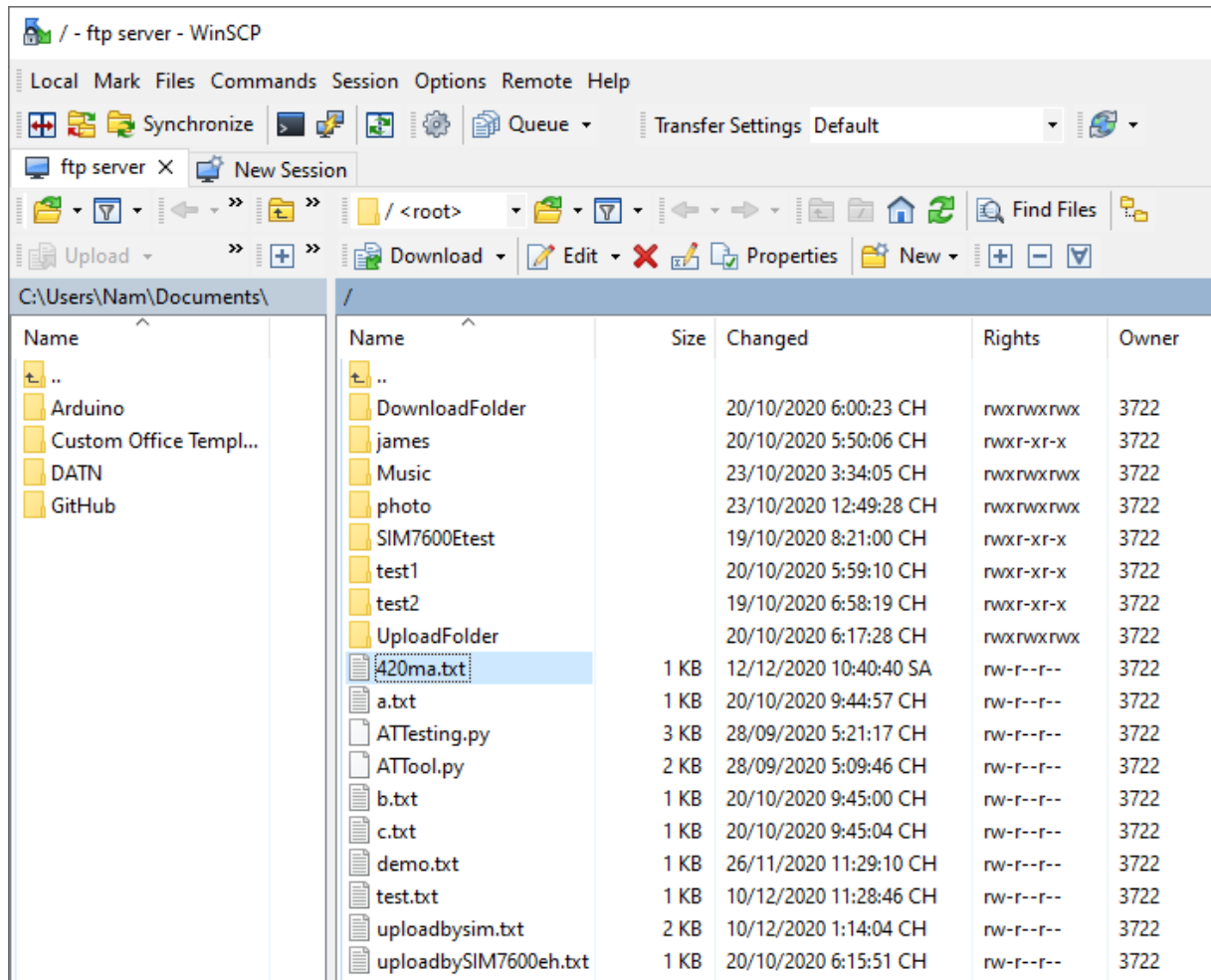


Hình 3.14 File 420ma.txt trong thẻ nhớ

### 3.4.3. Chức năng tải file lên Server

Kiểm tra FTP Server đã cấu hình, file 420ma.txt được tạo mới theo chu kỳ tải file lên server từ thiết bị. Sử dụng phần mềm WinSCP để kiểm tra dữ liệu.

Nội dung file tải lên Server tương tự nội dung file lưu trong thẻ nhớ.



Hình 3.15 Sử dụng phần mềm WinSCP truy cập FTP Server

#### 3.4.4. Chức năng tải file từ Server

Tại giao diện trang chủ, lựa chọn chức năng “Download file from server”, ESP32 thực hiện chương trình tải file từ FTP Server theo các thông số IP FTP Server, User, Password đã nhập. Khi tải file định dạng “.txt” thành công, nội dung file sẽ được hiển thị lại trên giao diện web.

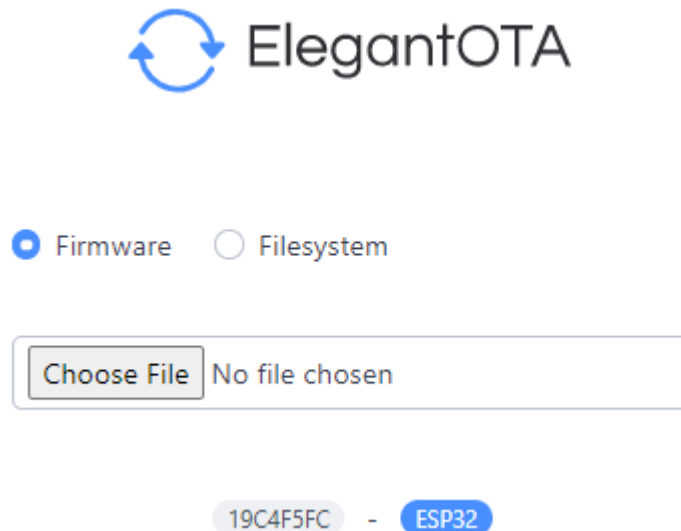
## Download file from server

PV|SV|Giờ|Phút|Ngày|Tháng|Năm

34	56	05	56	05	10	2020
99	100	7	25	27	11	2020
99	100	7	26	27	11	2020
26	183	7	26	27	11	2020
26	183	7	27	27	11	2020
26	183	7	27	27	11	2020
26	183	7	27	27	11	2020
26	183	7	27	27	11	2020
26	183	7	27	27	11	2020
26	183	7	27	27	11	2020
26	183	7	28	27	11	2020
26	183	7	28	27	11	2020
26	183	7	28	27	11	2020
26	183	7	28	27	11	2020
26	183	7	28	27	11	2020
26	183	7	28	27	11	2020
26	183	7	29	27	11	2020
26	183	7	29	27	11	2020
26	183	7	29	27	11	2020
26	183	7	29	27	11	2020
26	183	7	29	27	11	2020
26	183	7	29	27	11	2020
26	183	7	29	27	11	2020
26	183	7	30	27	11	2020
26	183	7	30	27	11	2020
26	183	7	30	27	11	2020
26	183	7	30	27	11	2020
26	183	7	30	27	11	2020
26	183	7	31	27	11	2020
26	183	7	31	27	11	2020
26	183	7	31	27	11	2020

Hình 3.16 Chức năng tải file từ Server

### 3.4.5. Chức năng OTA Update



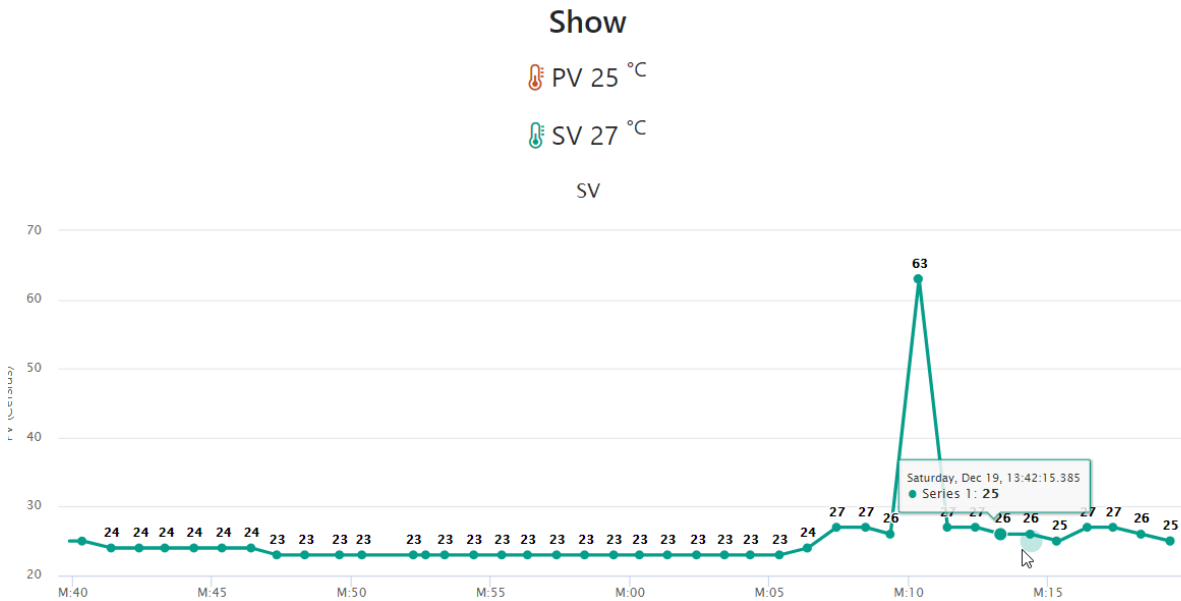
Hình 3.17 Giao diện chức năng OTA Update

Chức năng OTA Update được sử dụng để cập nhật chương trình cho vi xử lý ESP32, giúp việc nạp lại chương trình trở nên đơn giản hơn, có thể thực hiện từ xa mà không cần nối dây trực tiếp phân cứng.

Lựa chọn file **.bin** từ máy tính sau đó bấm Open. Firmware mới tự động được nạp và khởi động lại thiết bị.

### 3.4.6. Chức năng hiển thị giá trị PV

Khi lựa chọn chức năng hiển thị giá trị PV, ESP32 gửi gói tin đọc giá trị SV sang thiết bị BCS2 theo giao thức truyền thông modbus trên dây RS485. Gói tin trả về từ BCS2 được xử lý và giá trị nhiệt độ theo độ Celsius và hiển thị trên Web. Giá trị này được hiển thị theo thời gian thực, thời gian thay đổi giá trị mỗi 1s.



Hình 3.18 Chức năng hiển thị giá trị PV

### 3.5 Kết quả, đánh giá

#### 3.5.1. Kết quả

Module truyền thông IoT công nghiệp cho thiết bị Shinko WCL-13A đáp ứng được yêu cầu đề tài của công ty.

Thông số thiết bị:

- Điện áp đầu vào: 100V ~ 240VAC / 50~60Hz
- Kích thước: 155x110x60mm
- Kết nối MODBUS-RTU theo chuẩn giao tiếp RS-485
- Khe cắm thẻ nhớ SD Card dung lượng tối đa 32GB, Class 10
- Kết nối Internet qua LAN/Wifi.
- Hỗ trợ kết nối module 4G.

##### 3.5.1.1. Phần cứng

- Sản phẩm hoàn thiện dạng prototype dựng trong hộp PLC, đảm bảo tính thẩm mỹ.
- Chân Domino và cổng Lan được đưa ra ngoài, dễ dàng kết nối.
- Đèn báo nguồn sáng khi kết nối nguồn cho thiết bị.
- Nút nhấn điều khiển kết hợp chống nhiễu phần cứng và phần mềm, không có hiện tượng dôi khi nhấn.
- Màn hình LCD đặt phía ngoài hộp, dễ dàng quan sát.
- Domino cắm nguồn đặt khác phía với domino kết nối, đảm bảo an toàn khi sử dụng.



Hình 3.19 Sản phẩm hoàn thiện

Kết nối phân cứng giữa Module IoT và thiết bị BCS2:

- Kết nối dễ dàng qua giao thức vật lý RS-485
- Tín hiệu truyền nhận ổn định, không bị lỗi gói tin khi truyền thử nghiệm ở tốc độ cao 3800 bps.



Hình 3.20 Kết nối Module với BCS2

#### 3.5.1.2. Phần mềm

- Chức năng lựa chọn kết nối ban đầu khi khởi động Module hoạt động ổn định, địa chỉ IP của module sẽ được hiển thị lên màn hình LCD khi kết nối thành công.
- Chức năng truy cập qua tên miền hoạt động tốt trên máy tính và trên các thiết bị điện thoại hỗ trợ Multicast DNS.
- Trạng thái hoạt động của các thiết bị ngoại vi được cập nhập trong suốt quá trình Module hoạt động.
- Giao diện Web dễ sử dụng, các chức năng hoạt động ổn định.

#### 3.5.2. Đánh giá

##### 3.5.2.1. Ưu điểm

- Mạch điều khiển nhỏ gọn, hoạt động ổn định, thời gian đáp ứng nhanh.
- Thiết bị dễ lắp đặt và triển khai, không yêu cầu về mặt công nghệ.
- Giao diện điều khiển và giám sát dễ sử dụng, thân thiện người dùng.
- Mô hình hệ thống có tính thẩm mỹ cao, độ chính xác, tính an toàn và dễ dàng thao tác với người dùng.

#### 3.5.2.2. Nhược điểm

- Sản phẩm còn ở dạng prototype, sử dụng nhiều module tích hợp sẵn nên không tối ưu hoá được thiết kế cũng như kích thước mạch.
- Mạch thủ công 1 lớp tính thẩm mỹ không cao.
- Web server nhúng trên chip ESP32 chỉ đáp ứng được số lượng có hạn người truy cập web server.
- Các chức năng cấu hình, đọc ghi dữ liệu còn đơn giản, chưa cấu hình sâu được cho thiết bị.

#### 3.5.2.3. Hướng phát triển của đề tài.

- Hoàn thiện về phần cứng sản phẩm, thiết kế mạch in công nghiệp, sử dụng hộp in 3D.
- Bổ sung chức năng cho chương trình để có thể sử dụng với nhiều dòng sản phẩm hỗ trợ chuẩn truyền thông Modbus.
- Tích hợp dịch vụ truy cập từ xa mà không cần sử dụng qua mạng nội bộ.

## TÀI LIỆU THAM KHẢO

### Tiếng anh

- [1] Shinko Technos Co., LTD , BCx2 Instruction Manual
- [2] Shinko Technos Co., LTD , BCx2 Communication Instruction Manual
- [3] Datasheet [ESP32](#),
- [4] Schematic [SIM7600E-H-4G-HAT](#)

### Internet

- [1] [www.shinko.vn](http://www.shinko.vn)
- [2] <http://www.shinko-technos.co.jp>
- [3] <https://randomnerdtutorials.com/projects-esp32/>