

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG



HCMUTE

BÁO CÁO ĐỒ ÁN

MÔN HỌC: ĐỒ ÁN MÔN HỌC 2

ĐỀ TÀI:

**THIẾT KẾ VÀ ỨNG DỤNG HỆ THỐNG FIRMWARE OVER-
THE-AIR (FOTA) CHO PHÒNG LAB TỪ XA**

GVHD: TS. Đỗ Duy Tân

SVTH:

- 1. Ngô Trọng Nghĩa - 21161155**
- 2. Hồng Lý Trung Nhân - 21119109**

TP.HCM, tháng 12 năm 2024

LỜI CẢM ƠN

Để hoàn thành được đồ án môn học này, chúng em xin chân thành cảm ơn các thầy, cô ở khoa Điện - Điện tử ở trường đại học Sư Phạm Kỹ Thuật TP.HCM nói chung và bộ môn Kỹ thuật Máy tính – Viễn thông nói riêng vì đã hỗ trợ nhóm trong việc trang bị các kiến thức nền tảng và các trang thiết bị trong suốt quá trình học tập, nghiên cứu vừa qua. Đặc biệt, chúng em xin chân thành cảm ơn sự hướng dẫn nhiệt tình của thầy TS. Đỗ Duy Tân cùng với những lời góp ý, giải đáp thắc mắc và đưa ra những nhận xét giúp chúng em có cái nhìn đúng hơn về kiến thức chuyên ngành. Qua những điều đó, cho nhóm được những kiến thức, kinh nghiệm trong việc hoàn thành đồ án này.

Chúng em xin chân thành cảm ơn!

MỤC LỤC

CHƯƠNG 1	1
TỔNG QUAN ĐỀ TÀI.....	1
1.1. TỔNG QUAN.....	1
1.2. LÝ DO CHỌN ĐỀ TÀI	1
1.3. MỤC TIÊU ĐỀ TÀI	2
1.4. CÁC NGHIÊN CỨU TRƯỚC ĐÓ.....	2
1.5. GIỚI HẠN ĐỀ TÀI	2
1.6. BỐ CỤC BÁO CÁO.....	3
CHƯƠNG 2.....	4
CƠ SỞ LÝ THUYẾT	4
2.1. TỔNG QUAN VỀ FOTA	4
2.2. GIỚI THIỆU VỀ GIAO THỨC HTTP.....	5
2.3. GIỚI THIỆU VỀ BOOTLOADER.....	6
2.4. GIỚI THIỆU PHẦN CỨNG SỬ DỤNG.....	8
2.4.1. STM32F103C8T6	8
2.4.2. ESP32-CAM	9
2.4.3. LCD 16x2 và Module I2C PCF8574T.....	10
2.4.4. Quang trở.....	12
2.4.5. Module led ma trận 8x8 với IC MAX7219.....	13
2.4.6. Led 7 đoạn và IC 74HC139	14
2.4.7. Led đơn và IC 74HC245	15
2.4.8. Cảm biến nhiệt độ, độ ẩm DHT11	16
2.5. GIỚI THIỆU PHẦN MỀM SỬ DỤNG.....	17
2.5.1. STM32CUBEMX	17
2.5.2. Microsoft Visual Studio Code.....	18
2.5.3. KeilC	18
2.5.4. Arduino IDE	19
2.5.5. EasyEDA.....	20
2.5.6. Django Framework.....	21

2.5.7. Microsoft SQL Server	22
CHƯƠNG 3	25
THIẾT KẾ HỆ THỐNG	25
3.1. YÊU CẦU HỆ THỐNG.....	25
3.2. ĐẶC TẢ HỆ THỐNG.....	26
3.1.1. Chức năng hệ thống	26
3.1.2. Mô hình tổng quát của hệ thống.....	27
3.1.3. Sơ đồ khối và nguyên lý hoạt động của hệ thống	28
3.3. THIẾT KẾ PHẦN CỨNG	29
3.4. THIẾT KẾ PHẦN MỀM	32
3.3.1. Chương trình trên bootloader	32
3.3.2. Chương trình cho ESP32-CAM	36
3.3.3. Giao diện và chức năng trên Website.....	37
CHƯƠNG 4.....	39
KẾT QUẢ VÀ ĐÁNH GIÁ	39
4.1. KẾT QUẢ	39
4.2. ĐÁNH GIÁ.....	42
4.2.1. Giới thiệu các trường hợp kiểm chứng.	42
4.2.2. Kiểm chứng trường hợp dung lượng file từ 10KB trở xuống.....	42
4.2.3. Kiểm chứng trường hợp dung lượng file từ 20KB trở xuống.....	44
CHƯƠNG 5.....	47
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	47
5.1. KẾT LUẬN.....	47
5.2. HƯỚNG PHÁT TRIỂN.....	47
TÀI LIỆU THAM KHẢO	49

DANH MỤC HÌNH ẢNH

Hình 2.1: Sơ đồ mô tả FOTA cho ứng dụng cập nhật từ xa.	4
Hình 2.2: Sơ đồ mô tả giao tiếp giữa Server và Client thông qua giao thức HTTP.	6
Hình 2.3: Quy trình tổng quát thực hiện chương trình của vi điều khiển.	7
Hình 2.4: IC STM32F103C8T6.	8
Hình 2.5: kit phát triển ESP32-CAM.	9
Hình 2.6: Sơ đồ chân của ESP32-CAM.	9
Hình 2.7: LCD 16x2.	11
Hình 2.8: Sơ đồ ra chân của LCD 16x2.	11
Hình 2.9: Module mở rộng I/O I2C PCF8574T.	12
Hình 2.10: Quang trở.	13
Hình 2.11: Module led ma trận 8x8 với IC MAX 7219.	13
Hình 2.12: Thông số kỹ thuật của module LED ma trận 8x8.	14
Hình 2.13: Led 7 đoạn.	14
Hình 2.14: IC 74HC139D.	15
Hình 2.15: Sơ đồ chân IC 74HC245.	15
Hình 2.16: Cảm biến nhiệt độ, độ ẩm DHT11.	16
Hình 2.17: Giao diện của STM32CUBEMX.	17
Hình 2.18: Giao diện của Visual Studio Code.	18
Hình 2.19: Giao diện KeilC.	19
Hình 2.20: Logo Arduino IDE.	20
Hình 2.21: Giao diện chính của EasyEDA.	21
Hình 2.22: Django Framework.	21
Hình 2.23: Cấu trúc của SQL server.	23
Hình 3.1: Sơ đồ tổng quát hệ thống.	27
Hình 3.2: Sơ đồ khối hệ thống.	28
Hình 3.3: Sơ đồ tổng quát cho Kit phát triển.	29
Hình 3.4: Sơ đồ nguyên lý của kit phát triển.	30
Hình 3.5: Sơ đồ layout mạch PCB của kit phát triển.	31
Hình 3.6: Lưu đồ hoạt động kiểm tra xem có đang ở bootloader hay không.	32
Hình 3.7: Lưu đồ hoạt động cho việc xóa vùng nhớ chương trình.	33
Hình 3.8: Lưu đồ hoạt động ghi dữ liệu vào vùng nhớ chương trình.	35
Hình 3.9: Lưu đồ hoạt động của ESP32-CAM.	36
Hình 3.10: Lưu đồ hoạt động của website.	38
Hình 4.1: Sơ đồ thực tế mạch kit học tập phát triển.	39
Hình 4.2: Giao diện phần đăng nhập trên website.	40
Hình 4.3: Giao diện phần chọn thiết bị.	40
Hình 4.4: Giao diện chính tương tác với thiết bị.	41

Hình 4.5: Thông tin về dung lượng file hex dưới 10KB.	43
Hình 4.6: Kiểm chứng thời gian lần cập nhật thứ 1.	43
Hình 4.7: Kiểm chứng thời gian lần cập nhật thứ 2.	43
Hình 4.8: Kiểm chứng thời gian lần cập nhật thứ 3.	43
Hình 4.9: Kiểm chứng thời gian lần cập nhật thứ 4.	44
Hình 4.10: Kiểm chứng thời gian lần cập nhật thứ 5.	44
Hình 4.11: Thông tin về dung lượng file hex dưới 20KB.	45
Hình 4.12: Kiểm chứng thời gian lần cập nhật thứ 1.	45
Hình 4.13: Kiểm chứng thời gian lần cập nhật thứ 2.	45
Hình 4.14: Kiểm chứng thời gian lần cập nhật thứ 3.	45
Hình 4.15: Kiểm chứng thời gian lần cập nhật thứ 4.	45
Hình 4.16: Kiểm chứng thời gian lần cập nhật thứ 5.	45
Hình 4.17: Biểu đồ so sánh thời gian thực thi qua 5 lần thử nghiệm giữa file 10KB và file 20KB.	46

DANH MỤC BẢNG BIỂU

Bảng 2.1: Bảng thông số kỹ thuật của STM32F103C8T6.	8
Bảng 2.2: Bảng thông số kỹ thuật của module mở rộng I/O PCF8754T.....	12
Bảng 2.3: Bảng thông số kỹ thuật của quang trở.....	13
Bảng 2.4: Bảng thông số kỹ thuật của IC 74HC139.	15
Bảng 2.5: Bảng thông số kỹ thuật của IC 74HC245.	16
Bảng 2.6: Bảng thông số kỹ thuật của cảm biến nhiệt độ, độ ẩm DHT11.	16

DANH SÁCH CÁC TỪ VIẾT TẮT

ACK	Acknowledgement
ADC	Analog-to-Digital Converter
AP	Access Point
API	Application Programming Interface
BLE	Bluetooth Low Energy
CAN	Controller Area Network
CSS	Cascading Style Sheets
DAC	Digital-to-Analog Converter
DC	Direct Current
FOTA	Firmware Over The Air
FPGA	Field-Programmable Gate Array
GCC	GNU Compiler Collection
GPIO	General-Purpose Input/Output
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
MAC	Media Access Control
MCU	Microcontroller Unit
NAK	Negative Acknowledgement
NTC	Negative Temperature Coefficient

ODBC	Open Database Connectivity
OE	Output Enable
OLED	Organic Light-Emitting Diode
PCB	Printed Circuit Board
PDF	Portable Document Format
PNG	Portable Network Graphics
PWM	Pulse-Width Modulation
QR	Quick Response
RTOS	Real Time Operating System
SoC	System on Chip
SPI	Serial Peripheral Interface
STA	Spanning Tree Algorithm
SVG	Scalable Vector Graphics
SWIO	Serial Wire Input Output
TDS	Tabular Data Stream
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus

CHƯƠNG 1

TỔNG QUAN ĐỀ TÀI

1.1. TỔNG QUAN

Trong bối cảnh các thiết bị nhúng đang phát triển không ngừng nói chung cũng như các vi xử lý, vi điều khiển nói riêng và cũng như mọi vật đều có thể kết nối internet, kèm với những tính năng hiện đại và được ứng dụng ở hầu hết các lĩnh vực trong đời sống hiện nay. Thiết bị nhúng luôn hiện diện xung quanh mỗi con người, từ những thứ đơn giản cho đến những thứ phức tạp, từ kích cỡ vô cùng nhỏ cho đến rất lớn. Cho nên các xu hướng chính cho thiết bị nhúng là để phục vụ một hoặc nhiều việc gì đó và được sử dụng rộng rãi ở trong công nghiệp, nông nghiệp, dân dụng, và cả công cụ cũng như là phương tiện di chuyển như xe hơi, xe máy. Việc cập nhật thêm tính năng, giao diện với mục đích là phát triển những sản phẩm đã bán ra thị trường. Nhà sản xuất luôn luôn tìm cách nâng tầm thương hiệu của họ bằng cách nâng cấp sản phẩm đó ở một vài năm với mục đích giữ chân khách hàng. Để làm được điều đó, họ không thể nào lấy lại sản phẩm từ người dùng và nâng cấp chúng được, từ đó công nghệ cập nhật chương trình từ xa (FOTA) đã ra đời. Với tình hình công nghệ luôn luôn phát triển, thì chính nơi giúp cho công nghệ này phát triển cũng phải đi lên, chính các trường đại học, cao đẳng, hay các trung tâm đào tạo về lĩnh vực nhúng cũng như vi điều khiển. Đề tài này sẽ tập trung làm sao có thể được tìm hiểu, nghiên cứu, học tập các hệ thống nhúng ở xa.

1.2. LÝ DO CHỌN ĐỀ TÀI

Đề tài này hướng đến nhóm đối tượng là sinh viên các ngành về kỹ thuật điện tử, có các môn học liên quan về lập trình vi điều khiển cũng như là hệ thống nhúng và nó mang lại cho người dùng sự tiện lợi về mặt giáo dục cũng như là chi phí có thể để tự mua cho mình công cụ học tập. Thay vì phải mất nhiều chi phí cho việc học tập cũng như nghiên cứu, thì sinh viên có thể sử dụng hệ thống này như là nơi để các bạn có thể nghiên cứu thêm sau giờ học. Ngoài ra, hệ thống này còn phù

hợp cho các tình huống khó khăn khi cần phải lập trình trực tiếp khi có dụng cụ chẳng hạn như là dịch bệnh, thiên tai,...

1.3. MỤC TIÊU ĐỀ TÀI

Đối với đề tài “**THIẾT KẾ VÀ ỨNG DỤNG HỆ THỐNG FIRMWARE OVER-THE-AIR (FOTA) CHO PHÒNG LAB TỪ XA**” sẽ có những mục tiêu sau:

- Mô hình kit phát triển sử dụng vi điều khiển STM32F103C8T6 cũng với các ngoại vi.
- Trang web với mục đích đưa file hex cũng như quản lý người dùng và trạng thái của kit thông qua camera của ESP32-CAM.
- Chức năng nạp code từ xa cho vi điều khiển STM32F103C8T6 bằng ESP32-CAM.
- Xử lý trường hợp có nhiều hơn 1 người dùng cùng truy xuất vào 1 bộ kit.

1.4. CÁC NGHIÊN CỨU TRƯỚC ĐÓ

Trước đó cũng đã có một bài báo nghiên cứu có đề tài “**CẬP NHẬT CHƯƠNG TRÌNH TỪ XA CHO THIẾT BỊ NHÚNG CÓ INTERNET**”[] của nhóm tác giả đến từ Đại học Bách Khoa Đà Nẵng. Đề tài này đang hướng đến cập nhật chương trình từ xa cho một bộ datalogger để với mục đích là để phát triển hệ thống này từ xa nếu như có lỗi về chương trình của bộ datalogger này. Kèm theo đó là phương pháp để triển khai hệ thống có sử dụng FOTA và kiểm tra thời gian cập nhật đối với những tệp tin thực thi có nhiều dung lượng khác nhau. Vì vậy, sau khi tìm hiểu các đề tài có liên quan trên, nhận ra rằng là có quá ít đề tài sẽ tập trung về mặt giáo dục cho nên sẽ phát triển một đề tài liên quan đến giáo dục.

1.5. GIỚI HẠN ĐỀ TÀI

Đề tài này sẽ tập trung làm thế nào để quản lý 1 kit phát triển cũng như lập trình từ xa cho nên để đảm bảo có thể đáp ứng về mặt thời gian cũng như những mặt khác thì đề tài này được giới hạn ở những khía cạnh như sau:

- Website sẽ tập trung về mặt tính năng cũng như hoạt động bên trong, cho nên sẽ không tập trung về mặt giao diện cũng như hiệu ứng tương tác trên website.
- Kit phát triển cũng sẽ không có quá nhiều module đặc biệt sẽ không có các module tương tác trực tiếp như là nút nhấn, biến trở cũng như các module cần

hiều về khả năng tương tác trực tiếp. Các tính năng này sẽ được nghiên cứu, hiện thực hóa nếu đề tài này còn nhiều hướng phát triển sau này.

1.6. BỐ CỤC BÁO CÁO

Báo cáo được phân bổ thành năm chương như sau:

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.

Trình bày tổng quan, lý do chọn đề tài, mục tiêu của đề tài này, liệt kê ra những nghiên cứu trước đó, giới hạn, và bố cục báo cáo.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Trình bày các lý thuyết, các phương pháp để thực hiện đề tài.

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

Đặc tả hệ thống, làm rõ các chức năng, thiết kế và thi công sản phẩm.

CHƯƠNG 4: KẾT QUẢ VÀ ĐÁNH GIÁ

Trình bày kết quả đạt được và các trường hợp kiểm thử của hệ thống nhằm đánh giá khả năng hoạt động của hệ thống.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

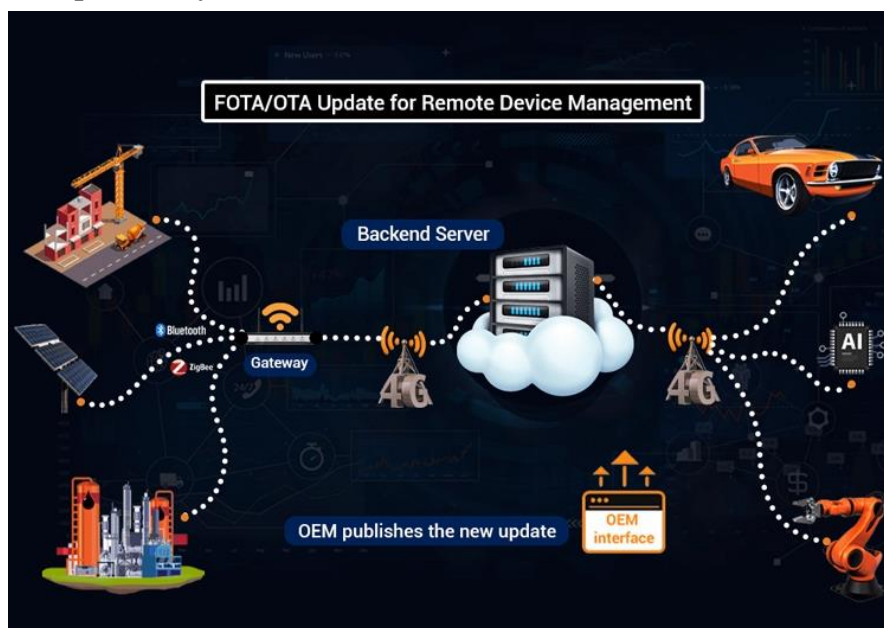
Dựa vào Chương 4, từ đó rút ra kết luận và đưa ra các hướng phát triển tiếp theo.

CHƯƠNG 2

CƠ SỞ LÝ THUYẾT

2.1. TỔNG QUAN VỀ FOTA

FOTA (Firmware Over-The-Air) là một công nghệ quan trọng trong việc cập nhật phần mềm từ xa cho các thiết bị điện tử, từ điện thoại di động đến các thiết bị IoT. Công nghệ này giúp cải thiện hiệu suất của thiết bị mà không cần phải kết nối trực tiếp với máy tính hoặc thiết bị khác.



Hình 2.1: Sơ đồ mô tả FOTA cho ứng dụng cập nhật từ xa[1].

Một ưu điểm quan trọng khác của công nghệ FOTA là khả năng nâng cao tính ổn định và bảo mật của các thiết bị điện tử. Việc cập nhật firmware từ xa cho phép nhanh chóng khắc phục các lỗ hổng bảo mật mới, bảo vệ dữ liệu cá nhân và thông tin của người dùng. Bên cạnh đó, các bản cập nhật này cũng có thể tăng cường hiệu suất và tích hợp thêm các tính năng mới, mang lại trải nghiệm tối ưu hơn cho người sử dụng.

***Cách Thức Hoạt Động của FOTA**

Tải xuống bản cập nhật: Thiết bị sẽ tự động kiểm tra và tải về bản cập nhật phần mềm từ máy chủ của nhà sản xuất.

Cài đặt bản cập nhật: Khi quá trình tải xuống hoàn tất, thiết bị sẽ tiến hành cài đặt bản cập nhật, thường yêu cầu khởi động lại thiết bị.

Kiểm tra và khắc phục lỗi: Thiết bị sẽ tự động kiểm tra để bảo đảm bản cập nhật đã được cài đặt đúng cách và không gặp lỗi.

***Thách Thức của FOTA**

Kết nối mạng ổn định: Để tải và cài đặt bản cập nhật, thiết bị cần có kết nối mạng ổn định.

Dung lượng lưu trữ: Thiết bị cần đủ dung lượng trống để lưu trữ bản cập nhật.

Rủi ro lỗi trong quá trình cập nhật: Quá trình cập nhật có thể gặp lỗi, gây ra các vấn đề cho thiết bị.

***Ứng Dụng của FOTA**

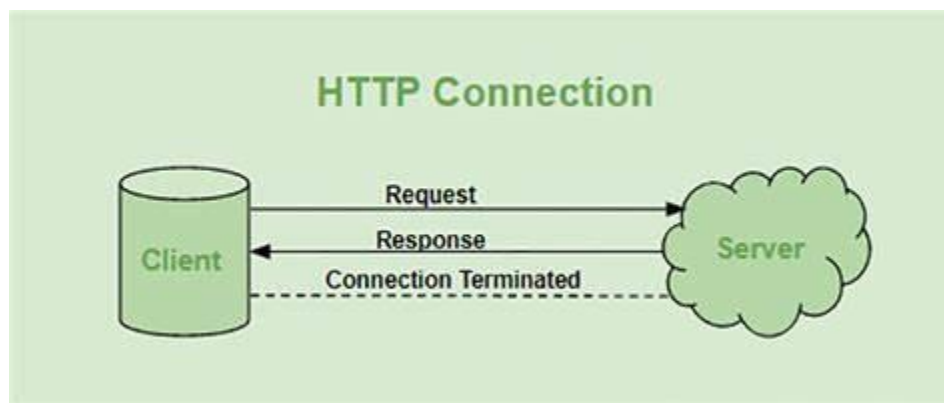
FOTA được sử dụng rộng rãi trong nhiều loại thiết bị điện tử, từ điện thoại di động, máy tính bảng đến các thiết bị IoT như camera an ninh và thiết bị gia dụng thông minh. Công nghệ này không chỉ giúp cải thiện hiệu suất và bảo mật, mà còn nâng cao trải nghiệm người dùng bằng cách cung cấp các tính năng mới và cập nhật bảo mật kịp thời. FOTA không chỉ là một phương thức cập nhật tiện lợi mà còn đóng vai trò quan trọng trong việc duy trì và cải thiện hiệu suất của các thiết bị điện tử hiện đại.

2.2. GIỚI THIỆU VỀ GIAO THỨC HTTP

Giao thức HTTP hay còn gọi là giao thức truyền tải siêu văn bản là một giao thức truyền tải dữ liệu ở tầng ứng dụng, đây là nền tảng phục vụ trao đổi và liên lạc dữ liệu thông qua World Wide Web, đây là một nơi chứa các tệp tin hoặc các tài nguyên số khác mà người dùng có thể truy cập thông qua các tương tác trên ứng dụng như chạm, kéo thả,... Vì vậy, người dùng có thể cho phép các tài nguyên số như văn bản, hình ảnh của các trang web khác thông qua HTTP và hiển thị chúng lên trên trình duyệt của người dùng.

Về hoạt động của HTTP được dựa trên mô hình Server-Client. Client ở đây có thể là một trình duyệt nào đó được một người thực hiện yêu cầu truy cập, còn Server ở đây có thể hiểu là một máy chủ trên đám mây hoặc một máy lưu trữ dữ liệu cục bộ đang chạy trên một chiếc máy tính khác. HTTP được sinh ra là để đảm bảo các thành phần mạng trung gian cho phép liên lạc giữa Server và Clients.

Giao thức HTTP sử dụng các phương thức như GET và POST, sau khi sử dụng một trong các phương thức đó, thì Server bắt buộc phải gửi lại cho Client giá trị trạng thái, chẳng hạn như nếu GET thành công thì giá trị trạng thái trả về là 200, trong trường hợp GET hoặc POST bị thất bại do không tìm thấy file hay gì đó chẳng hạn, thì giá trị trạng thái trả về sẽ là 404. Ngoài ra còn rất nhiều giá trị trả về khác và nó mang mỗi một ý nghĩa riêng [2].



Hình 2.2: Sơ đồ mô tả giao tiếp giữa Server và Client thông qua giao thức HTTP.

2.3. GIỚI THIỆU VỀ BOOTLOADER

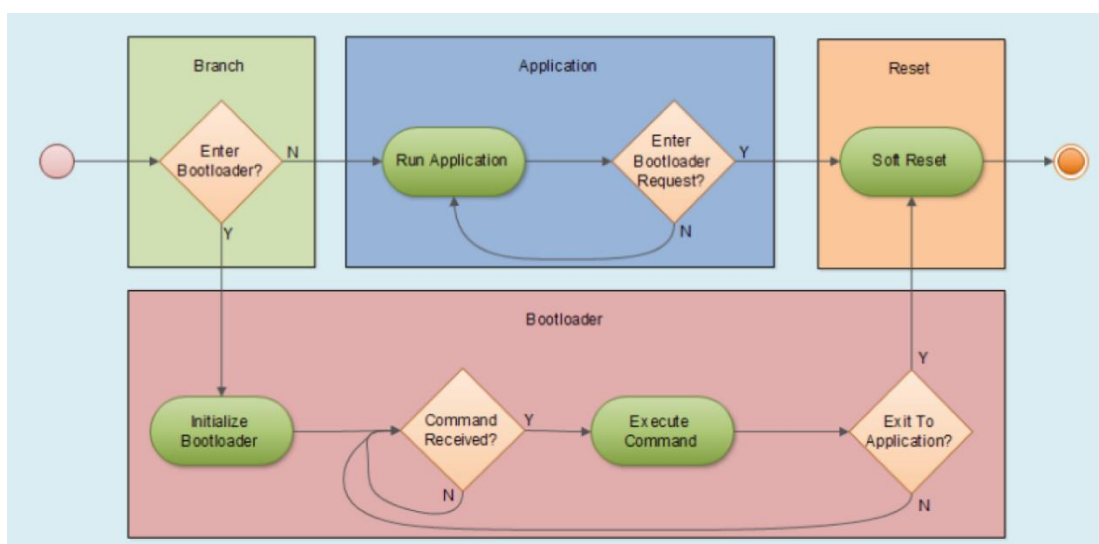
Bootloader là một chương trình khi cấp nguồn sẽ được thực thi chương trình đó, chương trình này sẽ tương tác với phần cứng, cụ thể là bộ nhớ. Chức năng của bootloader trong vi điều khiển thường được ứng dụng trong cập nhật hoặc thay đổi hệ thống.

Khác với chương trình ứng dụng thông thường, trường hợp vi điều khiển đó không được nạp bootloader thì sẽ nhảy vào chương trình ứng dụng chạy đầu tiên. Ngược lại, nếu như đã có bootloader thì con trỏ chương trình sẽ nhảy vào vùng nhớ của bootloader thực thi đầu tiên, sau khi thực thi xong thì con trỏ chương trình sẽ nhảy vào vùng nhớ chứa chương trình cho ứng dụng

hoặc cũng có thể nhảy đến chương trình ứng dụng khi đang thực thi bootloader, điều đó phụ thuộc vào chương trình bootloader được viết như thế nào. Đối với vi điều khiển, thay vì có thể nạp chương trình theo chuẩn nạp thông thường mà nhà sản xuất đưa ra thì có thể cấu hình lại bootloader để nạp chương trình theo chuẩn truyền thông khác, có thể là UART, I2C, SPI, CAN,...Tuy nhiên, nhà sản xuất cũng đã có một bootloader có sẵn cũng có thể nạp bằng chuẩn truyền thông như đã đề cập bên trên và bootloader này nằm ở một vùng nhớ được gọi là Factory Reset hoặc gọi là Factory Recovery, đây là nơi chứa những chương trình được nạp sẵn trước khi được phát hành và thương mại. Vì thế, nếu không muốn thì người dùng hoàn toàn có thể tự cấu hình một bootloader cho riêng mình nhưng sẽ phải sử dụng một vùng nhớ khác.

*Một số ứng dụng của bootloader:

- Cập nhật Firmware.
- Chuyển đổi chương trình ứng dụng.
- Một số ứng dụng khác.

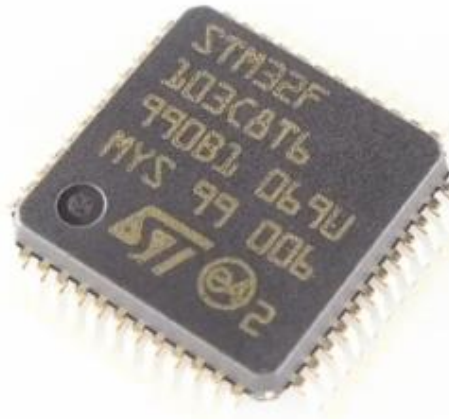


Hình 2.3: Quy trình tổng quát thực hiện chương trình của vi điều khiển[3].

2.4. GIỚI THIỆU PHẦN CỨNG SỬ DỤNG

2.4.1. STM32F103C8T6

STM32F103C8T6 là một vi điều khiển có kiến trúc ARM Cortex M3 32-bit, đây là dòng chip phổ thông của ST microelectronics, cùng với các ngoại vi cơ bản đầy đủ, có kích thước bộ nhớ phù hợp cùng với giá thành rẻ. Phù hợp cho người mới bắt đầu.



Hình 2.4: IC STM32F103C8T6.

Thông số kỹ thuật của IC STM32F103C8T6 [4]:

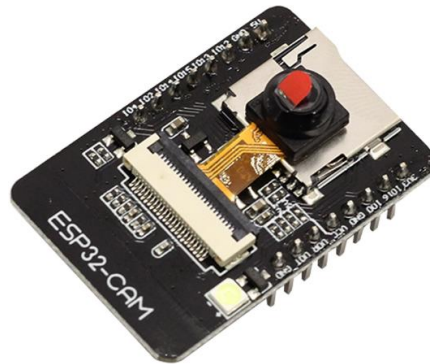
Bảng 2.1: Bảng thông số kỹ thuật của STM32F103C8T6.

Nguồn	2.7V – 5.5V
Input/Output	35
Số chân	48
Core	ARM 32 Cortex-M3 CPU
Dòng hoạt động	1.1mA
Dòng chờ	1uA
Dòng output	20mA
Thạch anh thiết lập	72Mhz
Thạch anh ngoài	4Mhz – 16Mhz
Các chuẩn giao tiếp	I ² C, SPI, UART / USART, USB, CAN
Bộ nhớ ROM	128KByte
Bộ nhớ SRAM	20KByte
Nhiệt độ hoạt động	-40°C ~ 85°C

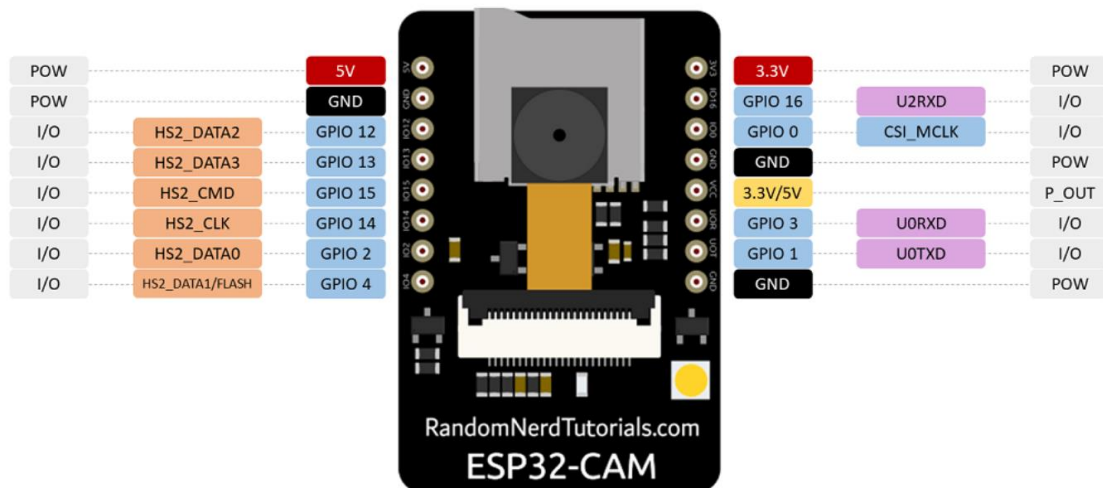
2.4.2. ESP32-CAM

ESP32-CAM là một module camera nhỏ gọn có khả năng hoạt động độc lập và kích thước chỉ 40 x 27 x 12 mm với dòng nghỉ chỉ 6mA. ESP32-CAM có thể ứng dụng rộng rãi trong IoT, phù hợp cho các thiết bị gia đình thông minh, điều khiển công nghiệp không dây, giám sát, nhận dạng QR và định vị không dây, là một giải pháp lý tưởng cho các ứng dụng IoT.

Module này được sản xuất từ hãng AI-Thinker, có độ ổn định và độ bền cao được trang bị camera OV2640 cho hình ảnh sắc nét và ổn định, hạn chế nhiễu và tình trạng treo khi hoạt động nhờ IC cấp nguồn chất lượng. ESP32-CAM Ai-Thinker hỗ trợ lập trình bằng Arduino IDE và được cộng đồng hỗ trợ mạnh mẽ.



Hình 2.5: kit phát triển ESP32-CAM.



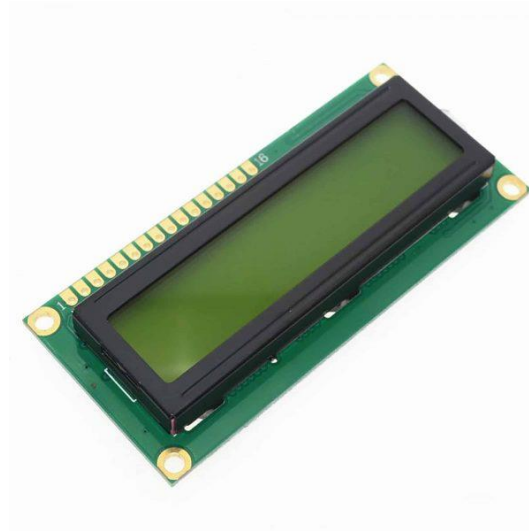
Hình 2.6: Sơ đồ chân của ESP32-CAM.

Thông số kỹ thuật: [5]

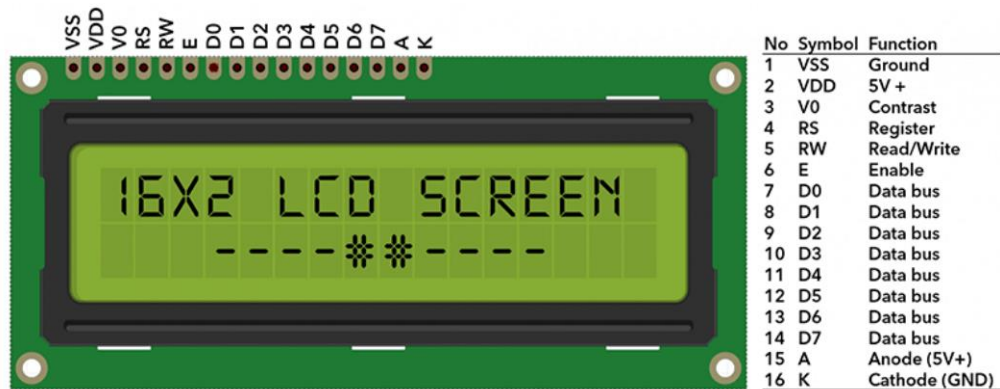
- IC chính: ESP32-S (AI-Thinker).
- Mô-đun Wi-Fi BT SoC 802.11 b/g/n/e/i.
- CPU 32-bit công suất thấp, cũng có thể phục vụ bộ xử lý ứng dụng.
- Tốc độ đồng hồ lên đến 160MHz, sức mạnh tính toán lên đến 600 DMIPS.
- Tích hợp 520 KB SRAM, 4MPSRAM bên ngoài.
- Dải tần số: 1421 ~ 2484 Mhz.
- Bluetooth: 4.2 BR/EDR BLE.
- Hỗ trợ UART / SPI / I2C / PWM / ADC / DAC.
- Hỗ trợ máy ảnh OV2640 và OV7670, đèn flash tích hợp.
- Hỗ trợ tải lên WiFi hình ảnh.
- Hỗ trợ thẻ TF.
- Hỗ trợ nhiều chế độ ngủ.
- Nhúng Lwip và FreeRTOS.
- Hỗ trợ chế độ hoạt động STA / AP / STA + AP.
- Hỗ trợ cấu hình thông minh / công nghệ AirKiss.
- Hỗ trợ nâng cấp cục bộ và từ xa cho công nối tiếp (FOTA).

2.4.3. LCD 16x2 và Module I2C PCF8574T

LCD 16x2 là một màn hình có 16 cột và 2 hàng và có thể hiển thị 32 ký tự chẳng hạn như là số, chữ, kí tự đặc biệt, có thể dễ dàng giao tiếp thông qua nhiều giao thức khác nhau cùng với giá thành vô cùng hợp lý.



Hình 2.7: LCD 16x2.



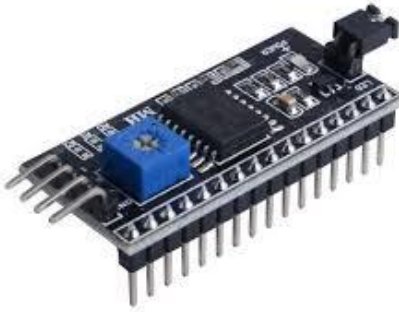
Hình 2.8: Sơ đồ ra chân của LCD 16x2.

Thông số kỹ thuật: [6]

- Điện áp hoạt động là 5V.
- Kích thước: 80 x 36 x 12.5mm.
- Chữ trắng, nền xanh dương.
- Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
- Tên các chân được ghi ở mặt sau của màn hình LCD hỗ trợ việc kết nối, đi dây điện.
- Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu.

Thay vì tốn nhiều chân hơn để giao tiếp với LCD 16x2 thì có giải pháp rút gọn số chân giao tiếp bằng module I2C.

Module I2C PCF8574T là một module mở rộng I/O chuyển đổi từ giao tiếp I2C thành 8-bit dữ liệu song song. Giúp mở rộng thêm chân cũng như đơn giản hóa quá trình thiết kế mạch.



Hình 2.9: Module mở rộng I/O I2C PCF8574T.

Thông số kỹ thuật của module mở rộng I/O PCF8574T [7]:

Bảng 2.2: Bảng thông số kỹ thuật của module mở rộng I/O PCF8574T.

Điện áp hoạt động	2.5-6VDC
Output	8 bit ngõ ra
Kích thước	41.5 x 19 mm
Giao tiếp	I2C, địa chỉ có thể chọn bằng điện trở A0, A1, A2 trên board

2.4.4. Quang trở

Quang trở hay còn gọi là quang điện trở là một linh kiện điện tử có khả năng thay đổi điện trở khi thay đổi cường độ chiếu sáng vào nó. Quang trở được ví như một tế bào quang điện hoạt động dựa trên nguyên lý quang dẫn. Quang trở được cấu tạo khá đơn giản bao gồm 2 bộ phận chính là phần chân và phần tiếp xúc ánh sáng:

- Phần chân là phân kim loại được liên kết với nhay thông qua các đầu cực.
- Phần tiếp xúc ánh sáng là phần được tiếp xúc trực tiếp với ánh sáng nhằm đọc được sự thay đổi về cường độ ánh sáng.



Hình 2.10: Quang trở.

Thông số kỹ thuật của quang trở[8]:

Bảng 2.3: Bảng thông số kỹ thuật của quang trở.

Điện áp tối đa	250VDC
Công suất	200mW
Giá trị đỉnh Spectrum	540 nm
Kháng ánh sáng	10 – 20k Ω
Kháng ban đêm	2M Ω
Nhiệt độ hoạt động	-30 - 70 °C
Giá trị c (1000 10)	0
Thời gian đáp ứng	Tăng: 30ms/ Giảm: 30ms

2.4.5. Module led ma trận 8x8 với IC MAX7219

Module led ma trận 8x8 là một module sử dụng led ma trận 8x8 bao gồm một ma trận led có 8 hàng và 8 cột kết hợp sử dụng IC MAX7219 để chuyển đổi từ giao thức SPI sang 2 giao thức 8-bit song song với mục đích giảm thiểu số chân giao tiếp.



Hình 2. 11: Module led ma trận 8x8 với IC MAX 7219.

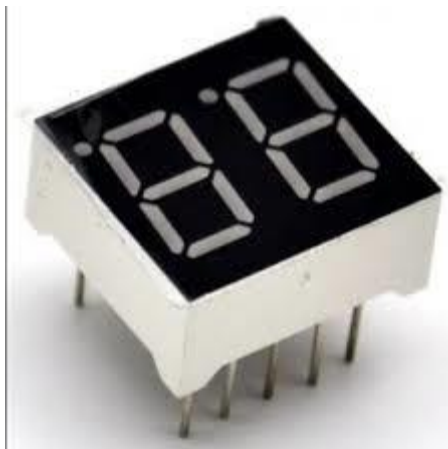
Thông số kỹ thuật của module LED ma trận 8x8 [9]:

Hình 2. 12: Thông số kỹ thuật của module LED ma trận 8x8.

Loại led	Anot chung
Điện áp hoạt động	4.7 ~ 5.3VDC
Dòng điện tiêu thụ	320mA
Nhiệt độ hoạt động	0 – 50°C
Kích thước	50x32mm

2.4.6. Led 7 đoạn và IC 74HC139

LED 7 đoạn là một loại LED để hiển thị chữ số, chữ cái thông qua những đoạn led, nó được cấu tạo từ 7 đoạn led được kí hiệu từ a đến g và cùng 1 dấu chấm để hiển thị chữ số thập phân.



Hình 2. 13: Led 7 đoạn.

Để có thể điều khiển được nhiều led 7 đoạn hơn, cần sử dụng kỹ thuật quét led, để giảm thiểu số chân quét nhiều nhất có thể, IC 74HC139 là một lựa chọn khi cần quét 4 led 7 đoạn liên tục.

IC 74HC139 là một IC gồm 2 bộ giải mã 2 sang 4, đầu vào là 2 bit nhị phân cùng với chân cho phép tích cực mức thấp và ngõ ra là 4 bit nhị phân. IC này có ngõ ra mức thấp cho nên phù hợp để điều khiển các thiết bị tích cực mức thấp.



Hình 2.14: IC 74HC139D.

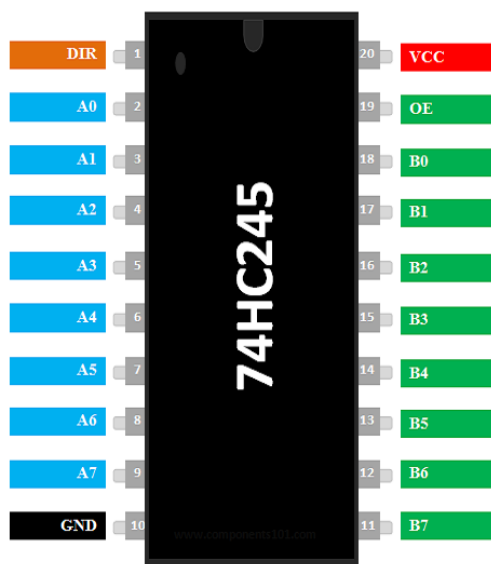
Thông số kỹ thuật của IC 74HC139D[10]:

Bảng 2.4: Bảng thông số kỹ thuật của IC 74HC139.

Điện áp hoạt động	2 - 7VDC
Dòng điện tiêu thụ	20mA
Nhiệt độ hoạt động	-40 – 85°C
Kích thước	13.1x10.3mm

2.4.7. Led đơn và IC 74HC245

IC 74HC245 là một IC thu phát 8-bit với đầu ra có ba trạng thái. Dữ liệu có thể được truyền theo hai chiều phụ thuộc vào chân điều khiển. Có thể kích hoạt thông qua chân cho phép OE.



Hình 2.15: Sơ đồ chân IC 74HC245.

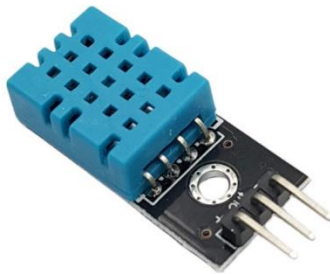
Thông số kỹ thuật của IC 74HC245 [11]:

Bảng 2.5: Bảng thông số kỹ thuật của IC 74HC245.

Điện áp hoạt động	2 - 6VDC
Dòng điện tiêu thụ	35mA
Nhiệt độ hoạt động	-40 – 85°C
Kích thước	13.1x10.3mm

2.4.8. Cảm biến nhiệt độ, độ ẩm DHT11

DHT11 là một cảm biến nhiệt độ và độ ẩm thường được sử dụng đi kèm với một NTC chuyên dụng để đo nhiệt độ và một bộ vi điều khiển 8 bit để xuất ra các giá trị nhiệt độ và độ ẩm dưới dạng dữ liệu nối tiếp.



Hình 2.16: Cảm biến nhiệt độ, độ ẩm DHT11.

Thông số kỹ thuật của cảm biến nhiệt độ, độ ẩm DHT11 [12]:

Bảng 2.6: Bảng thông số kỹ thuật của cảm biến nhiệt độ, độ ẩm DHT11.

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

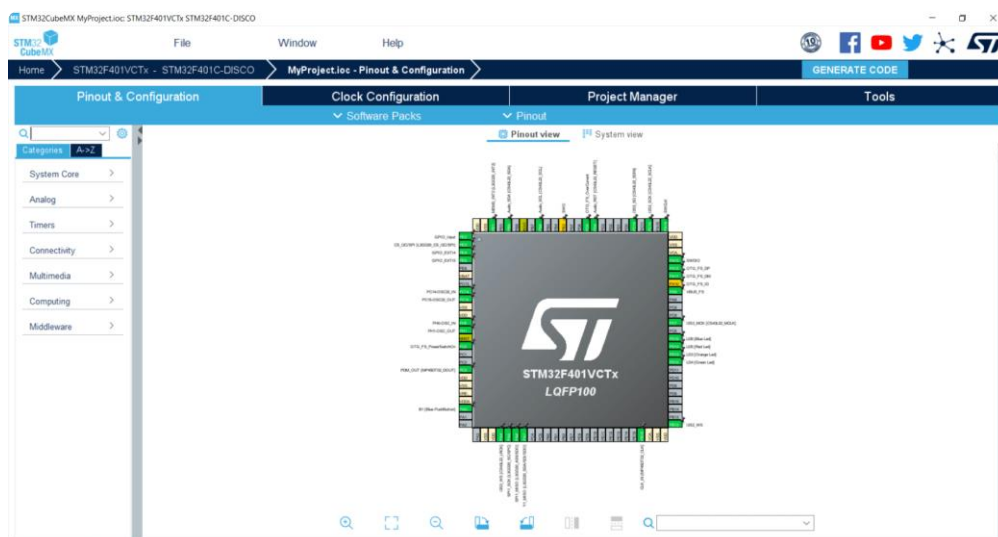
2.5. GIỚI THIỆU PHẦN MỀM SỬ DỤNG

2.5.1. STM32CUBEMX

STM32CubeMX là một công cụ giao diện cho phép người dùng có thể dễ dàng cấu hình các dòng vi điều khiển và vi xử lý STM32, đầu ra của công cụ này chính là tạo ra code với các cấu hình sẵn của vi điều khiển này, công việc sau đó của người dùng chính là lập trình chức năng bằng logic mà không cần quan ngại gì về các thanh ghi bên trong nó, ngoài ra còn có thể cung cấp các API có sẵn phục vụ cho các chức năng khác như FreeRTOS,...

Các chức năng chính của STM32CubeMx bao gồm những tính năng sau đây:

- Lựa chọn vi điều khiển và vi xử lý STM32 bằng tương tác giao diện.
- Giao diện người dùng dễ sử dụng và cho phép các chức năng sau đây:
 - Cấu hình sơ đồ chân và giải quyết xung đột chân một cách tự động.
 - Tạo ra mã ngôn ngữ C cùng với trình biên dịch GCC cho lõi ARM-Cortex-M.
 - Tích hợp các gói mở rộng vào dự án.
- Đa nền tảng hệ điều hành: Window, Linux, MacOS.



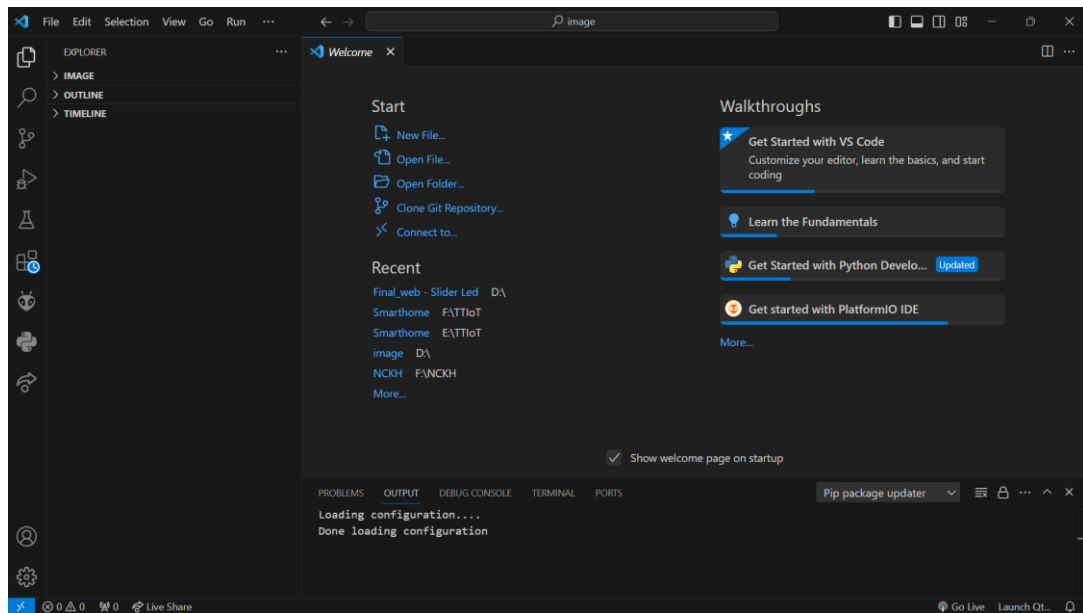
Hình 2. 17:Giao diện của STM32CUBEMX[13].

2.5.2. Microsoft Visual Studio Code

Visual Studio Code là một trình soạn thảo mã nguồn mở, đa nền tảng và nhẹ, được thiết kế để hỗ trợ lập trình viên trong việc viết và phát triển phần mềm một cách nhanh chóng và hiệu quả.

Các ưu điểm của Visual Studio Code:

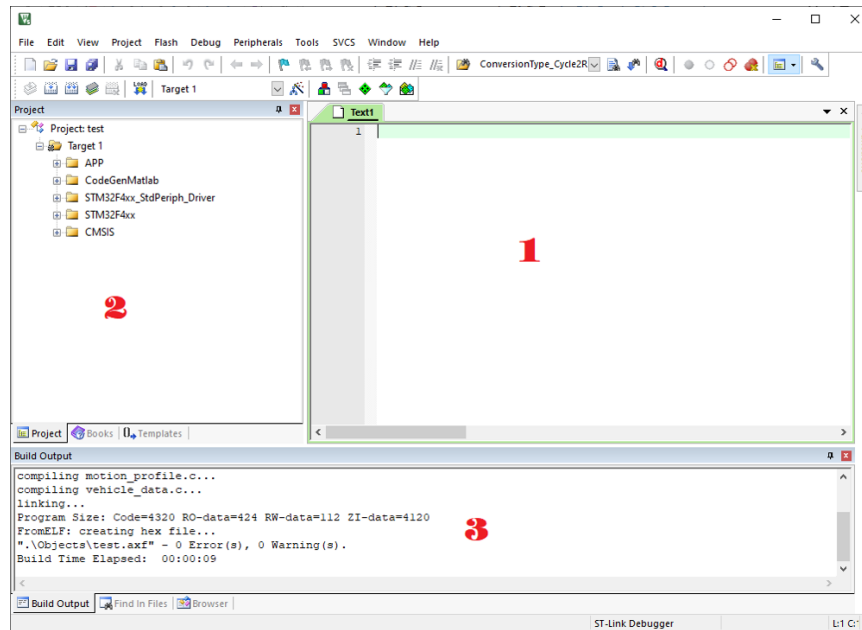
- Đa nền tảng hệ điều hành: Linux, Mac, Windows,...
- Hỗ trợ đa ngôn ngữ: C/C++, C#, F#, JavaScript, JSON, HTML, CSS,...
- Dung lượng thấp.
- Nhiều tính năng.
- Giao diện thân thiện, dễ sử dụng, có thể tùy biến theo người dùng[14].



Hình 2.18: Giao diện của Visual Studio Code.

2.5.3. KeilC

Hiện nay có rất nhiều phần mềm có tích hợp trình biên dịch của ngôn ngữ C dành cho các dòng vi điều khiển, đối với các vi điều khiển ARM như STM32 thường sẽ sử dụng các phần mềm này để có thể biên dịch ra file nhị phân, thập lục phân để có thể nạp lên các dòng vi điều khiển này. Chúng vừa đóng vai trò là một trình soạn thảo mã nguồn cho ngôn ngữ C và assembly, ngoài ra chúng còn có được tích hợp trình biên dịch cũng như tích hợp tích năng hỗ trợ gỡ lỗi.



Hình 2.19: Giao diện KeilC.

Cửa sổ làm việc chính của KeilC có 3 vùng làm việc chính như sau:

- Vùng 1: vùng này dùng để soạn thảo mã nguồn, cho phép lập trình viên có thể lập trình ở vùng này.
- Vùng 2: Vùng này giúp cho lập trình viên quản lý các tệp tin liên quan đến dự án, quản lý các chương trình con khác của dự án.
- Vùng 3: Vùng cửa sổ ngõ ra, cho phép hiển thị thông tin về quá trình biên dịch dự án. Sau khi biên dịch xong, thì cửa sổ này sẽ thể hiện trạng thái của dự án đó như là lỗi, không có lỗi, cảnh báo,...

2.5.4. Arduino IDE

Arduino IDE là một ứng dụng máy tính hỗ trợ nền tảng về lập trình Arduino hoặc các dòng vi điều khiển khác bằng cách sử dụng framework Arduino để lập trình.

Ngôn ngữ chính của nó được dựa trên ngôn ngữ C/C++ và có thể dễ dàng học và sử dụng, hiện nay đây là một trong những nền tảng có cộng đồng hỗ trợ mạnh mẽ nhất trong lĩnh vực điện tử cũng như là vi điều khiển, có thể dễ dàng tìm thấy các video cũng như là các website hướng dẫn sử dụng cũng như là lập trình cho Arduino từ cơ bản đến nâng cao.



Hình 2. 20: Logo Arduino IDE.

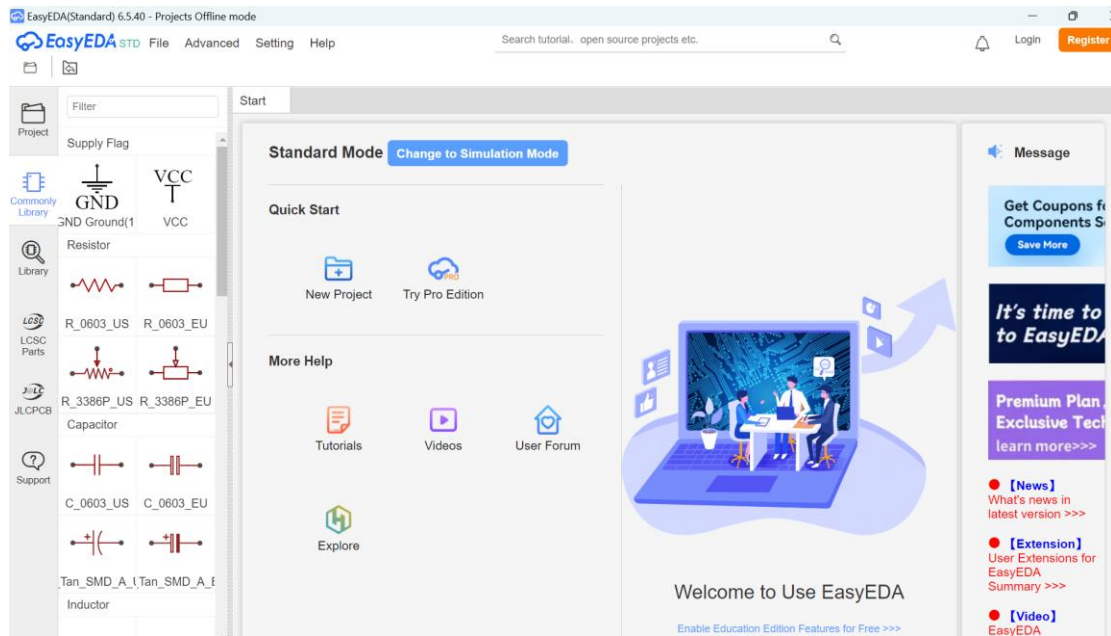
Arduino IDE được tích hợp cả tính năng biên dịch chương trình và hỗ trợ nạp file hex vào vi điều khiển để sử dụng, đây là một phần mềm mã nguồn mở cho nên có thể dùng tất cả tính năng của nó một cách miễn phí. Hiển nhiên là có cũng hỗ trợ cho các hệ điều hành phổ biến hiện nay như là Window, Linux, MacOS mà dung lượng không quá nhiều.

2.5.5. EasyEDA

EasyEDA là công cụ dựa trên web giúp hỗ trợ thiết kế, mô phỏng, chia sẻ và xem xét sơ đồ mạch và PCB. Phần mềm này miễn phí, dễ sử dụng dành cho các kỹ sư điện, phần cứng, nhúng... Người dùng chỉ cần đăng ký hoặc đăng nhập để tạo mạch trên mọi hệ điều hành như Windows, Mac, Linux và các trình duyệt như Chrome, Firefox, Safari, Internet Explorer. Dự án sẽ được lưu tự động trên trang web của EasyEDA, đảm bảo không bị mất dữ liệu.

EasyEDA được tích hợp nhiều tính năng hữu ích như:

- Thiết kế mạch dễ dàng và hiệu quả.
- Chia sẻ dự án ở dạng công khai hoặc riêng tư.
- Hỗ trợ phát triển dự án mã nguồn mở, tập lệnh và API.
- Xuất tài liệu dưới dạng PDF, SVG, PNG. Netlist sang FreePCB, Spice, Pads, Altium Designer; BOM/DXF.
- Truy cập thư viện công cộng với hơn 1 triệu mục và quản lý các thư viện khác nhau.
- Tạo, chỉnh sửa linh kiện, footprint và mô hình Spice.
- Kiểm tra quy tắc thiết kế, mô phỏng và chuyển đổi layout từ schematic sang PCB.



Hình 2.21: Giao diện chính của EasyEDA.

2.5.6. Django Framework

Django là một framework dành cho ngôn ngữ lập trình Python cho phép phát triển và duy trì websites. Cùng với một cộng đồng phát triển khá rộng, Django cho thấy rằng nó có thể đảm bảo được người dùng có thể tập trung phát triển thay vì phải xử lý lại các vấn đề liên quan nền tảng trước. Django là một nền tảng mã nguồn mở, cùng với nhiều tài liệu phong phú, cùng với rất nhiều sự lựa chọn về mặt hỗ trợ trên các cộng đồng[15].



Hình 2.22: Django Framework.

***Các đặc điểm của Django:**

- **Linh hoạt:** Django được dùng để phát triển cho nhiều loại website khác nhau từ hệ thống thông tin quản lý cho đến mạng xã hội hoặc là trang web về tin tức. Nó hỗ trợ linh hoạt ở nền tảng bên phía Clients và có thể xuất ra dữ liệu với nhiều định dạng khác nhau như HTML, Json,...

- **Bảo mật:** Django được thiết kế với mục đích để giảm rủi ro về bảo mật như SQL injection,... Tính năng này giúp người dùng có thể quản lý tài khoản cá nhân và mật khẩu mà không phải bận tâm về việc bảo mật như thế nào. Ngoài ra, Django giúp người dùng bảo vệ website khỏi các rủi ro thường gặp.

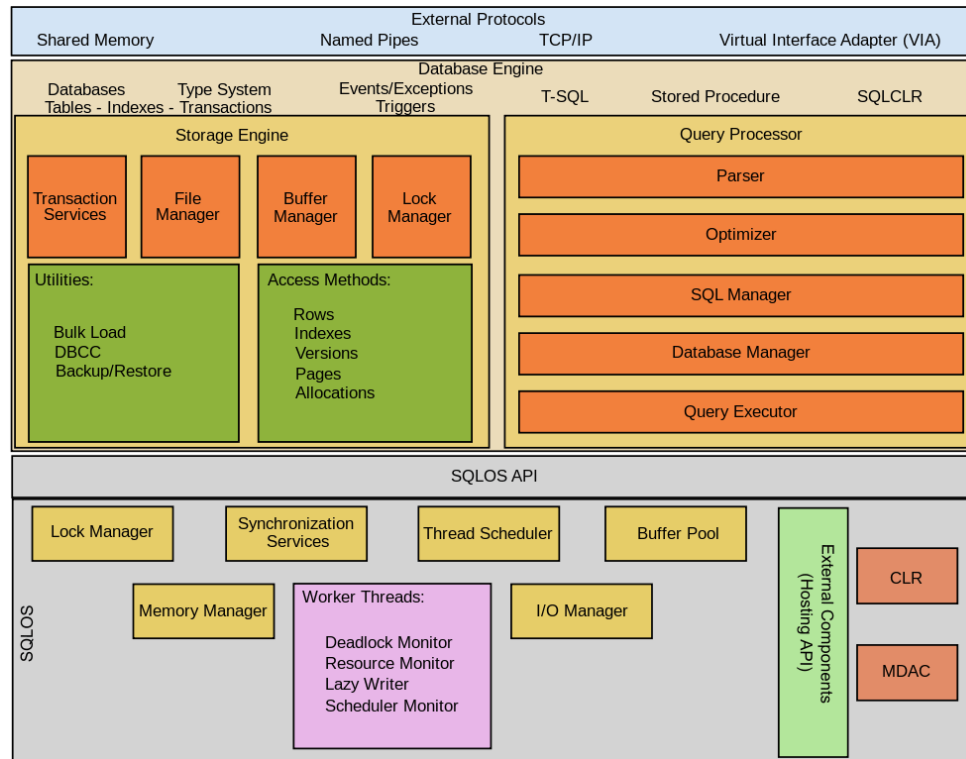
- **Dễ bảo trì:** Django có kiến trúc giúp dễ dàng mở rộng quy mô của ứng dụng bằng cách thêm phần cứng vào các thành phần khác nhau như bộ lưu trữ tạm thời, máy chủ cơ sở dữ liệu hoặc các loại thành phần khác.

- **Tính di động cao:** Django sử dụng ngôn ngữ Python cho nên các ứng dụng của nó có thể chạy trên đa nền tảng như Linux, Window, ... tạo ra độ linh hoạt cũng như có thể dễ dàng hỗ trợ đến từ nhà cung cấp dịch vụ.

2.5.7. Microsoft SQL Server

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu theo quan hệ và được hỗ trợ nhiều loại ứng dụng xử lý giao dịch, phân tích dữ liệu trong nhiều môi trường khác nhau. Phần mềm này được xây dựng trên ngôn ngữ truy vấn có cấu trúc, đây là ngôn ngữ thường được các lập trình viên có thâm niên cao ưa dùng, và được sử dụng rộng rãi vì nó có được tối ưu để có thể chạy trên những môi trường dữ liệu lớn có thể lên đến hàng trăm TB với mục đích cung cấp khả năng lưu trữ dành cho hàng trăm hay hàng ngàn người dùng.

Cùng với khả năng như trên thì đòi hỏi SQL server phải có một cấu trúc phức tạp để có thể quản lý nhiều người dùng, cùng với nhiều thao tác khác nhau trong cùng một thời điểm. SQL server sẽ giúp người dùng cũng như lập trình viên xử lý những chuyện lưu trữ và duy trì dữ liệu, hoặc phân tích và đưa ra báo cáo.



Hình 2.23: Cấu trúc của SQL server[16].

Dựa vào hình trên, cấu trúc chính của SQL server bao gồm:

- Database Engine: đây là thành phần chủ chốt của SQL server, nó đảm nhiệm về quản lý và xử lý dữ liệu. Bên trong nó còn những thành phần con như sau:
 - Storage Engine: đây là thành phần quản lý các tệp dữ liệu và tệp nhật ký giao dịch, quản lý bộ nhớ đệm và ghi lại mọi hoạt động của dữ liệu để đảm bảo về mặt tránh rủi ro của dữ liệu.
 - Query Processor: dùng để phân tích cú pháp các câu lệnh của SQP và chuyển đổi thành các cây cú pháp. Ngoài ra, thành phần này còn có thể tối ưu hóa các kế hoạch thực hiện đảm bảo việc truy vấn đạt được hiệu suất cao nhất có thể, cuối cùng thực hiện các kế hoạch đã được tối ưu trước đó và trả về kết quả.
 - Relational Engine: đây là nơi để quản lý cấu trúc của cơ sở dữ liệu như là bảng, chỉ mục, ràng buộc với các đối tượng khác kết hợp với điều khiển đồng thời, sử dụng các kỹ thuật để quản lý truy cập đồng thời đến dữ liệu.

- SQL server operating system (SQLOS): đây là lớp trừu tượng của cơ sở dữ liệu về mặt phần cứng, chịu trách nhiệm quản lý các thành phần của phần cứng như vùng nhớ, khả năng xử lý và ngoại vi.
- Giao thức bên ngoài: SQL cho phép các giao thức để tương tác với các hệ thống bên ngoài chẳng hạn như là TDS(Tabular Data Stream), ODBC(Open Database Connectivity),...

CHƯƠNG 3

THIẾT KẾ HỆ THỐNG

3.1. YÊU CẦU HỆ THỐNG

Hệ thống FOTA cho KIT học tập phát triển này cần được đáp ứng các yêu cầu sau:

- Một trang web có những chức năng sau:
 - **Lưu trữ file HEX:** Sử dụng một hệ cơ sở dữ liệu để lưu trữ file cũng như có thể xóa hoặc thêm file trên cơ sở dữ liệu này.
 - **Có khả năng gửi file HEX thông qua HTTP:** Đây là yếu tố quan trọng trong việc cập nhật chương trình cho bộ KIT, trang web sẽ sử dụng giao thức HTTP để có thể truyền tải dữ liệu file gửi cho phần cứng có kết nối internet và cập nhật cho vi điều khiển.
 - **Quan sát KIT bằng Camera và hiển thị trạng thái cập nhật:** Bên cạnh đó để đảm bảo rằng người dùng có thể đánh giá dữ liệu cập nhật thì sẽ có thêm một màn hình hiển thị camera trạng thái của kit khi được cập nhật xong, bên cạnh đó, sẽ không tránh khỏi được những yếu tố về kết nối mạng như chập chờn,... điều này sẽ ảnh hưởng đến dữ liệu cập nhật cho nên phải có thêm trạng thái thành công hay thất bại.
 - **Quản lý người dùng:** giống với lưu trữ file thì hệ cơ sở dữ liệu ở đây cũng sẽ lưu thông tin người dùng, cùng với xử lý việc nhiều người cùng truy cập cũng là một trường hợp không khó để gặp được.
- Một bộ KIT học tập có những yếu tố như sau:
 - **Có các module cơ bản phục vụ cho việc học tập:** Xây dựng bộ kit cùng với các ngoại vi cơ bản phục vụ cho việc thực hành cũng như cung cấp sự tiện lợi cho việc học tập nói chung.
 - **Có khả năng quay video và gửi về website kết quả sau khi cập nhật:** Phần cứng bắt buộc phải có ngoại vi để ghi lại hình ảnh kit để gửi về trang web cho người dùng.

○ **Có khả năng nạp được code cho vi điều khiển và trả về trạng thái cập nhật:** để có thể nạp được chương trình thì việc lấy file từ website về và phân tích nội dung trong đó cũng rất quan trọng, điều này giúp tăng độ chính xác của hệ thống này.

Để có thể hoàn thành được những yêu cầu ở trên, hệ thống này sẽ được thiết kế và được xây dựng dựa trên các kiến thức nền tảng và xây dựng trên các phần mềm hỗ trợ cùng với các tài liệu được công khai. Phần cứng ở đây toàn bộ các thành phần điện tử cơ bản cũng như các cảm biến. Hệ thống này sẽ sử dụng các thư viện cũng như là tài liệu để xây dựng từ phần cứng đến phần mềm.

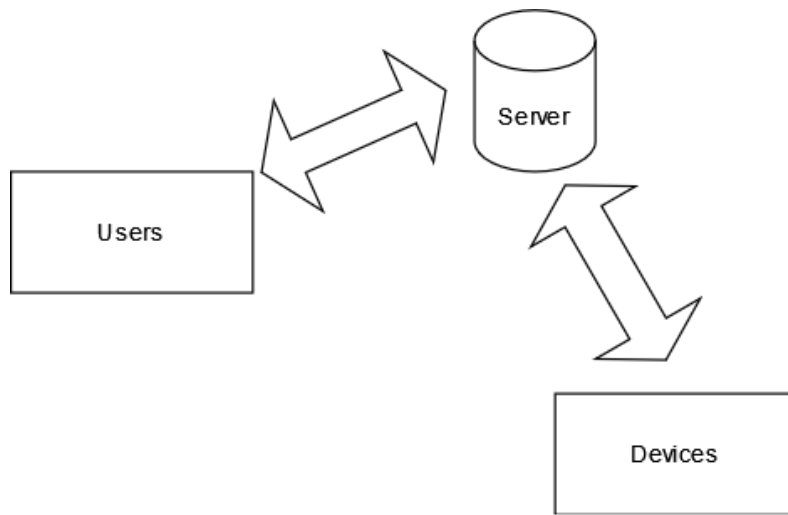
3.2. ĐẶC TẢ HỆ THỐNG

3.1.1. Chức năng hệ thống

Dựa trên yêu cầu hệ thống, thì nhóm đã quyết định đề tài sẽ được hiện thực hóa những chức năng như sau:

- Quản lý người dùng: Đảm bảo rằng trang web có thể quản lý thông tin người dùng, tránh việc xung đột do nhiều người truy cập vào bộ kit.
- Lưu trữ dữ liệu: Có thể thêm, xóa và lưu trữ file HEX để có thể nạp vào KIT.
- Cập nhật chương trình và hiển thị kết quả cập nhật: Cập nhật chương trình từ xa cho vi điều khiển và trả về trạng thái cập nhật thành công hoặc thất bại trong quá trình cập nhật.
- Stream video kết quả sau khi cập nhật: Giúp người dùng có thể quan sát được trạng thái KIT sau khi cập nhật cũng như kết quả giúp hệ thống này đảm bảo sự tiện lợi của việc thực hành từ xa.
- Bộ kit học tập phục vụ môi trường phòng lab: Bộ kit này phục vụ cho việc thực hành lập trình vi điều khiển và giao tiếp với các ngoại vi cơ bản.

3.1.2. Mô hình tổng quát của hệ thống

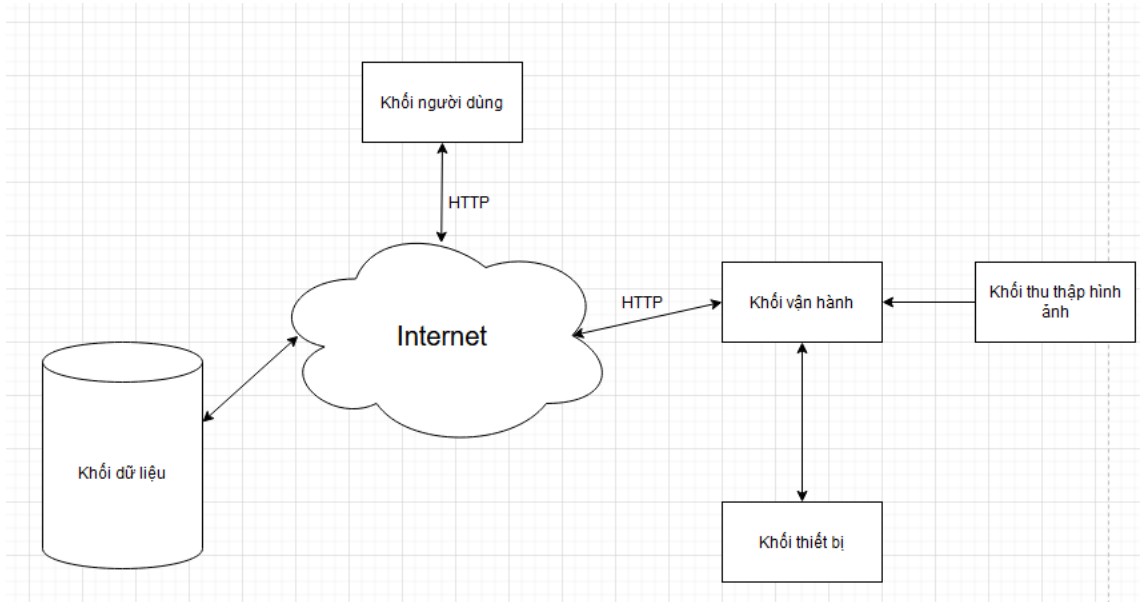


Hình 3. 1: Sơ đồ tổng quát hệ thống.

***Giải thích sơ đồ:**

- User: đây là phần giao diện người dùng tương tác trực tiếp với người dùng, đây sẽ là nơi người dùng thêm, xóa file, gửi file cập nhật chương trình cũng như quan sát kết quả thông qua camera.
- Server: đây là nơi lưu trữ dữ liệu thông tin người dùng cũng như lưu trữ file HEX phục vụ cho việc cập nhật.
- Devices: đây là khối cho các thiết bị bao gồm bộ kit và và thiết bị nạp cũng như camera để stream kết quả sau cập nhật và khối này sẽ giao tiếp với khối Server thông qua HTTP.

3.1.3. Sơ đồ khối và nguyên lý hoạt động của hệ thống



Hình 3.2: Sơ đồ khối hệ thống.

*Giải thích sơ đồ:

-Khối người dùng: Khối này sẽ tương tác trực tiếp với người dùng thông qua website. Cụ thể hơn các tương tác này sẽ là cập nhật chương trình và quan sát trạng thái của khối thiết bị cũng như là thông báo trạng thái cập nhật tới cho người dùng, khối này sẽ dùng giao thức để giao tiếp với internet.

-Khối dữ liệu: Khối này là một server chứa những dữ liệu quan trọng như thông tin người dùng và các file cập nhật chương trình cho khối thiết bị.

-Khối vận hành: Khối này sẽ sử dụng 1 bộ ESP32-CAM đóng vai trò là thiết bị vận hành bằng cách lấy dữ liệu cập nhật từ HTTP xong sẽ cập nhật chương trình đó bằng giao thức UART có trên vi điều khiển đó, trong quá trình cập nhật thì nếu có bất cứ lỗi hay sai sót thì sẽ ngừng cập nhật và gửi về cho người dùng thông báo là nó đang gặp vấn đề. Nếu không thì sẽ tiến hành thực hiện quay video lại bằng cách mở lại camera và đưa về cho người dùng thông qua HTTP.

-Khối thu thập hình ảnh: Khối này sử dụng 1 camera đã tích hợp sẵn trên module ESP32-CAM để thu thập hình ảnh sau khi cập nhật chương trình thành công.

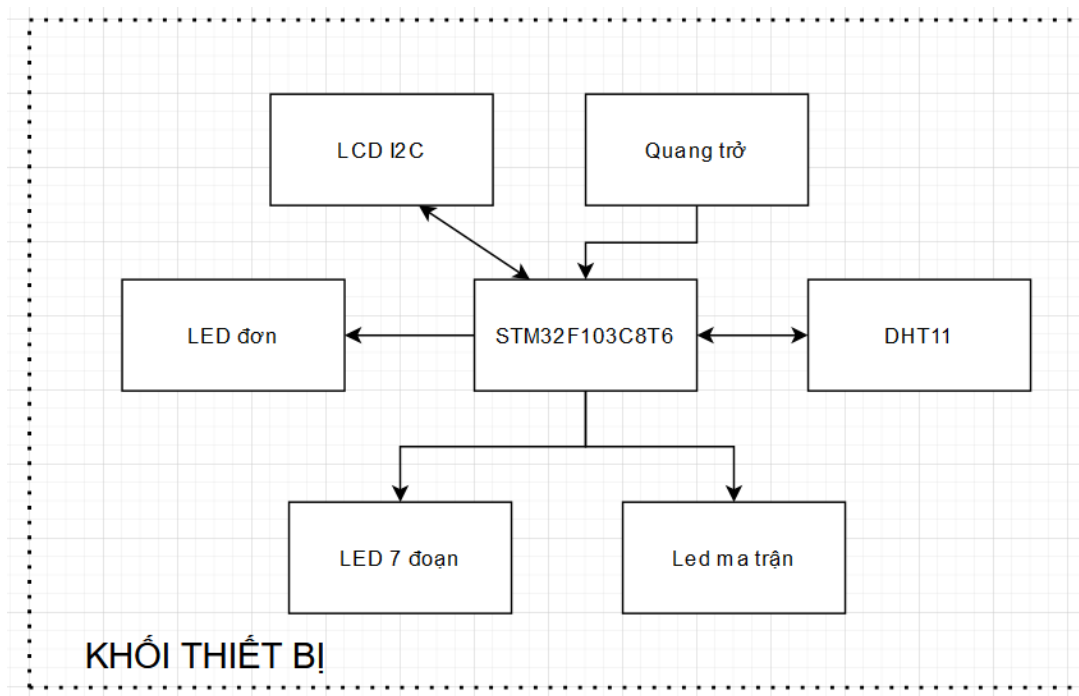
-Khối thiết bị: Khối này là một bộ kit học tập gồm bộ xử lý chính là STM32F103C8T6 để giao tiếp các ngoại vi cơ bản như là Led đơn, led 7 đoạn,

led ma trận, cảm biến nhiệt độ và độ ẩm DHT11, LCD 16x2, quang trở cùng với các IC mở rộng để giúp tối ưu số lượng chân giao tiếp.

***Nguyên lý hoạt động:**

Về nguyên lý hoạt động, bắt đầu từ người tạo tài khoản cá nhân cho người đó, sau đó sẽ có thể vào được giao diện để tương tác với kit phát triển, người dùng có thể thấy được có ai đang sử dụng kit đó hay chưa, nếu chưa có nghĩa là kit đó không có ai sử dụng thì người dùng có thể click để truy cập vào kit đó để thao tác, người dùng sẽ có thể uploads file Hex từ máy tính cá nhân đã được biên dịch sẵn. Sau đó nhấn nút nạp để gửi file đến kit đó và cập nhật. Cập nhật thành công hay thất bại thì người dùng đều nhận được thông báo. Sau khi cập nhật xong thì người dùng có thể quan sát camera để kiểm tra kết quả của đoạn code đó nếu như cập nhật thành công, tương tự với các lần cập nhật tiếp theo. Nếu không sử dụng nữa thì người dùng sẽ phải thoát ra để kit trống và người dùng khác có thể sử dụng được.

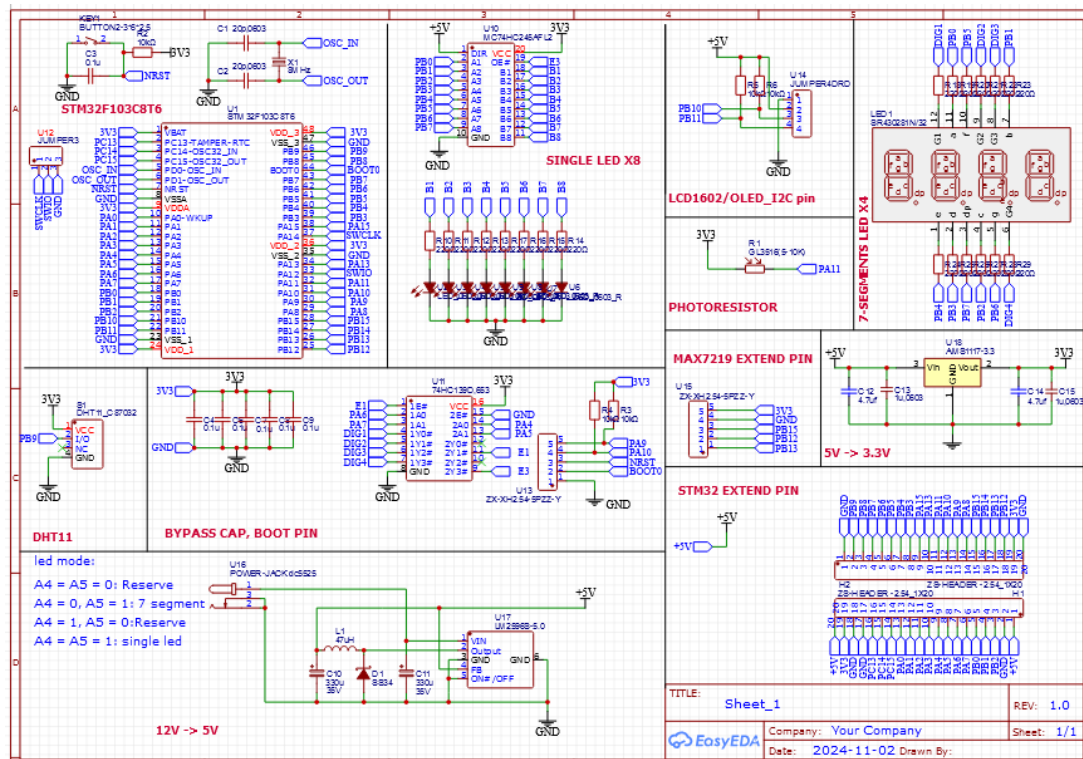
3.3. THIẾT KẾ PHẦN CỨNG



Hình 3.3: Sơ đồ tổng quát cho Kit phát triển.

Hình 3.3 trên mô tả sơ đồ tổng quát các khối trong mô hình kit phát triển, kit phát triển này gồm các module ngoại cơ bản phục vụ cho việc học tập như led đơn, led 7 đoạn, màn hình LCD và các loại cảm biến.

Để có thể thiết kế được theo cách giao tiếp theo kiểu trực tiếp thì không thể vì không đủ số lượng chân giao tiếp, do đó cần phải bổ sung các IC mở rộng cũng như điều khiển lựa chọn thiết bị sử dụng. Để có thể làm được điều đó ta lựa chọn được các IC phù hợp để đảm bảo nguồn điện có khả năng cung cấp đủ cho IC đó. Chẳng hạn như led ma trận, nhóm sẽ dùng IC MAX7219 để giao tiếp dữ liệu dựa trên giao thức SPI, hoặc là xử lý quét led 7 đoạn bằng IC 74HC139D,...

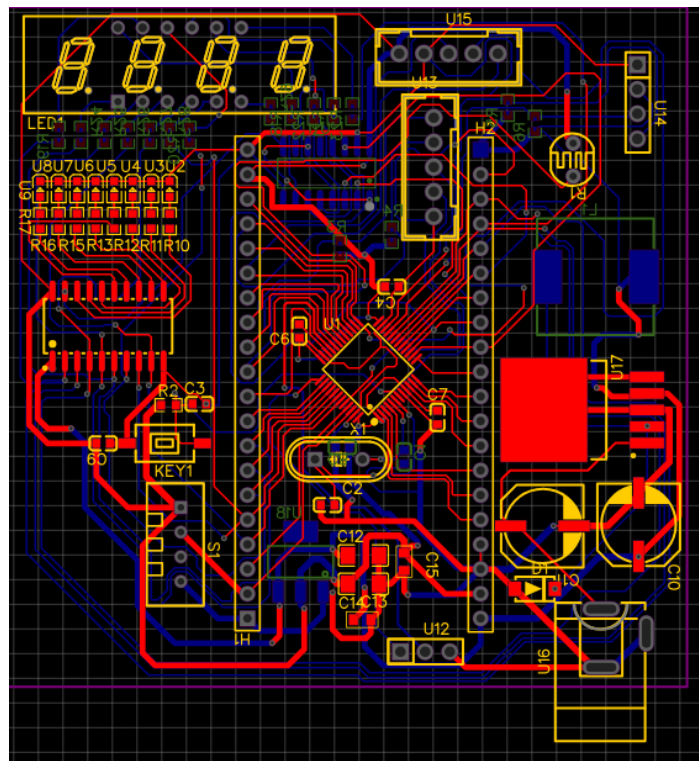


Hình 3.4: Sơ đồ nguyên lý của kit phát triển.

Hình trên là sơ đồ nguyên lý của mạch được chia thành các khối riêng biệt để dễ quản lý hơn. Bao gồm các khối như:

- Khối STM32F103C8T6: đây là khối quản lý vi điều khiển cũng như là bộ xử lý chính trong mạch. Bên cạnh đó được trang bị thêm nút nhấn Reset và thạch anh dao động để dễ quản lý. Ngoài ra còn có 1 jumper để có thể dùng cổng SWIO để lập trình viên có thể gỡ lỗi chương.
- Khối Single Led: đây là khối quản lý led đơn thông qua IC 74HC245 với mục đích đệm dòng và có thể lựa chọn không sử dụng thông qua chân cho phép.

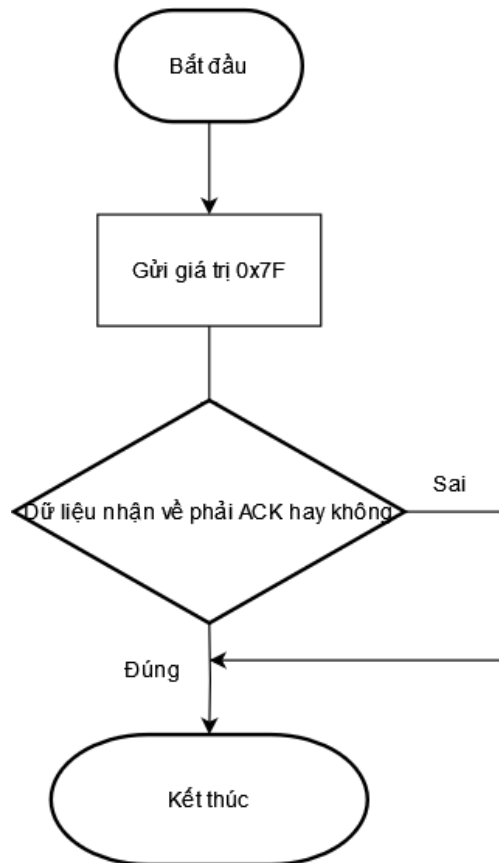
- Khối LCD/OLED: đây là 1 jumper để có thể kết nối module LCD hay OLED phụ thuộc vào lựa chọn người dùng.
- Khối Matrix LED: tương tự với khối LCD là sử dụng 1 jumper để mở rộng tùy ý người dùng.
- Khối 7Segment Led: Khối này quản lý LED 7 đoạn thông qua IC 74HC139D để quét LED và lựa chọn sử dụng module này.
- Khối Photoresistor: đây là 1 quang trở được kết nối trực tiếp với vi điều khiển nhằm phục vụ nghiên cứu về ADC.
- Khối DHT11: tương tự với quang trở, đây cũng là một cảm biến được kết nối trực tiếp với vi điều khiển.
- Khối Bypass và bootPin: đây là khối chứa tụ lọc nguồn của vi điều khiển và một jumper mở rộng phục vụ cho ứng dụng FOTA thông qua giao thức UART.
- Khối nguồn: khối nguồn ở đây được chia thành 2 khối nhỏ, một là một khối nhận nguồn đầu vào là 12V đi qua IC LM2596S-5.0 để cho ra ngõ ra điện áp 5V dùng để cung cấp các module cần thiết, hai là một khối chuyển đổi ngõ ra 5.0 của LM2596S-5.0 thành 3.3V bằng AMS1117 cung cấp nguồn cho vi điều khiển và các module cần thiết.



Hình 3.5: Sơ đồ layout mạch PCB của kit phát triển.

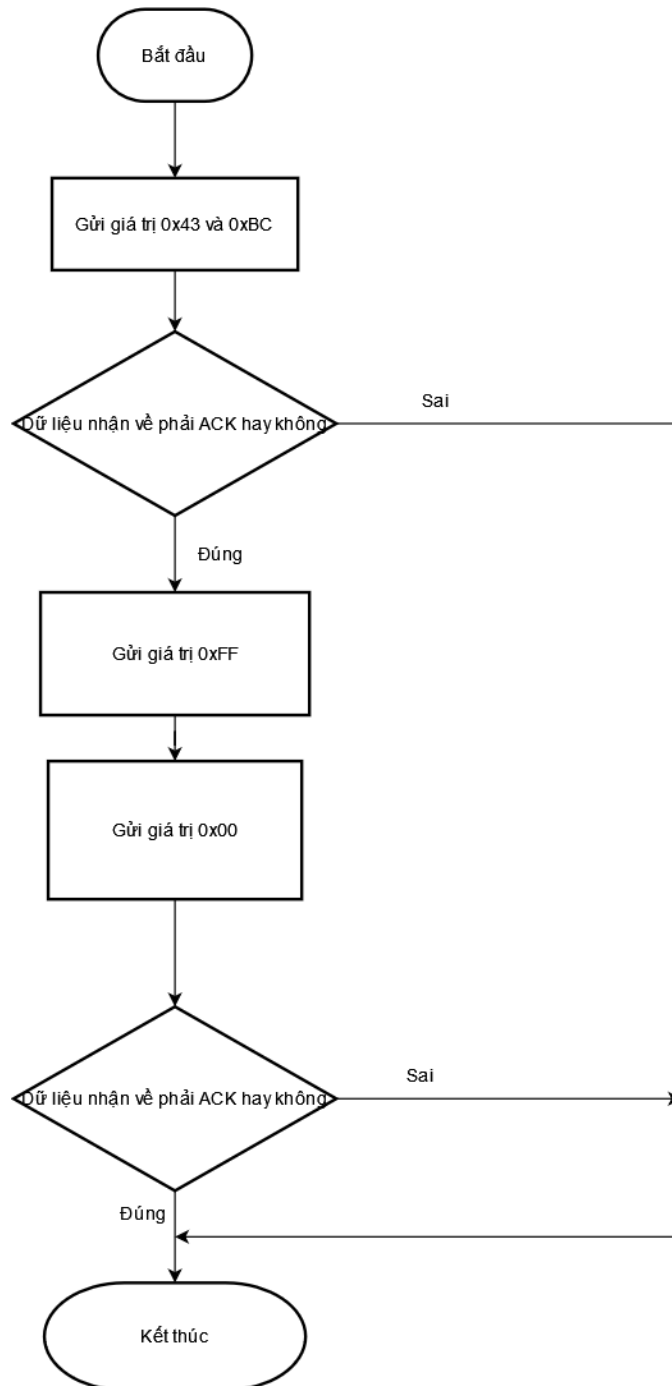
3.4. THIẾT KẾ PHẦN MỀM

3.3.1. Chương trình trên bootloader



Hình 3.6: Lưu đồ hoạt động kiểm tra xem có đang ở bootloader hay không.

Hình 3.6 bên trên là lưu đồ cho hoạt động kiểm tra có đang ở trong chương trình bootloader hay không bằng cách là gửi đi giá trị 0x7F thông qua UART, sau đó nếu dữ liệu trả về là ACK hay 0x79 thì có nghĩa rằng là hệ thống đang ở chương trình bootloader, ngược lại nếu dữ liệu trả về là NAK hay 0x1F hoặc không trả về giá trị nào có nghĩa rằng hệ thống chưa về chế độ boot hoặc gửi đi không đúng cách. Mục đích vào bootloader là để thực thi các lệnh như xóa vùng nhớ, đọc dữ liệu hay ghi dữ liệu trong vùng nhớ Flash. Đây là tiền đề cho các hệ thống về FOTA cho việc cập nhật dữ liệu mới cho vi điều khiển. Và ngoài ra, các hoạt động khác ghi, đọc, xóa đều cần đọc lại trạng thái trả về như là ACK hay NAK.



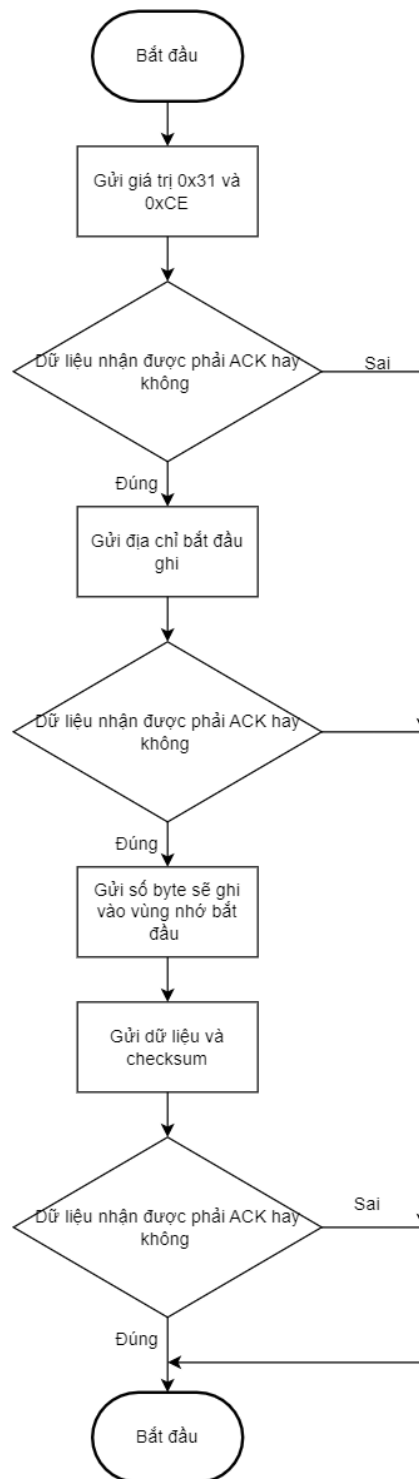
Hình 3.7: Lưu đồ hoạt động cho việc xóa vùng nhớ chương trình.

Hình trên mô tả việc xóa dữ liệu chương trình trong vùng nhớ Flash, bằng cách sẽ gửi đi lệnh xóa 0x43 sau đó sẽ gửi đi 0xBC để gọi lệnh xóa đến từ vi điều khiển, và vi điều khiển sẽ kiểm tra xem lệnh này có hợp lệ hay không, nếu có sẽ trả về ACK tức 0x79 về thông qua giao thức UART. Do ở vi điều khiển STM32F103xx thì sẽ có hai cách xóa dữ liệu: đó là xóa toàn bộ và

xóa từng trang. Ở đây nhóm chỉ thao tác với việc xóa toàn bộ dữ liệu cũ đi và sử dụng lệnh ghi để ghi mới dữ liệu vào vùng nhớ Flash, để xóa toàn bộ dữ liệu trong vùng nhớ đó thì sẽ gửi thêm lệnh xóa toàn bộ là 0xFF và 0x00 cho vi điều khiển. Nếu lệnh hợp lệ thì vi điều khiển sẽ tự động xóa hết dữ liệu trong vùng nhớ và trả về ACK. Hoặc trả về NAK nếu nhập lệnh sai hoặc không trả bất kì gì về.

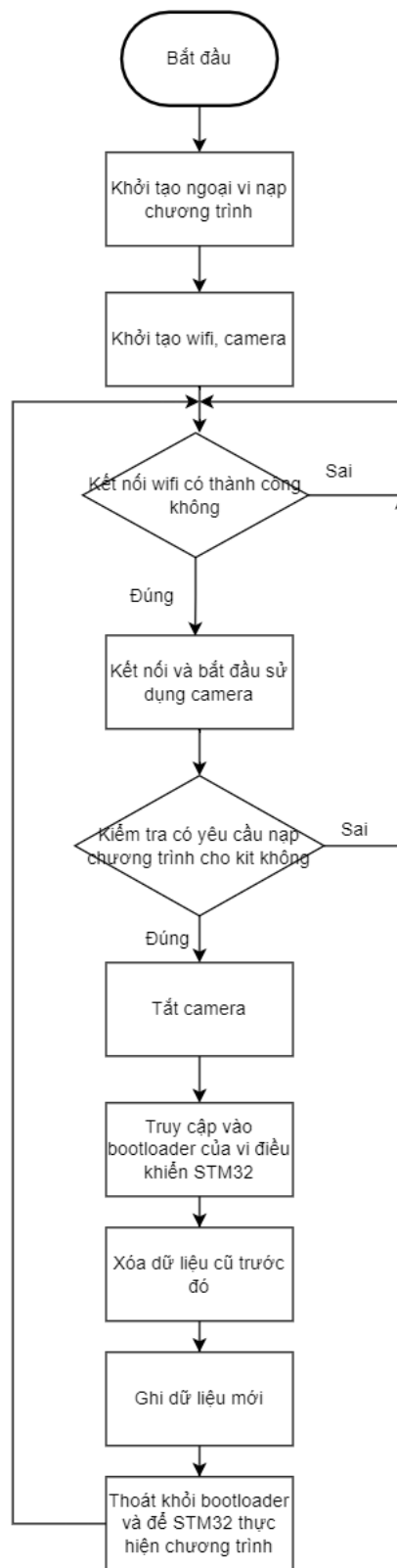
Thay vì vậy ta có thể ghi đè lên dữ liệu cũ bằng dữ liệu mới mà không cần phải xóa toàn bộ. Do sẽ có một vài rủi ro nếu ta không xóa trước đó chẳng hạn như là trong trường hợp dữ liệu mới sẽ ít hơn dữ liệu cũ thì khi đó và sau khi ghi đè lên xong phần dữ liệu thừa của dữ liệu cũ sẽ vẫn còn đó, điều đó có thể khiến cho chương trình chạy sai, không như mong muốn người dùng.

Hình 3.8 là một lưu đồ giải thích hoạt động ghi dữ liệu vào vùng nhớ Flash của vi điều khiển, và hoạt động ghi này cần bốn dữ liệu quan trọng từ file hex đó là số lượng byte cần gửi, địa chỉ cần ghi, dữ liệu, cuối cùng là checksum của toàn bộ ba thành phần trên. Đầu tiên, để có ghi thể bắt đầu lệnh ghi ta cần gửi đi lệnh ghi bằng cách gửi giá trị 0x43 và 0xBC cho vi điều khiển và vi điều khiển sẽ kiểm tra xem đó có phải là một lệnh ghi phù hợp hay chưa, nếu đúng sẽ trả về giá trị ACK. Sau khi nhận được ACK, ta tiến hành gửi đi số lượng byte cần ghi bao gồm số lượng byte dữ liệu địa chỉ và số lượng byte dữ liệu thực thi của chương trình. Sau đó ta sẽ lấy địa chỉ từ file hex của chương trình, do là địa chỉ trong file đó chỉ có 2 bytes 0xFFFF để có thể gửi toàn bộ đi thì bắt buộc phải kết hợp với địa chỉ bắt đầu của vi điều khiển STM32F1 là 0x80000000 (4 bytes) và gửi đi cho vi điều khiển, nếu nó trả về giá trị ACK thì ta mới tiến hành gửi các dữ liệu thực thi vào trong vùng nhớ flash, tối đa 16 bytes, và cuối cùng ta tính toán checksum và gửi đi cho vi điều khiển, nếu như vi điều khiển trả về kết quả ACK thì đồng nghĩa với việc ta đã ghi thành công, và ta sẽ tận dụng đó để ghi thêm dữ liệu khác mà không cần phải gửi đi lệnh ghi lần nữa.



Hình 3.8: Lưu đồ hoạt động ghi dữ liệu vào vùng nhớ chương trình.

3.3.2. Chương trình cho ESP32-CAM



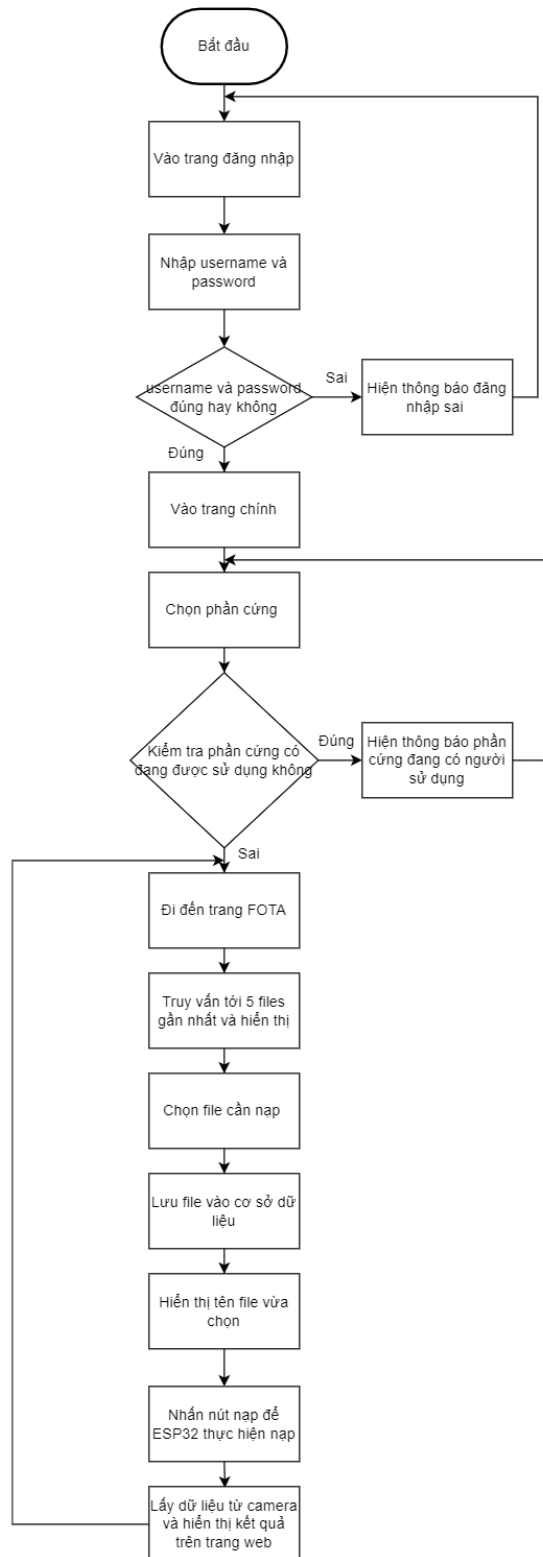
Hình 3.9: Lưu đồ hoạt động của ESP32-CAM.

Về hoạt động của esp32-CAM như hình 3.9 sẽ có ba nhiệm vụ chính, đầu tiên là download firmware file từ website và lưu về vùng nhớ nội của ESP32, tiếp theo sẽ lấy dữ liệu đó nạp dữ liệu cho STM32F103xx thông qua giao thức UART, cuối cùng sẽ quay video lại và hiển thị trên website.

Đầu tiên, khi khởi động nguồn ESP32-CAM sẽ kết nối với WIFI đã được thiết lập sẵn trong chương trình và sau đó sẽ khởi tạo các ngoại vi cần thiết cho việc nạp và cập nhật dữ liệu cho kit học tập phát triển, và khởi tạo cả camera, sau đó kiểm tra wifi có còn được kết nối hay không nếu còn thì gửi camera và chờ lệnh nạp code, nếu không sẽ phải yêu cầu để kết nối lại wifi đó. Trong tình huống có yêu cầu nạp code từ phía website thì esp32 cam bắt buộc phải tạm thời ngừng stream video và download file được yêu cầu nạp và tiến hành gửi lệnh cho STM32 rằng phải vào vùng bootloader sau đó gửi lệnh xóa và nạp dữ liệu. Sau khi nạp thành công thì thoát khỏi bootloader và để STM32 thực thi dữ liệu đã nạp và mở lại cam chờ lần nạp tiếp theo.

3.3.3. Giao diện và chức năng trên Website

Về phía website, thì website này đang có vài chức năng cơ bản như đăng nhập xác thực người dùng. Lựa chọn truy cập vào phần cứng đang trống. Tải file lên cơ sở dữ liệu và gửi file thông qua HTTP, ngoài ra còn có khả năng xem video đến từ esp32-cam để xem kết quả. Lưu đồ 3.10 dưới đây mô tả các hoạt động cơ bản về chức năng của website này. Đầu tiên, phải đăng nhập để làm bước tiếp theo, nếu tài khoản hoặc mật khẩu sai, thì người dùng bắt buộc phải nhập đúng mới làm được các bước tiếp theo. Sau đó, nếu đăng nhập thành công thì lựa chọn phần cứng. Sau đó ta vào được trang để tương tác và nạp code cho phần cứng, ta có thể đưa file mới lên hoặc dùng lại file cũ đã đưa lên trước đó, nếu người dùng muốn chọn file nào thì có thể nhấn nút select để chọn file đó, sau khi chọn file thành công thì esp32-cam sẽ tiến hành nạp code cho kit phát triển trong khoảng vài giây, sau khi nạp xong thì người dùng có thể xem được kết quả ở phần camera.



Hình 3.10: Lưu đồ hoạt động của website.

CHƯƠNG 4

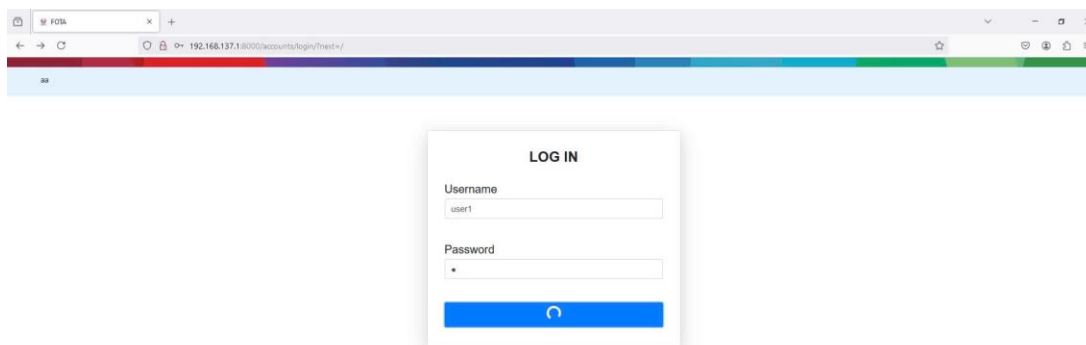
KẾT QUẢ VÀ ĐÁNH GIÁ

4.1. KẾT QUẢ



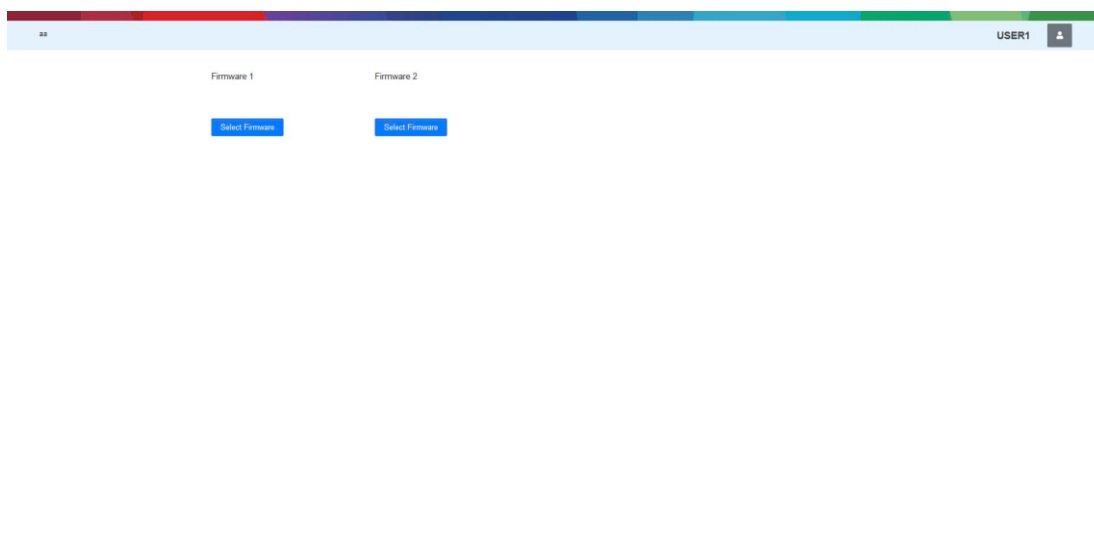
Hình 4. 1: Sơ đồ thực tế mạch kit học tập phát triển.

Hình trên là kết quả sau thi công đối với mô hình kit phát triển. Có các module cơ bản phục vụ học tập. Tuy nhiên, trong quá trình thi công thì nhóm đã gặp một vài khó khăn khiến cho mô hình này tốn nhiều thời gian hơn để sửa chữa chẳng hạn như là lựa chọn linh kiện trong quá trình thiết kế mạch chưa hợp lý, cần vài yếu tố thêm để có thể chạy được. Kết quả là mô hình này sẽ không được đẹp mắt như dự kiến, nhưng nó vẫn hoạt động bình thường. Nguồn điện cung cấp được cho mô hình là 12V thông qua Jack cắm DC, sau đó thông qua các IC khác để giảm nguồn thành 5V và 3.3V để đáp ứng cho các thành phần bên trong mạch đó.



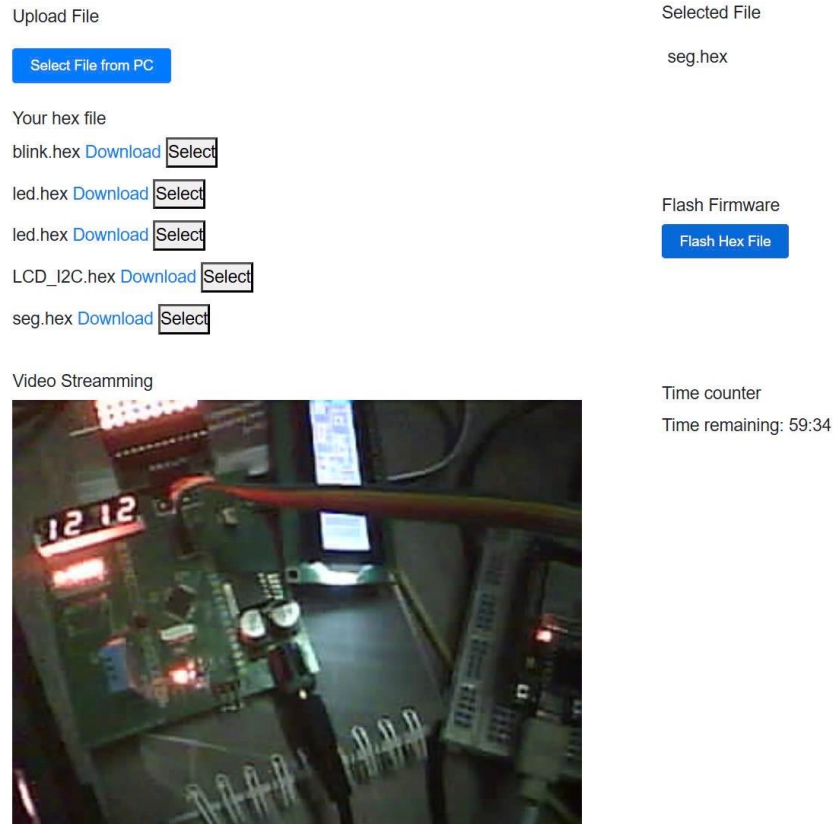
Hình 4.2: Giao diện phần đăng nhập trên website.

Đối với website thì trên hình 4.2 là giao diện xác thực người dùng thông qua tài khoản và mật khẩu, nếu nhập sai thì sẽ không được vào các bước tiếp theo của website này. Sau khi đăng nhập thành công thì người dùng có thể truy cập vào trang website chọn thiết bị như hình dưới đây.



Hình 4.3: Giao diện phần chọn thiết bị.

Hình 4.3 trên đây là trang để người dùng có thể thấy được các thiết bị được trang web quản lý. Người dùng có thể chọn một trong số các thiết bị hiển thị trên web. Sau đó người dùng sẽ được dẫn đến một trang khác để có thể nạp code và đưa file code lên.



Hình 4.4: Giao diện chính tương tác với thiết bị.

Sau khi lựa chọn được thiết bị, người dùng được dẫn đến trang web giống như hình 4.4 để có thể chọn file, và nạp phần chương trình xuống kit phát triển. Người dùng có hai sự lựa chọn để chọn file, có thể dùng nút upload file để chọn file mới từ máy tính của người dùng. Hoặc có thể chọn file đã có sẵn từ trước, ở đây website sẽ lưu 5 file gần nhất mà người dùng để tải lên, do đó người dùng có thể lựa chọn file để có thể nạp cho vi điều khiển STM32F103 một cách linh động. Ngoài ra ở trang này, sẽ có một đồng hồ đếm giờ với mục đích để người dùng có thể thấy được thời gian còn lại để sử dụng kit phát triển, nếu thời gian đếm về 00:00 thì trang web sẽ tự động đưa người dùng ra khỏi trang hiện tại và đưa đến trang web như hình 4.3. Nếu người dùng muốn dùng thêm thì có thể truy cập lại web đó và thời gian sẽ được đặt lại về thời gian mới nhất. Tương tự với lần trước đó khi người dùng hết thời gian.

4.2. ĐÁNH GIÁ

4.2.1. Giới thiệu các trường hợp kiểm chứng.

Ở đề tài này có rất nhiều thứ dễ bị ảnh hưởng với nhau. Ví dụ như là giả sử nếu có nhiều người dùng cần sử dụng nhưng số lượng kit phát triển thì có giới hạn. Hay là các chức năng cơ bản khác, và những chức năng đó có thể vô tình bị xung đột với nhau. Điều đó là không thể tránh khỏi. Về việc kiểm chứng các trường hợp khác của website thì thời gian không đủ đáp ứng cho việc hiện thực hóa các tính năng quản lý như quản lý nhiều người truy cập cùng lúc. Nhóm sẽ cân nhắc trong tương lai về hướng phát triển của đề tài này và hiện thực hóa nó. Ngoài những điều xung đột khi hiện thực hóa với nhau thì người dùng rất quan tâm đến việc thời gian nạp của các files với các dung lượng khác nhau. Vì người dùng có 1 thời gian nhất định để thao tác với kit phát triển, và giao thức UART cũng không thể đáp ứng được việc nạp ngay tức thì. Cho nên nhóm sẽ kiểm chứng hai trường hợp dễ gặp nhất vì các dự án cơ bản sẽ dễ gặp hai trường hợp này. Đó là đối với dung lượng khác nhau thì thời gian nạp của mỗi dung lượng đó sẽ như thế nào. Đây cũng là một điểm đặt biệt trong hướng phát triển của đề tài này. Đó là đề xuất thuật toán để giảm thời gian nạp xuống nhiều nhất có thể và mang lại cho người dùng cảm giác không còn phải chờ nạp.

4.2.2. Kiểm chứng trường hợp dung lượng file từ 10KB trở xuống.

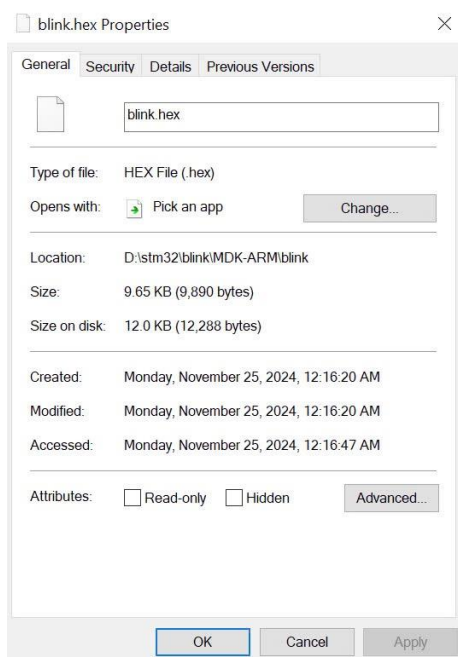
*Điều kiện tiên quyết:

- Cần có một file thực thi cho STM32 với dung lượng 10KB trở xuống.
- Thiết bị được kết nối đầy đủ.
- ESP32 phải được kết nối mạng.

*Kết quả kiểm chứng:

- Số lần kiểm chứng: 5 lần.

Hình 4.5 dưới đây cung cấp thông tin của file thực thi với dung lượng là 10KB, kến đến nhóm sẽ thử nghiệm với 5 lần download file và nạp lên kit, đo xem với dung lượng này thì hệ thống này có bị ảnh hưởng đến thời gian nạp hay không.



Hình 4.5: Thông tin về dung lượng file hex dưới 10KB.

```
Write data to address successfully.
Exiting bootloader mode...
Exit bootloader mode done
Firmware upload complete
Finish at: 4063
HTTP GET code: 200
```

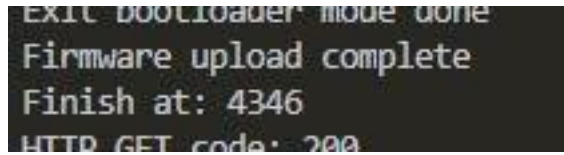
Hình 4.6: Kiểm chứng thời gian lần cập nhật thứ 1.

```
Exit bootloader mode done
Firmware upload complete
Finish at: 4098
HTTP GET code: 200
```

Hình 4.7: Kiểm chứng thời gian lần cập nhật thứ 2.

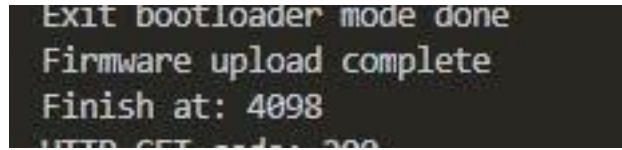
```
Exiting bootloader mode...
Exit bootloader mode done
Firmware upload complete
Finish at: 4063
HTTP GET code: 200
```

Hình 4.8: Kiểm chứng thời gian lần cập nhật thứ 3.



```
EXIT bootloader mode done
Firmware upload complete
Finish at: 4346
HTTP GET code: 200
```

Hình 4.9: Kiểm chứng thời gian lần cập nhật thứ 4.



```
EXIT bootloader mode done
Firmware upload complete
Finish at: 4098
HTTP GET code: 200
```

Hình 4.10: Kiểm chứng thời gian lần cập nhật thứ 5.

Kiểm chứng với 5 lần thử nghiệm như trên, ta nhận thấy giữa các lần sẽ có thời gian chênh lệch một lượng cũng khá đáng kể, với thời gian cao nhất ở lần thử thứ 4 là 4346 ms tức là 4.3s là thời gian để ESP32 có thể download file này và nạp vào ESP32. Và thời gian thấp nhất để nạp là 4063 ms tức là khoảng 4s.

4.2.3. Kiểm chứng trường hợp dung lượng file từ 20KB trở xuống.

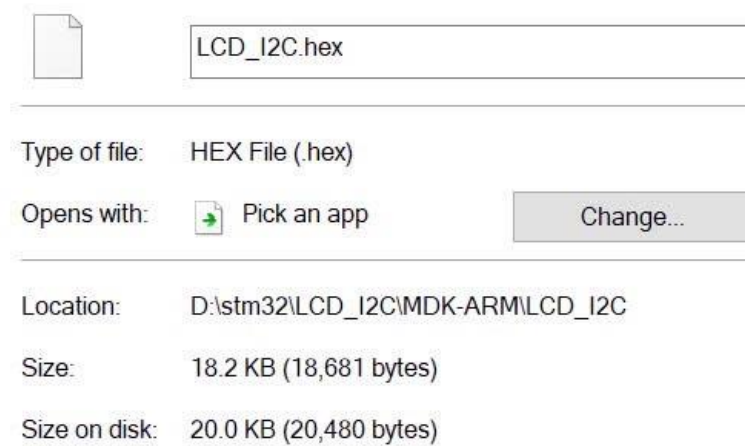
***Điều kiện tiên quyết:**

- Cần có một file thực thi cho STM32 với dung lượng 20KB trở xuống và phải lớn hơn 10KB.
- Thiết bị được kết nối đầy đủ.
- ESP32 phải được kết nối mạng.

***Kết quả kiểm chứng:**

- Số lần kiểm chứng: 5 lần.

Hình 4.11 dưới đây cung cấp thông tin của file thực thi với dung lượng là 20KB, kén đến nhóm sẽ thử nghiệm với 5 lần download file và nạp lên kit, đo xem với dung lượng này thì hệ thống này có bị ảnh hưởng đến thời gian nạp hay không. Sau đó sẽ so sánh thời gian này với thời gian cập nhật của file với dung lượng 10KB bằng cách trực quan hóa thông qua biểu đồ.



Hình 4.11: Thông tin về dung lượng file hex dưới 20KB.

```
Firmware upload complete
Finish at: 6716
HTTP GET code: 200
```

Hình 4.12: Kiểm chứng thời gian lần cập nhật thứ 1.

```
Exit bootloader mode done
Firmware upload complete
Finish at: 7442
```

Hình 4.13: Kiểm chứng thời gian lần cập nhật thứ 2.

```
Firmware upload complete
Finish at: 6957
HTTP GET code: 200
```

Hình 4.14: Kiểm chứng thời gian lần cập nhật thứ 3.

```
Exit bootloader mode done
Firmware upload complete
Finish at: 7229
```

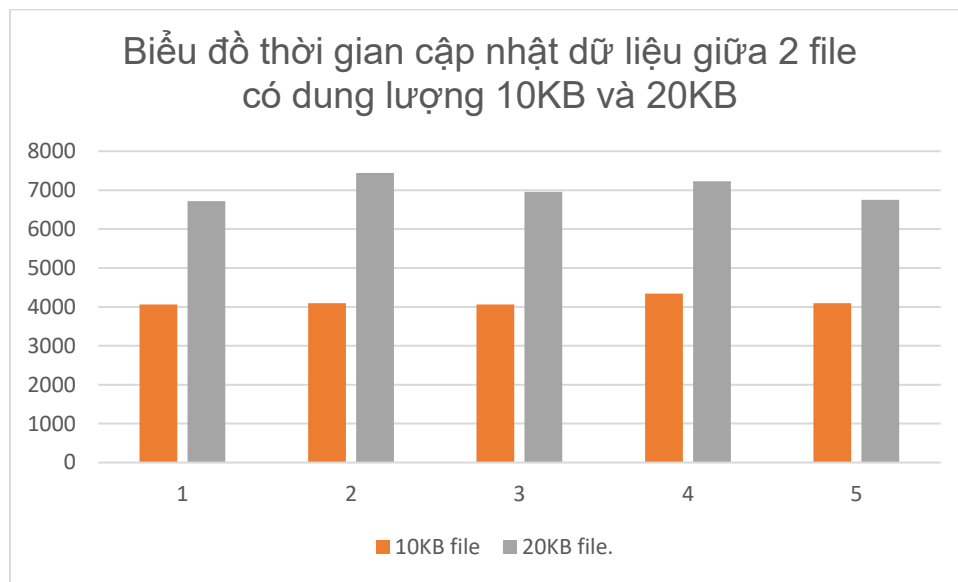
Hình 4.15: Kiểm chứng thời gian lần cập nhật thứ 4.

```
Firmware upload complete
Finish at: 6752
HTTP GET code: 200
```

Hình 4.16: Kiểm chứng thời gian lần cập nhật thứ 5.

Tương tự với file có dung lượng nhỏ hơn, thì files có dung lượng 20KB cũng sẽ bị ảnh hưởng vì chênh lệch giữa các lần với nhau với nguyên nhân là do tốc độ tải file về. Nhóm nhận thấy rằng thời gian nhỏ nhất để thực thi thành công là 6716ms tức khoảng 6.7s để cập nhật và thời gian lớn nhất để cập nhật thành công đó là khoảng 7.4s một con số chênh lệch khá lớn.

Kế đến nhóm sẽ tiến hành so sánh giữa các lần với nhau như hình 4.17 bên dưới:



Hình 4.17: Biểu đồ so sánh thời gian thực thi qua 5 lần thử nghiệm giữa file 10KB và file 20KB.

Nhìn vào hình 4.17, có thể thấy được cột màu cam(file 10KB) và cột màu xám(file 20KB) có sự chênh lệch với nhau khá lớn. Nhưng so với tỉ lệ giữa hai file là 1:2 thì thời gian trung bình của cột màu cam là khoảng 4000ms hay 4s và cột màu xám khoảng 7000ms hay 7s thì tỉ lệ thời gian sẽ không giống như so với tỉ lệ dung lượng giữa hai file này, nhưng nhìn chung thì khi có sự chênh lệch về mặt dung lượng thì thời gian cũng sẽ chênh lệch theo. Đây cũng là một điểm cần chú ý nếu cần nạp một file với kích thước dung lượng lớn hơn và là một điểm đặc biệt cho hướng phát triển trong tương lai.

CHƯƠNG 5

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. KẾT LUẬN

Sau khoảng thời gian tìm hiểu, phát triển và vượt qua những khó khăn nhất định trong việc thực hiện đề tài này. Cùng với tham khảo qua các nguồn tài liệu khác nhau trên các website, thư viện số, tài liệu của nhà sản xuất,... Song cũng có tham khảo những ý kiến đóng góp của thầy Đỗ Duy Tân giúp hoàn thiện đề tài này một cách tốt nhất có thể, cuối cùng đề tài “THIẾT KẾ VÀ ỨNG DỤNG HỆ THỐNG FIRMWARE OVER-THE-AIR (FOTA) CHO PHÒNG LAB TỪ XA” cùng với các tính năng nổi bật như sau:

- Thiết kế kit phát triển cùng với sáu ngoại vi cơ bản như led đơn, led 7 đoạn, led ma trận, LCD 16x2, cảm biến nhiệt độ và độ ẩm, quang trở cùng với các thành phần linh kiện điện tử khác.
- Xây dựng chương trình để nạp firmware cho kit STM32 thông qua mạng không dây.
- Xây dựng 1 website để cho phép người dùng có thể nạp code và quản lý file thực thi.
- Tích hợp đồng hồ để quản lý thời lượng sử dụng trên Website.
- Tích hợp ô hiển thị camera stream kết quả.

Bên cạnh đó, hệ thống này còn thiếu nhiều tính năng quan trọng như sau:

- Về việc quản lý người dùng trên hệ thống này cần nhiều điều kiện về mặt thời gian để có thể thực thi.
- Người dùng chưa thể nhận được kết quả cập nhật sau khi cập nhật xong.

5.2. HƯỚNG PHÁT TRIỂN

Dựa vào kết quả và đánh giá, do đề tài này tính ứng dụng còn mới, nên có rất nhiều hướng phát triển trong tương lai về mặt học thuật và ứng dụng. Chính vì

thể nhóm đề xuất 1 vài hướng phát triển tiêu biểu cũng về lĩnh vực phần mềm và cả phần cứng như sau:

***Về mặt phần cứng:**

- Mở rộng mô hình kit phát triển, bổ sung thêm các module cơ bản khác giao tiếp với vi điều khiển khác phục vụ học tập.

- Mở rộng mô hình hệ thống, nâng cấp hệ thống mạng lớn hơn cho phù hợp môi trường phòng LAB.

- Thay đổi giao thức nạp dữ liệu một cách linh hoạt.

- Sử dụng nút nhấn bằng phần mềm thay cho phần cứng, do đôi khi trong phòng LAB các nghiên cứu hay học về các nút nhấn cũng rất cần thiết.

- Có thể bật tắt nguồn từ xa bằng cách sử dụng chân ON/OFF của LM2596S-5.0 bằng các giải pháp IoT.

***Về mặt phần mềm:**

- Sử dụng giải thuật chỉ ghi vào những địa chỉ có dữ liệu khác với dữ liệu đã có trước đó để tối ưu thời gian thực hiện lệnh, còn dữ liệu giống với cũ thì giữ nguyên.

- Chuyển đổi các giao thức một cách linh hoạt.

- Tích hợp thêm tính năng lớp học lớn và nhóm học nhỏ trên website phù hợp cho môi trường các lớp học thực hành.

TÀI LIỆU THAM KHẢO

- [1]. embitel. (2024). “*Firmware Over The Air – For IoT And Automotive Devices*”. <https://www.embitel.com/firmware-over-the-air-fota-updates-for-iot-and-automotive-devices>
- [2]. GeeksforGeeks. (2024). “*What is HTTP?*”. <https://www.geeksforgeeks.org/what-is-http/>.
- [3]. Beningo. (2017). “*Tips and Tricks – Jumping from the Bootloader to the Application Code Cleanly*”. <https://www.beningo.com/tips-and-tricks-jumping-from-the-bootloader-to-the-application-code-cleanly/#>.
- [4]. Dientunhattung. (2024). “*STM32F103C8T6*”. <https://dientunhattung.com/san-pham/stm32f103c8t6-ic-roi-vi-dieu-khien-ho-stm-cua-atmel/>.
- [5]. Nshop. (2024). “*Giới thiệu mạch thu phát Wifi BLE ESP32-CAM Ai-Thinker*”. <https://nshopvn.com/blog/gioi-thieu-mach-thu-phat-wifi-ble-esp32-cam-ai-thinker-huong-dan-su-dung-voi-arduino-thuc-hanh-lam-bo-mo-khoa-cua-nhan-dien-khuon-mat-bang-esp32-cam/>.
- [6]. Nshop. (2024). “*Màn hình LCD1602 xanh lá*”. <https://nshopvn.com/product/man-hinh-lcd-1602-xanh-la/>.
- [7]. Nshop. (2024). “*Mạch chuyển đổi giao tiếp I2C cho LCD*”. <https://nshopvn.com/product/mach-chuyen-doi-giao-tiep-i2c-cho-lcd/>.
- [8]. anasa. (2024). “*Quang Trở CDS 5mm - Cảm Biến Ánh Sáng 5mm 5537*”. <https://anasa.vn/quang-tro-cds-5mm-cam-bien-anh-sang-5mm>.
- [9]. Thegioiic, (2024). “*Module led ma trận với ic MAX7219*”. <https://www.thegioiic.com/module-led-ma-tran-8x8-max7219-5vdc>.
- [10]. Nexperia. (2024). “*74HC139D datasheet*”. https://assets.nexperia.com/documents/data-sheet/74HC_HCT139.pdf.
- [11]. Nexperia. (2024). “*74HC245 datasheet*”. https://assets.nexperia.com/documents/data-sheet/74HC_HCT245.pdf.
- [12]. Nshop. (2024). “*Cảm biến nhiệt độ và độ ẩm DHT11*”. <https://nshopvn.com/product/module-cam-bien-do-am-nhiet-do-dht11/>.
- [13]. Waveshare. (2024). “*STM32CubeMX Tutorial Series: Overview*”. https://www.waveshare.com/wiki/STM32CubeMX_Tutorial_Series:_Overview.

- [14]. VisualStudioCode. (2024). “*Why did we build Visual Studio Code?*”. <https://code.visualstudio.com/docs/editor/whyvscode>.
- [15]. IBM. (2024). “What is Django?”. <https://www.ibm.com/topics/django>.
- [16]. Wikipedia. (2024). “*Microsoft SQL Server*”. https://en.wikipedia.org/wiki/Microsoft_SQL_Server.
- [17]. STMicroelectronics (2024). “*AN2606 application note STM32 microcontroller system memory boot mode*”. https://www.st.com/resource/en/application_note/cd00167594-stm32-microcontroller-system-memory-boot-mode-stmicroelectronics.pdf.
- [18]. STMicroelectronics (2015). “*RM0008 reference manual*”. https://www.st.com/resource/en/reference_manual/rm0008-stm32f101xx-stm32f102xx-stm32f103xx-stm32f105xx-and-stm32f107xx-advanced-armbased-32bit-mcus-stmicroelectronics.pdf.
- [19]. STMicroelectronics (2024). “*USART protocol used in the STM32 bootloader*”. https://www.st.com/resource/en/reference_manual/rm0008-stm32f101xx-stm32f102xx-stm32f103xx-stm32f105xx-and-stm32f107xx-advanced-armbased-32bit-mcus-stmicroelectronics.pdf.

