

LỜI CẢM ƠN

Trong quá trình thực hiện đề tài, nhóm đã nhận được nhiều sự góp ý, giúp đỡ của quý thầy cô và bạn bè để hoàn thiện đề tài và bài báo cáo này.

Nhóm xin gửi lời cảm ơn chân thành đến thầy ThS. Huỳnh Hoàng Hà giảng viên Bộ môn Kỹ thuật Máy tính - Viễn thông, người đã trực tiếp hỗ trợ, đưa ra những lời khuyên từ kinh nghiệm thực tiễn của mình, giải đáp thắc mắc và đưa ra những góp ý, chỉnh sửa kịp thời giúp nhóm khắc phục nhược điểm, phát huy những điểm mạnh của nhóm để quá trình thực hiện đề tài của nhóm luôn đi theo định hướng ban đầu của đề tài.

Nhóm cũng xin gửi lời cảm ơn chân thành các quý thầy cô trong khoa Điện - Điện Tử nói chung và ngành Công Nghệ Kỹ Thuật Máy Tính nói riêng, đã tận tình truyền đạt những kiến thức nền tảng, những điều kiện tốt để nhóm có thể tìm hiểu và thực hiện tốt đề tài.

Đề tài này được đúc kết từ sự tận tâm, nghiêm túc và nỗ lực hết mình của các thành viên trong nhóm. Bên cạnh những điểm tích cực mà nhóm thực hiện được cũng có tồn tại một vài sai sót, nhóm thực hiện hy vọng nhận được sự góp ý và phê bình của thầy cô và các bạn.

Người thực hiện đề tài

Vũ Trần Hoàng Sơn

Huỳnh Đặng Thanh Khiêm

TÓM TẮT

Đề tài “THIẾT KẾ VÀ THI CÔNG HỆ THỐNG KHÓA CỦA AN NINH” sử dụng vi điều khiển ESP32 làm bộ điều khiển trung tâm để giao tiếp, xử lý dữ liệu được thu thập từ các ngoại vi. ESP32-CAM để thu thập dữ liệu ở dạng video và hình ảnh. Bên cạnh đó, hệ thống cũng sử dụng các thiết bị có độ bảo mật cao hơn để xử lý, cho phép mở cửa như: RFID, vân tay, mật khẩu. Nhóm thực hiện sử dụng email làm phương tiện thông báo cho người dùng, những hoạt động vào cửa, cảnh báo sẽ được ESP32-CAM ghi lại hình ảnh và gửi email cho người dùng (khi cảnh báo đột nhập). Các dữ liệu, thông tin liên quan đến hệ thống (RFID, vân tay, lịch sử) sẽ được lưu trữ trên server và được hiển thị trên website.

MỤC LỤC

DANH MỤC BẢNG	vii
DANH MỤC HÌNH ẢNH	viii
CÁC TỪ VIẾT TẮT	xii
CHƯƠNG 1: TỔNG QUAN.....	1
1.1 ĐẶT VÂN ĐÈ.....	1
1.2 MỤC TIÊU ĐÈ TÀI.....	2
1.3 NỘI DUNG NGHIÊN CỨU	2
1.4 GIỚI HẠN ĐÈ TÀI.....	2
1.5 BỐ CỤC QUYỀN BÁO CÁO	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	4
2.1 GIỚI THIỆU VỀ NHẬN DẠNG VÂN TAY	4
2.2 CÁC CHUẨN TRUYỀN DỮ LIỆU	5
2.2.1 Chuẩn giao tiếp UART	5
2.2.2 Chuẩn giao tiếp I2C	6
2.3 TỔNG QUAN VỀ WEBSITE	7
2.3.1 Tổng quan về HTML, CSS	7
2.3.2 Ngôn ngữ lập trình backend PHP	8
2.3.3 Tổng quan về MySQL và phpMyAdmin	9
2.3.4 Giao thức HTTP request: POST và GET	11
2.3.5 Hàm mail() trong PHP và gửi mail	13
2.4 TỔNG QUAN VỀ WEB SERVER.....	14
2.5 TỔNG QUAN VỀ MQTT	15
2.5.1 Nguyên tắc truyền nhận dữ liệu MQTT.....	15
2.5.2 Hoạt động giao tiếp giữa máy chủ và khách.....	15

2.6 GIỚI THIỆU VỀ PHẦN CỨNG.....	16
2.6.1 Khóa chốt điện 12VDC LY-03.....	16
2.6.2 Cảm biến từ MC-38	16
2.6.3 Cảm biến vân tay AS608	17
2.6.4 Module RTC DS1307	18
2.6.5 LCD 20x4 I2C	19
2.6.6 Module MFRC-522 NFC 13.56MHz	19
2.6.7 Còi buzzer thụ động 5V	20
2.6.8 Relay SRD-12VDC-SL-C.....	21
2.6.9 Transistor NPN C1815.....	21
2.6.10 Vi điều khiển ESP32 NodeMCU	22
2.6.11 Vi điều khiển ESP32-CAM	24
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG	26
3.1 GIỚI THIỆU VÀ TÓM TẮT	26
3.1.1 Yêu cầu chung của hệ thống	26
3.1.2 Phương án thiết kế	26
3.2 SƠ ĐỒ KHỐI CỦA HỆ THỐNG	27
3.3 THIẾT KẾ CHI TIẾT KHÓA CỦA AN NINH	28
3.3.1 Khối xử lý trung tâm.....	29
3.3.2 Khối bàn phím	31
3.3.3 Khối xác thực	31
3.3.4 Khối hiển thị	32
3.3.5 Khối xử lý ảnh ESP32-CAM	33
3.3.6 Khối thời gian thực	34
3.3.7 Khối thực thi	34

3.3.8 Khối cảm biến	36
3.3.9 Khối nguồn	36
3.4 LUU ĐO HOAT ĐỘNG CỦA HỆ THỐNG	38
3.4.1 Xử lý truy cập	39
3.4.2 Các menu điều hướng	42
3.4.3 Camera	48
3.4.4 Báo động quên đóng cửa	50
3.4.5 Báo động sai mật khẩu.....	50
3.5 LẬP TRÌNH CHO ESP32.....	51
3.5.1 Công cụ lập trình.....	51
3.5.2 Một số xử lý cho ESP32 và ESP32-CAM	52
3.5.3 Vùng nhớ EEPROM sử dụng	54
CHƯƠNG 4: THI CÔNG HỆ THỐNG	56
4.1 THIẾT KẾ WEBSITE.....	56
4.1.1 Tạo tên miền	56
4.1.2 Thiết kế cơ sở dữ liệu	57
4.1.3 Lập trình các file xử lý dữ liệu bằng PHP	60
4.1.4 Thiết kế giao diện website	62
4.2 HƯỚNG DẪN SỬ DỤNG VÀ THAO TÁC.....	68
4.2.1 Tài liệu hướng dẫn kết nối	68
4.2.2 Thao tác thực hiện đăng ký tài khoản để giám sát hệ thống	71
4.3 THIẾT KẾ BOARD MẠCH	71
CHƯƠNG 5: KẾT QUẢ	74
5.1 KẾT QUẢ ĐẠT ĐƯỢC.....	74
5.2 KẾT QUẢ MÔ HÌNH THI CÔNG.....	74

5.2.1 Mạch in thực tế của nhóm.....	74
5.2.2 Mô hình sản phẩm của nhóm.....	75
5.2.3 Kết quả của sản phẩm.....	76
5.2.4 Kết quả phần mềm	82
5.3 NHẬN XÉT VÀ ĐÁNH GIÁ	86
CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	87
6.1 KẾT LUẬN	87
6.2 HƯỚNG PHÁT TRIỂN.....	87
PHỤ LỤC	89
TÀI LIỆU THAM KHẢO.....	98

DANH MỤC BẢNG

Bảng 2.1: Kiểu dữ liệu trong MySQL.....	10
Bảng 2.2: Thông số kỹ thuật của ESP32 NodeMCU	23
Bảng 2.3: Bảng thông số kỹ thuật Module ESP32-CAM	25
Bảng 3.1: Bảng mô tả kết nối với khói xử lý trung tâm.....	30
Bảng 3.2: Bảng tính toán công suất mạch.....	37
Bảng 3.3: Bảng địa chỉ lưu trữ thông tin vùng nhớ EEPROM	55
Bảng 4.1: Bảng lưu trữ dữ liệu “ref_table” trên MySQL	58
Bảng 4.2: Bảng lưu trữ dữ liệu “user_infor” trên MySQL	58
Bảng 4.3: Bảng lưu trữ dữ liệu “History<x>” trên MySQL	59
Bảng 4.4: Bảng lưu trữ dữ liệu RFID<x> trên MySQL.....	59
Bảng 4.5: Bảng lưu trữ dữ liệu Finger<x> trên MySQL	59
Bảng 4.6: Bảng lưu trữ dữ liệu “device_table” trên MySQL	60

DANH MỤC HÌNH ẢNH

Hình 2.1: Nguyên lý hoạt động của công nghệ quét vân tay [1]	4
Hình 2.2: Data Frame của chuẩn truyền thông UART	5
Hình 2.3: Mô hình Master và Slave I2C	6
Hình 2.4: Data Frame của chuẩn truyền thông I2C	6
Hình 2.5: Ví dụ về một trang HTML cơ bản	7
Hình 2.6: Giao diện HTML khi được định dạng bởi CSS	8
Hình 2.7: Ví dụ về HTTP request với body message json.....	12
Hình 2.8: Hoạt động của webserver.....	14
Hình 2.9: Ví dụ về hoạt động của mô hình MQTT	15
Hình 2.10: Chốt điện 12VDC.....	16
Hình 2.11: Cảm biến MC-38.....	17
Hình 2.12: Cảm biến vân tay AS608	17
Hình 2.13: Sơ đồ chân của cảm biến vân tay	17
Hình 2.14: Module thời gian thực RTC DS1307	18
Hình 2.15: Sơ đồ chân module thời gian thực RTC DS1307	18
Hình 2.16: Module LCD 20x04 tích hợp I2C	19
Hình 2.17: Module MFRC-522 NFC 13.56MHz.....	20
Hình 2.18 Sơ đồ chân của module MFRC-522 NFC 13.56MHz.....	20
Hình 2.19: Buzzer thụ động 5V	20
Hình 2.20: Relay SRD-12VDC-SL-C.....	21
Hình 2.21: Transistor NPN C1815.....	22
Hình 2.22: Sơ đồ chân của module ESP32 NodeMCU	23
Hình 2.23: Block Diagram của ESP32 NodeMCU.....	23
Hình 2.24: Sơ đồ chân của module ESP32-CAM.....	24
Hình 3.1: Sơ đồ khói của hệ thống.....	27
Hình 3.2: Sơ đồ nguyên lý toàn hệ thống.....	29
Hình 3.3: Sơ đồ nguyên lý ESP32 và kết nối ngoại vi.....	30
Hình 3.4: Sơ đồ nguyên lý của module keypad 4x4 và kết nối với khối xử lý....	31
Hình 3.5: Sơ đồ nguyên lý của MFRC522, AS608 và kết nối khối xử lý	32

Hình 3.6: Sơ đồ nguyên lý của LCD và kết nối khói xử lý.....	33
Hình 3.7: Sơ đồ nguyên lý ESP32-CAM và kết nối khói xử lý.....	33
Hình 3.8: Sơ đồ nguyên lý của module RTC DS1307 và kết nối khói xử lý.....	34
Hình 3.9: Sơ đồ nguyên lý khối thực thi và kết nối khói xử lý.....	34
Hình 3.10: Sơ đồ nguyên lý của khối cảm biến và kết nối khói xử lý	36
Hình 3.11: Mạch giảm áp LM2596.....	36
Hình 3.12: Adapter 12V-2A.....	38
Hình 3.13: Lưu đồ giải thuật chương trình chính	38
Hình 3.14: Lưu đồ giải thuật hàm xử lý xác nhận keypad	39
Hình 3.15: Lưu đồ giải thuật hàm xử lý xác nhận RFID	40
Hình 3.16: Lưu đồ giải thuật hàm xử lý xác nhận vân tay	41
Hình 3.17: Các menu điều hướng của hệ thống	42
Hình 3.18: Lưu đồ giải thuật cho hàm đổi mật khẩu keypad người dùng	43
Hình 3.19: Lưu đồ giải thuật cho hàm thêm thẻ RFID	44
Hình 3.20: Lưu đồ giải thuật cho hàm xóa RFID bằng mã định danh RFID.....	45
Hình 3.21: Lưu đồ giải thuật cho hàm xóa RFID bằng vị trí.....	46
Hình 3.22: Lưu đồ giải thuật cho hàm thêm vân tay.....	47
Hình 3.23: Lưu đồ giải thuật cho hàm xóa vân tay	48
Hình 3.24: Lưu đồ giải thuật xử lý trên ESP32-CAM	49
Hình 3.25: Lưu đồ báo động quên đóng cửa.....	50
Hình 3.26: Lưu đồ báo động sai mật khẩu	51
Hình 3.27: Giao diện cửa sổ lập trình trên Arduino IDE	52
Hình 4.1: Trang quản lý website trên 000webhost	56
Hình 4.2: Trang quản lý database MySQL trên 000webhost.....	57
Hình 4.3: Hoạt động cập nhật dữ liệu lên MySQL	60
Hình 4.4: Giao diện trang đăng ký tài khoản trên website.....	62
Hình 4.5: Giao diện trang đăng nhập trên website.....	63
Hình 4.6: Thanh chức năng được thiết kế trên trang web	64
Hình 4.7: Giao diện trang xem lịch sử ra vào cửa	64
Hình 4.8: Giao diện trang quản lý RFID và vân tay	65

Hình 4.9: Giao diện trang quản lý hình ảnh được gửi từ ESP32-CAM	66
Hình 4.10: Giao diện trang giám sát video camera.....	67
Hình 4.11: Giao diện trang hỗ trợ đổi mật khẩu tài khoản	68
Hình 4.12: Giao diện trang khôi phục mật khẩu cho tài khoản	68
Hình 4.13: Kết nối với access point để thiết lập WIFI cho hệ thống	69
Hình 4.14: Mục chọn cấu hình	70
Hình 4.15: Cấu hình email cảnh báo.....	70
Hình 4.16: Cấu hình wifi kết nối.....	70
Hình 4.17: Sơ đồ nguyên lý của thiết kế phần cứng của nhóm	72
Hình 4.18: Sơ đồ dây ở mặt dưới	72
Hình 4.19: Sơ đồ dây ở mặt trên	72
Hình 4.20: Thiết kế mạch nhìn từ góc độ 3D.....	73
Hình 5.1: Mạch thực tế của nhóm sau khi thi công	75
Hình 5.2: Mô hình sản phẩm của nhóm	76
Hình 5.3: Menu truy cập chờ quét KEY xác nhận	77
Hình 5.4: Menu Admin điều hướng tùy chọn	77
Hình 5.5: Menu Password Option	78
Hình 5.6: Màn hình yêu cầu nhập mật khẩu cũ	78
Hình 5.7: Màn hình yêu cầu nhập mật khẩu mới	79
Hình 5.8: Menu RFID Option	79
Hình 5.9: Thêm thẻ RFID mới	80
Hình 5.10: Màn hình xóa thẻ NFC.....	80
Hình 5.11: Menu Finger Option.....	81
Hình 5.12: Màn hình thêm vân tay.....	81
Hình 5.13: Màn hình xóa vân tay.....	82
Hình 5.14: Kết quả hiển thị 25 thông điệp trên trang web.....	83
Hình 5.15: Email cảnh báo được gửi từ hệ thống	83
Hình 5.16: Bảng thông tin KEY lưu trữ và cập nhật	84
Hình 5.17: Lưu trữ các hình ảnh mở cửa trên trang web	84
Hình 5.18: Video stream giám sát hệ thống thông qua camera	85

Hình 5.19: Tên người dùng trước và sau khi đổi 85

CÁC TỪ VIẾT TẮT

Viết tắt	Mô tả
IoT	Internet of Things
RFID	Radio Frequency Identification Detection
UART	Universal Asynchronous Receiver/Transmitter
LSB	Least Significant Bit
MSB	Most Significant Bit
I2C	Inter-Integrated Circuit
SDA	Serial Data
SCL	Serial Clock
ACK	Acknowledgement
SPI	Serial Peripheral Interface
HTTPS	Hyper Text Transfer Protocol Security
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
JS	Javascript
PHP	Hypertext Preprocessor
SQL	Structured Query Language
URL	Uniform Resource Locator
TSMC	Taiwan Semiconductor Manufacturing Company
IDE	Integrated Development Environment
FTP	File Transfer Protocol
CSDL	Cơ Sở Dữ Liệu
LED	Light Emitting Diode
AP	Access Point
RTC	Real Time Clock
RF	Radio Frequency
IC	Integrated Circuit
SSID	Service Set IDentifier

FAR	False Accept Rate
FRR	False Reject Rate
MQTT	Message Queuing Telemetry Transport
NFC	Near-Field Communications
API	Application Programming Interface

CHƯƠNG 1: TỔNG QUAN

1.1 ĐẶT VẤN ĐỀ

Trải qua mùa đại dịch covid-19, nhu cầu về các vật tư, vật dụng thiết yếu về cuộc sống ngày càng tăng cao, tuy nhiên do sự hoành hành của đại dịch mà nhiều cá nhân không thể tìm được nguồn thu nhập để duy trì cuộc sống. Để giải quyết vấn đề này, nhà nước, tổ chức cá nhân đã tài trợ, quyên góp để góp phần giúp đỡ người dân hòa nhập lại với cuộc sống. Tuy nhiên, một vài cá nhân lại có nhu cầu cao hơn và dẫn đến các hành vi đi ngược lại với pháp luật như trộm cắp. Vấn đề này càng đáng lo ngại hơn khi biện pháp bảo vệ tài sản đầu tiên là khóa cửa lại quá thô sơ. Thủ đoạn phá, bẻ các loại khóa truyền thống thì ngày càng được chia sẻ rộng rãi trên các phương tiện truyền thông và các kẻ trộm lại có thể dễ dàng học được. Nhu cầu về một hệ thống khóa cửa có nhiều tính năng bảo mật thay thế cho khóa cửa truyền thống ngày càng tăng cao. Vì lý do này, nhóm thực hiện đã quyết định thực hiện đề tài “**THIẾT KẾ VÀ THI CÔNG HỆ THỐNG QUÁ CỬA AN NINH**”.

Hiện nay, khóa cửa tích hợp nhiều biện pháp bảo mật không còn quá xa lạ với mọi người, tuy nhiên hạn chế của các loại khóa này trên thị trường là thiếu đi 1 hệ thống giám sát để tăng cường tính bảo mật cho sản phẩm.

Với mục tiêu cung cấp một giải pháp bảo mật tốt hơn giúp bảo vệ nhà cửa và tài sản của người dân nhóm đã thực hiện đã tiến hành thêm một hệ thống có thể giám sát và các chức năng để đáp ứng nhu cầu của người dùng.

- Có hệ thống camera giám sát đi kèm với sản phẩm.
- Có khả năng chụp ảnh những người tương tác với các biện pháp bảo mật.
- Ghi lại lịch trình ra vào cửa của người dùng.
- Gửi email cho người dùng thông tin ra vào cửa, báo động.
- Giám sát trực tiếp camera thông qua trang web.

1.2 MỤC TIÊU ĐỀ TÀI

Mục tiêu của đề tài là xây dựng một hệ thống có độ bảo mật cao kết hợp 3 phương pháp mở khóa. Tạo ra một sản phẩm nhỏ gọn, dễ lắp đặt và sử dụng. Hệ thống giúp các gia đình có thể giám sát căn hộ nhà mình, giúp bảo vệ được tài sản tránh được các rủi ro về mất tài sản. Hệ thống có độ ổn định cao có thể hoạt động xuyên suốt và đảm bảo được độ chính xác.

1.3 NỘI DUNG NGHIÊN CỨU

Nội dung 1: Tổng quan về thực trạng hiện nay và các hệ thống khóa cửa an ninh.

Nội dung 2: Thiết kế khóa cửa với 3 phương pháp mở cửa: RFID, vân tay, keypad.

Nội dung 3: Trao đổi dữ liệu thông qua Json và UART.

Nội dung 4: Tổng quan về ESP32-CAM và Camera webserver (local IP).

Nội dung 5: Trang web tĩnh HTML, Javascript và CSS.

Nội dung 6: Lập trình PHP, cơ sở dữ liệu phpmyadmin.

Nội dung 7: Giao thức HTTP request: POST và GET.

Nội dung 8: Gửi mail thông qua PHP (server host mail).

Nội dung 9: Thực hiện viết chương trình cho mạch điều khiển.

Nội dung 10: Thi công và viết báo cáo.

1.4 GIỚI HẠN ĐỀ TÀI

Vì một số yếu tố (kiến thức kỹ thuật, tài chính) mà nội dung thực hiện đề tài này chỉ thực hiện trong phạm vi sau:

- Mở khóa sử dụng 3 phương pháp: RFID, vân tay, password.
- Có báo động qua thiết bị âm thanh khi có đột nhập hoặc quên đóng cửa.
- Đèn báo mở/đóng cửa.
- Lưu trữ các thông tin thẻ RFID, vân tay, password trên bộ nhớ flash của ESP32.
- Hệ thống mất điện không bị mất dữ liệu đã tồn tại trước đó.
- Có khả năng giám sát thông tin qua trang web.

- Gửi cảnh báo thông qua email.
- Thêm/xóa mật khẩu cho hệ thống.
- Có camera giám sát, chụp lại hoạt động vào cửa, cảnh báo đột nhập.

1.5 BỘ CỤC QUYỀN BÁO CÁO

Đồ án thực hiện bao gồm 6 chương:

Chương 1: Tổng quan

Giới thiệu tổng quan, các mục tiêu, giới hạn của đề tài thực hiện.

Chương 2: Cơ sở lý thuyết

Giới thiệu các lý thuyết liên quan đến đề tài. Giới thiệu các phần cứng, lựa chọn linh kiện thiết bị được sử dụng, giới thiệu về các chức năng, thông số kỹ thuật của các linh kiện.

Chương 3: Thiết kế hệ thống

Trình bày sơ đồ khái của hệ thống, giải thích hoạt động của các khái, tính toán các thông số các linh kiện đưa vào hệ thống. Trình bày sơ đồ nguyên lý của hệ thống. Trình bày thiết kế hệ thống.

Chương 4: Thi công hệ thống

Chương này trình bày nội dung việc tiến hành thi công làm mạch cho hệ thống. Thiết kế tạo ra trang web để giao tiếp trực quan với người dùng. Thực hiện viết chương trình, thiết kế giao diện cho trang web.

Chương 5: Kết quả

Trình bày tất cả các kết quả đã đạt được, nhận xét, đánh giá về hệ thống, đưa ra các hình ảnh thực tiễn, hình ảnh hoạt động của hệ thống.

Chương 6: Kết luận và hướng phát triển

Chương này đưa ra kết luận dựa trên các kết quả trong chương trước và hướng phát triển của đề tài.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

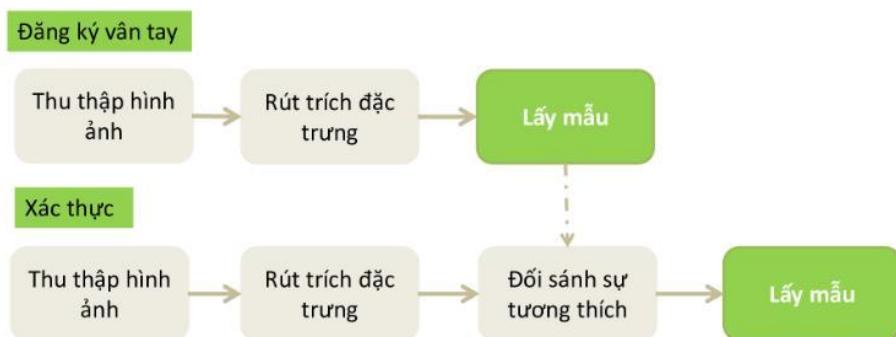
2.1 GIỚI THIỆU VỀ NHẬN DẠNG VÂN TAY

Vân tay theo nghĩa hẹp của nó là một ấn tượng để lại bởi các đường vân masing của ngón tay người. Dấu vân tay dễ dàng lắng đọng trên các bề mặt phù hợp (như thủy tinh hoặc kim loại hoặc đá đánh bóng) bởi các chất tiết mồ hôi tự nhiên từ các tuyến eccrine có trong các đường vân biếu bì.

Dấu vân tay của con người là độc đáo và không thay đổi suốt cuộc đời. Vân tay hình thành từ các gai da trên lớp biểu bì và được tạo ra từ tuần thứ 13 đến tuần thứ 19 trong thai kỳ. Vân tay được phóng đại nhưng giữ nguyên cấu trúc ban đầu từ khi sinh ra cho đến khi già. Tái tạo vân tay sau tổn thương là điều thông thường, trừ khi có tổn thương sâu và sẹo vết hủy hoại hoàn toàn. Mỗi người có dấu vân tay riêng biệt do tác động của hệ thống gen di truyền và môi trường. [1]

Công nghệ nhận dạng dấu vân tay hoạt động bằng cách quét hình ảnh ngón tay và so sánh các đặc điểm với dữ liệu đã lưu trữ trong hệ thống. Quá trình này chuyển đổi hình ảnh thành dữ liệu số và sử dụng thuật toán để xử lý thông tin. Sau khi quá trình xử lý hoàn tất, hệ thống đưa ra thông báo xác định xem dấu vân tay là hợp lệ hay không hợp lệ.. [1]

Xử lý nhận dạng vân tay bao gồm hai quy trình chủ yếu: đăng ký và xác thực



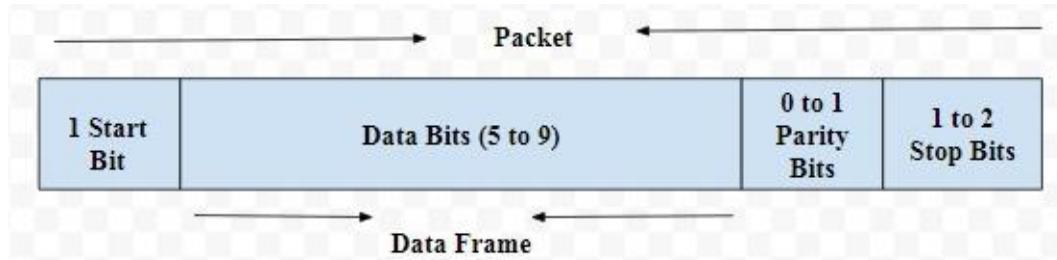
Hình 2.1: Nguyên lý hoạt động của công nghệ quét vân tay [1]

2.2 CÁC CHUẨN TRUYỀN DỮ LIỆU

2.2.1 Chuẩn giao tiếp UART

UART là kiểu truyền thông nối tiếp không đồng bộ. Chức năng chính thường là truyền dữ liệu nối tiếp (Serial).

Dữ liệu truyền trong UART sẽ chia ra thành các gói (Packet) gồm: 1 start bits, 5-9 data bits (Tùy theo thiết đặt), 0-1 parity bits và 1-2 stop bits.



Hình 2.2: Data Frame của chuẩn truyền thông UART

Bit bắt đầu (start bit): được gọi là bit đồng bộ hóa đặt trước dữ liệu. Để bắt đầu truyền dữ liệu đường truyền sẽ được kéo từ điện áp từ mức cao (1) xuống mức điện áp thấp (0).

Khung dữ liệu (Data Bits): Các bit dữ liệu giao tiếp giữa bên gửi và bên nhận. Độ dài khung dữ liệu bao gồm từ 5-9 bits (nếu có bit parity thì tối đa là 8 bits). Thường được truyền với bit có trọng số thấp (LSB) trước.

Parity bit: Bit dùng để kiểm tra đảm bảo dữ liệu truyền có đúng hay không.

Bit dừng (stop bit): Bit dừng đặt ở cuối dữ liệu truyền. Thông thường bit này có độ dài 1 bit. Báo hiệu việc truyền kết thúc.

- **Ưu điểm:**

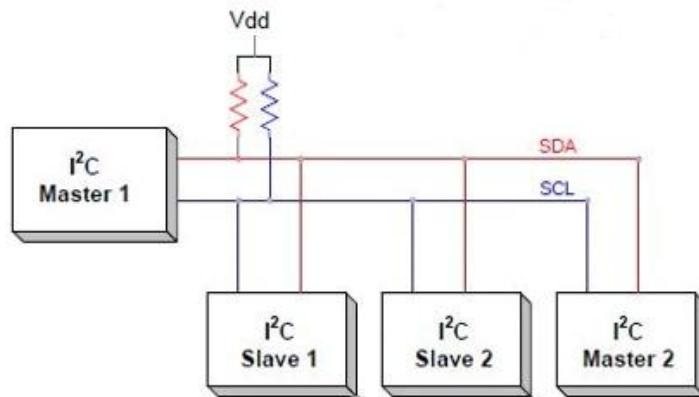
- Không cần tín hiệu đồng bộ.
- Có thể có hoặc không bit chẵn lẻ để kiểm tra lỗi.
- Cấu trúc dữ liệu nhận có thể thay đổi.

- **Khuyết điểm:**

- Kích thước dữ liệu giới hạn (9 bits).
- Truyền với khoảng cách ngắn (trên board mạch).
- Tốc độ truyền UART phải nằm trong 10% giữa thiết gửi và nhận.

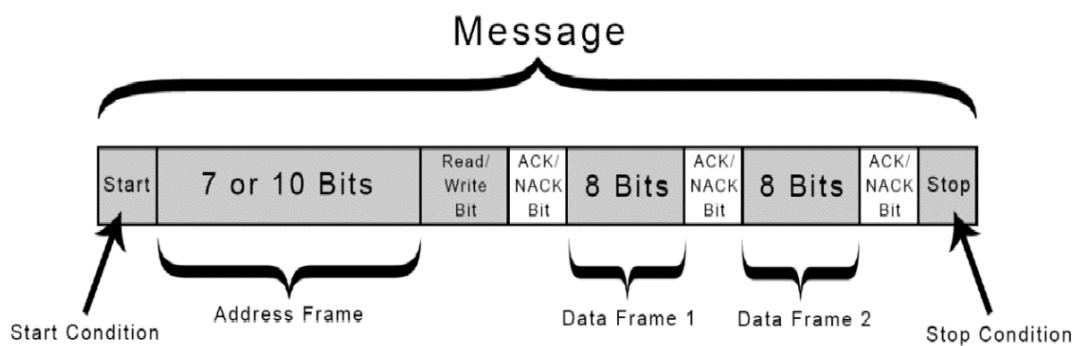
2.2.2 Chuẩn giao tiếp I2C

I2C giao thức này cho phép truyền dữ liệu từ một master với nhiều slave. Chỉ sử dụng 2 đường truyền dữ liệu là SDA và SCL. Đây là một loại giao thức giao tiếp nối tiếp đồng bộ, gửi nhận dữ liệu thông qua tín hiệu xung SCL. Là một chuẩn giao tiếp thông qua địa chỉ.



Hình 2.3: Mô hình Master và Slave I2C

Với I2C, dữ liệu được truyền trong các tin nhắn. Tin nhắn được chia thành các khung dữ liệu. Mỗi tin nhắn có một khung địa chỉ chứa địa chỉ nhị phân của địa chỉ slave và một hoặc nhiều khung dữ liệu chứa dữ liệu đang được truyền.



Hình 2.4: Data Frame của chuẩn truyền thông I2C

- Ưu điểm:

- Hỗ trợ nhiều thiết bị chủ, tớ (master/slave)
- Có bit ACK/NACK xác nhận truyền gửi thành công.
- Phần cứng ít phức tạp.

- Nhược điểm:

- Kích thước khung dữ liệu giới hạn ở 8 bits.
- Tốc độ truyền chậm khi so với SPI.

2.3 TỔNG QUAN VỀ WEBSITE

Website là 1 hoặc nhiều trang nội dung chứa các nội dung tĩnh như văn bản, hình ảnh, liên kết hay cũng có thể chứa các nội dung động như các animation, video... và chỉ có thể truy cập bằng trình duyệt web bằng giao thức HTTP hoặc HTTPS thông qua Internet.

Nội dung trên các trang web được xây dựng dựa trên 3 ngôn ngữ chính là HTML, CSS và JS.

2.3.1 Tổng quan về HTML, CSS

HTML (Hypertext Markup Language) hay ngôn ngữ đánh dấu siêu văn bản, là ngôn ngữ dùng để xây dựng nội dung cho một trang web. Bằng cách sử dụng các thẻ(tags), các nội dung sẽ được đánh dấu bằng các thẻ khác nhau. Điều này làm cho trình duyệt có thể hiểu được nội dung của trang web và giúp nó hiển thị lên đó. Tuy nhiên, HTML chỉ dừng lại ở việc cung cấp nội dung, hình ảnh, cấu trúc cơ bản cho trang web, nếu trang web chỉ được tạo nên từ HTML, nó chỉ như một trang văn bản có nội dung được sắp xếp 1 cách rành mạch, rõ ràng nhưng nó không có định dạng.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Document</title>
6  </head>
7  <body>
8
9  </body>
10 </html>
```

Hình 2.5: Ví dụ về một trang HTML cơ bản

CSS (Cascading Style Sheets) là 1 ngôn ngữ được tạo ra nhằm mục đích định dạng cho trang web được viết bằng HTML. CSS có khả năng thay đổi kích thước, màu sắc, vị trí, các hiệu ứng cho các phần tử trong trang web, làm cho trang web có thể được hiển thị 1 cách sinh động, trực quan, thu hút người truy cập. Để làm

được điều này, CSS chọn ra các trường ID, CLASS, tags của nội dung cần được định dạng và tiến hành thêm các thuộc tính và giá trị thích hợp.



Hình 2.6: Giao diện HTML khi được định dạng bởi CSS

Cấu trúc và bố cục của CSS

```
<vùng chọn> {  <thuộc tính> : <giá trị>;  <thuộc tính>: <giá trị>;  ..... }
```

Vùng chọn tương ứng là một thuộc tính của thẻ HMTL xác định (class hoặc id). Thuộc tính định dạng sẽ có giá trị riêng dùng để định dạng được quy định bởi W3C.

2.3.2 Ngôn ngữ lập trình backend PHP

PHP(Hypertext Preprocessor) là một ngôn ngữ kịch bản được lập trình phía máy chủ (backend) phổ biến được sử dụng chủ yếu để phát triển các ứng dụng web động nhằm xử lý các yêu cầu từ máy khách(client) và tương tác với máy chủ. PHP là ngôn ngữ được thiết kế đặc biệt để dễ dàng tích hợp với HTML và có khả năng tạo nội dung động trên trang web.

Hiện nay, PHP được sử dụng và ứng dụng rộng rãi trong các trang web vì là 1 ngôn ngữ dễ sử dụng và có cú pháp gần giống với các loại ngôn ngữ khác. Vì vậy, nhiều kỹ sư đã tạo nên các bộ thư viện và nhiều hàm được xây dựng sẵn nhằm tạo điều kiện cho người dùng dễ dàng tiếp cận và sử dụng. Bên cạnh đó, PHP còn đem lại khả năng tương thích với nhiều nền tảng hệ điều hành. Đây là 1 điểm mạnh mà khó có 1 ngôn ngữ nào có thể so sánh được.

2.3.3 Tổng quan về MySQL và phpMyAdmin

MySQL và phpMyAdmin là hai công cụ thường được sử dụng để quản lý cơ sở dữ liệu trong môi trường PHP.

MySQL là một hệ quản trị cơ sở dữ liệu phổ biến và mạnh mẽ. Nó được sử dụng để lưu trữ và quản lý dữ liệu trong các ứng dụng web. MySQL có khả năng xử lý dữ liệu có cấu trúc và quan hệ thông qua các bảng, hàng và cột. MySQL cung cấp các tính năng như truy vấn dữ liệu, thêm, sửa đổi và xóa dữ liệu, tạo bảng, quản lý quyền truy cập và nhiều tính năng khác để làm việc với cơ sở dữ liệu.

phpMyAdmin là một công cụ quản lý cơ sở dữ liệu MySQL dựa trên giao diện web. Nó cung cấp một giao diện đồ họa cho phép bạn thao tác với cơ sở dữ liệu MySQL một cách dễ dàng và thuận tiện. Với phpMyAdmin, bạn có thể thực hiện các tác vụ như tạo, sửa đổi và xóa cơ sở dữ liệu, bảng, truy vấn SQL, nhập và xuất dữ liệu, quản lý quyền truy cập người dùng và nhiều tác vụ quản lý khác liên quan đến cơ sở dữ liệu MySQL.

Cả MySQL và phpMyAdmin thường được sử dụng kết hợp với PHP để phát triển các ứng dụng web. PHP cung cấp các chức năng và thư viện để kết nối và tương tác với cơ sở dữ liệu MySQL thông qua các truy vấn SQL. phpMyAdmin giúp quản lý và thao tác với cơ sở dữ liệu MySQL một cách trực quan và thuận tiện qua giao diện web.

Sự kết hợp giữa MySQL, PHP và phpMyAdmin cung cấp một môi trường phát triển toàn diện cho việc xây dựng và quản lý ứng dụng web dựa trên cơ sở dữ liệu.

Những định nghĩa cơ bản

Cơ sở dữ liệu SQL lưu trữ các bảng dữ liệu của người dùng tạo ra. Một cơ sở dữ liệu có thể có một hoặc nhiều bảng. Sử dụng các câu lệnh truy vấn Query tác động lên CSDL.

Bảng: là 1 bảng giá trị nằm trong CSDL. Bảng được dùng để lưu trữ các trường dữ liệu.

Một số thuật ngữ:

- NULL: giá trị cho phép rỗng.
- AUTO_INCREMENT: Cho phép giá trị tăng dần (Tự động).

- PRIMARY KEY: cho phép nó là khóa chính trong bảng.

Một số kiểu dữ liệu trong CSDL MySQL:

Bảng 2.1: Kiểu dữ liệu trong MySQL

Kiểu dữ liệu	Mô tả
VARCHAR	Độ dài tối đa 8000 ký tự (dữ liệu không phải Unicode có độ dài tùy biến)
text	Dữ liệu có độ dài tùy biến với độ dài tối đa là 2.147.483.647 ký tự.
INT	Giá trị -2,147,483,648 đến 2,147,483,647
datetime	Giá trị từ Jan 1, 1753 đến Dec 31, 9999

Những cú pháp cơ bản trong SQL

Việc tác động đến các bảng hay cấu trúc trong SQL cần phải được thông qua các truy vấn SQL (có thể sử dụng các công cụ hỗ trợ), một vài truy vấn cơ bản:

- **Tạo một cơ sở dữ liệu:**

- Create DATABASE <tên CSDL>

- **Tạo một bảng:**

- CREATE TABLE <tên bảng> (<tên cột> <mô tả> ... <tên cột n> <mô tả n>)

- **Thêm giá trị vào bảng:**

- INSERT INTO <tên bảng> (<tên cột>) VALUES (<giá trị tương ứng>)

- **Truy xuất dữ liệu của bảng:**

- SELECT <trường 1>,<trường 2>,<trường 3>... FROM <tên bảng> WHERE <điều kiện lọc> ORDER BY <tên trường> ASC|DESC LIMIT start, limit.

- **Thêm giá trị vào bảng:**

- INSERT INTO <tên bảng> (<tên cột>) VALUES(<giá trị tương ứng>)

- **Cập nhật dữ liệu trong bảng:**

- Update <tên bảng> SET <tên cột> = <Giá trị mới> Where <điều kiện>
- Nếu không có ràng buộc điều kiện, Bảng sẽ cập nhật toàn bộ giá trị cho cột đó giống nhau.

- **Xóa dữ liệu trong bảng:**

- DELETE FROM <tên bảng> WHERE <điều kiện>
- Nếu không có ràng buộc điều kiện, tất cả dữ liệu record trong bảng sẽ bị xóa.

2.3.4 Giao thức HTTP request: POST và GET

2.3.4.1 Giao thức HTTP

Giao thức HTTP (Hypertext Transfer Protocol) là một giao thức truyền thông phổ biến được sử dụng để trao đổi thông tin giữa máy chủ và máy khách trên web. Trong HTTP, có hai phương thức chính được sử dụng để gửi yêu cầu từ máy khách đến máy chủ: POST và GET.

Cấu trúc HTTP request

HTTP Request có cấu tạo gồm ba phần chính. Đó là request line, header và message body.

a) Request Line

Đây là dòng đầu tiên của HTTP Request, với ba loại chính là method, path (hay URL) và HTTP version. Cụ thể:

- Method: gồm nhiều loại nhưng phổ biến nhất là GET và POST. Trong đó, phương thức GET thường dùng để yêu cầu các tài nguyên cung cấp trong URL. POST có chức năng gửi dữ liệu, có tính bảo mật cao hơn.
- Path (URL): có tác dụng định danh các nguồn tài nguyên được yêu cầu bởi client và bắt buộc phải có dấu “/”.
- HTTP version: Đây là phiên bản HTTP được sử dụng, trong đó phổ biến nhất là HTTP/1.0 hay HTTP/1.1.

b) Headers

Yếu tố thứ hai góp phần làm hình thành HTTP Request đó là các header. Thông tin được bổ sung sẽ truyền tải giữa cả máy chủ và máy khách, chẳng hạn như cookie, thông tin về ủy quyền, tác nhân người dùng... Tương tự một HTTP Request, header sẽ phân biệt chữ thường và chữ hoa, theo sau đó là dấu “.” và một giá trị.

- Một số headers thường dùng: application/json, text/application, application/x-www-form-urlencoded, multipath/form-data,...

c) Message Body

Yếu tố thứ ba được đề cập đến đó là message body. Máy chủ dùng nội dung thư để cung cấp những thông tin cần thiết nhất đến với máy khách. Message body có chứa các dòng yêu cầu, thông tin, dòng trống, tiêu đề, và nội dung. Trong đó, yếu tố nội dung sẽ tùy chọn.

```
POST /update-sensor HTTP/1.1
Host: example.com
{api_key: "api", sensor_name: "name", temperature: value1, humidity: value2,
pressure: value3}
Content-Type: application/json
```

Hình 2.7: Ví dụ về HTTP request với body message json

2.3.4.2 HTTP POST và GET request

Phương thức POST request là một phương thức được sử dụng để gửi dữ liệu từ máy khách (client) lên máy chủ (server) để xử lý. Khi sử dụng POST request, dữ liệu được gửi đi sẽ được đính kèm trong phần thân (body) của yêu cầu HTTP. Thông thường, khi gửi một POST request, dữ liệu được đóng gói và gửi dưới dạng các cặp khóa-giá trị (key-value pairs) hoặc đối tượng JSON. Các dữ liệu này thường được gửi lên máy chủ để thực hiện các hành động như tạo, cập nhật, hoặc xóa dữ liệu trên máy chủ. Khi máy chủ nhận được POST request, nó sẽ xử lý dữ liệu được gửi lên và phản hồi với kết quả tương ứng. Phương thức POST request cho phép gửi dữ liệu lớn và đa dạng, và thường được sử dụng trong các trường hợp cần truyền nhiều thông tin hoặc khi dữ liệu cần được bảo mật hơn so với phương thức GET request.

Phương thức GET request là một phương thức được sử dụng để yêu cầu dữ liệu từ máy chủ bằng cách gửi các thông số trên URL (Uniform Resource Locator). Khi sử dụng GET request, các thông số được gắn kết trực tiếp vào URL và được mã hóa. Khi máy khách (client) gửi một GET request, nó yêu cầu máy chủ (server) trả về một tài nguyên cụ thể, chẳng hạn như một trang HTML, một hình ảnh, hoặc dữ liệu từ cơ sở dữ liệu. Thông thường, dữ liệu trả về từ máy chủ sẽ được hiển thị

trực tiếp trong trình duyệt web. Phương thức GET request thường được sử dụng khi người dùng truy cập và yêu cầu hiển thị thông tin từ máy chủ một cách đơn giản và nhanh chóng.

2.3.5 Hàm mail() trong PHP và gửi mail

PHP mail() là một hàm PHP được tích hợp sẵn sử dụng để gửi email. PHP mail() function kích hoạt chương trình Sendmail ở trên server hỗ trợ, thường được cấu hình bởi system admin. Chương trình này giúp bạn gửi email được.

Để gửi được mail chúng ta cần tạo một file PHP và gửi nội dung mong muốn qua PHP và sử dụng hàm PHP mail(). Khi truy cập vào file PHP thì lập tức mail sẽ được gửi.

Hàm PHP mail() có cú pháp cơ bản sau:

```
mail($to_email_address, $subject, $message, [$headers], [$parameters]);
```

Với:

\$to_email_address là địa chỉ email của người nhận

\$subject là tiêu đề email

\$message là nội dung bạn muốn gửi

\$headers là tham số tùy chọn, nó có thể được sử dụng để bổ sung thông tin như cc, bcc

Ngoài ra ta có thể thêm tệp đính kèm với email bằng cách sử dụng header của http request để mã hóa nội dung của file và gửi đi. Với việc sử dụng kết hợp các phương thức POST và GET ta có thể nhận dữ liệu từ các client và gửi lên mail server. [2]

Ví dụ về gửi nội dung đính kèm với email bằng hàm mail() [2]:

```
<?php  
    $to = "xyz@somedomain.com";  
    $subject = "Gửi email có đính kèm file";  
    $message = "Đây là nội dung email.";  
    //header chính  
    $header = "From:xyz@somedomain.com\r\n";  
    $header .= "MIME-Version: 1.0\r\n";  
    $header .= "Content-Type: multipart/mixed; "  
    $header .= "boundary=$num\r\n";  
    $header .= "--$num\r\n";  
  
    //Nội dung email
```

```

$header .= "Content-Type: text/plain\r\n";
$header .= "Content-Transfer-Encoding:8bit\r\n\r\n";
$header .= "$message\r\n";
$header .= "--$num\r\n";

//Nội dung đính kèm
$header .= "Content-Type: multipart/mixed; ";
$header .= "name=\"test.txt\"\r\n";
$header .= "Content-Transfer-Encoding:base64\r\n";
$header .= "Content-Disposition:attachment; ";
$header .= "filename=\"text.txt\"\r\n";
$header .= "$encoded_content\r\n";
$header .= "--$num--"; //Kết thúc các phân đoạn

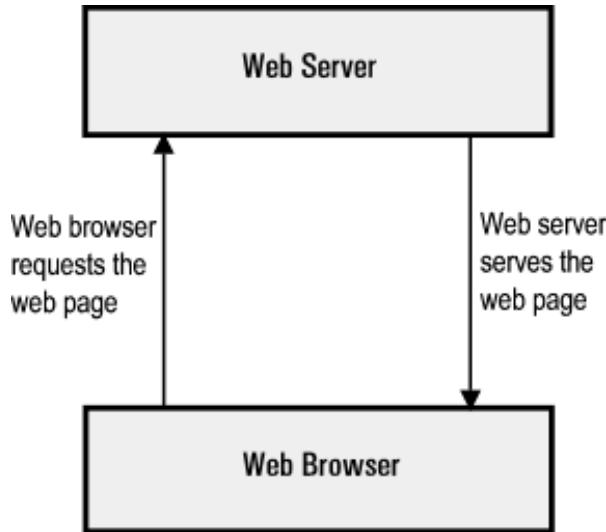
//Gửi mail
$success = mail($to,$subject,$message,$header);
?>

```

2.4 TỔNG QUAN VỀ WEB SERVER

Webserver là một máy chủ cài đặt những ứng dụng web. Có khả năng tiếp nhận các request từ trình duyệt hoặc các ứng dụng. Server xử lý các yêu cầu và trả về các trang web tương ứng đến client thông qua các giao thức mạng như HTTP, websocket,...

Khi bạn tương tác với một trang web (xem, gửi mail, nhận tin nhắn,...) thì tức là đang tương tác với một server. Server sẽ xử lý các request và phản hồi với client.



Hình 2.8: Hoạt động của webserver

Các ứng dụng của webserver phổ biến như: Dịch vụ hosting, các dịch vụ API để tương tác với các hệ thống, dịch vụ stream video online, lưu trữ các tập tin,...

2.5 TỔNG QUAN VỀ MQTT

MQTT (Message Queuing Telemetry Transport) là một giao thức nhắn tin dựa trên các tiêu chuẩn giao tiếp với máy chủ. Hoạt động theo mô hình đám mây, nhận tín hiệu dữ liệu văn bản (json, text,...) được lưu trên các broker (tạm dịch là máy chủ môi giới). MQTT được ứng dụng nhiều trong các mô hình IoT vì giao thức này dễ triển khai và giao tiếp dữ liệu IoT một cách hiệu quả.

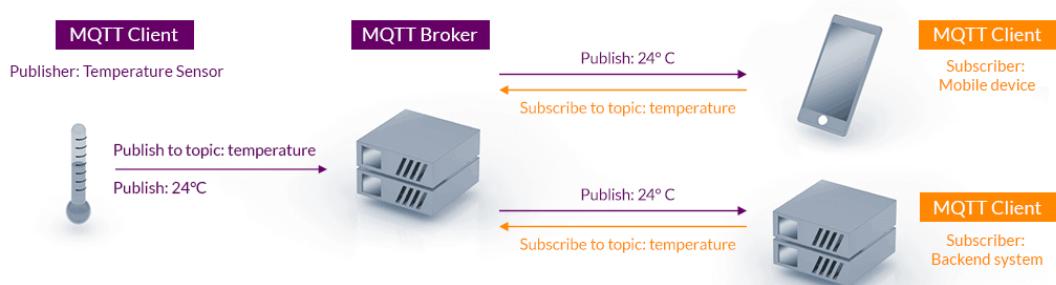
2.5.1 Nguyên tắc truyền nhận dữ liệu MQTT

Giao thức MQTT hoạt động theo mô hình xuất bản (publish) và đăng ký (subscribe), trong đó thông điệp được gửi từ một bên xuất bản đến một bên đăng ký thông qua một trình truyền tải thông điệp. Trong mô hình truyền thống, máy khách và máy chủ giao tiếp trực tiếp, nhưng MQTT tách riêng bên gửi thông điệp và bên nhận thông điệp. Trình truyền tải thông điệp đóng vai trò trung gian, nhận thông điệp từ bên gửi và chuyển chúng đến bên nhận một cách chính xác.

2.5.2 Hoạt động giao tiếp giữa máy chủ và khách

Dưới đây là tổng quan về cách thức hoạt động của MQTT.

1. Máy khách MQTT thiết lập kết nối với quá trình truyền tải MQTT.
2. Sau khi kết nối, máy khách có thể xuất bản/đăng ký thông điệp trên MQTT.
3. Sau khi trình truyền tải MQTT, trình truyền tải sẽ chuyển các thông điệp đó đến máy khách đã xuất bản/đăng ký.



Hình 2.9: Ví dụ về hoạt động của mô hình MQTT

Chủ đề (Topic) trong MQTT: Thuật ngữ “chủ đề” đề cập đến các từ khóa trình truyền tải MQTT sử dụng để lọc các thông điệp cho máy khách MQTT

Quá trình xuất bản qua MQTT: Máy khách MQTT xuất bản các thông điệp có chứa chủ đề và dữ liệu dưới định dạng byte. Máy khách xác định định dạng dữ liệu như dữ liệu văn bản, dữ liệu nhị phân, tệp XML hoặc JSON.

Quá trình đăng ký qua MQTT: Máy khách MQTT gửi một thông điệp đăng ký đến trình truyền tải MQTT để nhận thông điệp về các chủ đề quan tâm. Thông điệp này chứa một mã định danh duy nhất và một danh sách đăng ký.

2.6 GIỚI THIỆU VỀ PHẦN CỨNG

2.6.1 Khóa chốt điện 12VDC LY-03

Khóa chốt điện là một ổ khóa được kích đóng/mở bằng điện được sử dụng nhiều trong nhà thông minh hoặc các hệ thống an ninh. Sử dụng điện áp 12VDC để cấp nguồn cho chốt hoạt động. Chốt được sử dụng là loại thường đóng.



Hình 2.10: Chốt điện 12VDC

Thông số kỹ thuật của khóa chốt điện [3]:

- Vật liệu: Thép không giòi.
- Dòng điện làm việc: 0.8A.
- Công suất tiêu thụ: $12V \times 0.8A = 9.6W$.

2.6.2 Cảm biến từ MC-38

Cảm biến từ MC-38 là loại cảm biến có dây được gắn trên cửa, tủ hoạt động như 1 công tắc từ. Được ứng dụng trong việc chống trộm trong tủ, cửa, két sắt... Trong đề tài này, cảm biến từ hoạt động như 1 công tắc tự động, khi để 2 mặt gần

nhau công tắc bên trong được đóng lại (loại thường mở), dựa vào tín hiệu này để đưa ra các điều khiển phù hợp.



Hình 2.11: Cảm biến MC-38

Thông số kỹ thuật của cảm biến MC38 [4]:

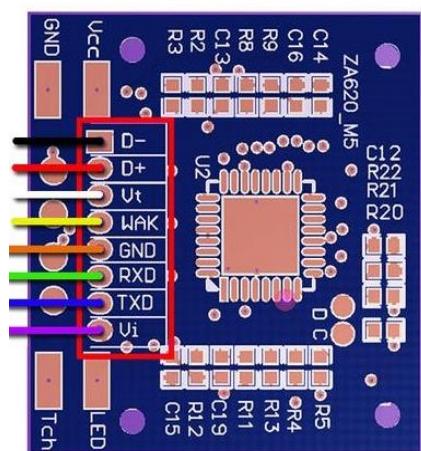
- Khoảng cách hoạt động: $18\text{mm} \pm 6\text{mm}$.
- Điện áp giữa 2 tiếp điểm : 100 VDC (Max).
- Dòng tiêu thụ : 100mA (Max).
- Dạng ngõ ra : thường mở.

2.6.3 Cảm biến vân tay AS608

Cảm biến sử dụng giao thức UART để giao tiếp với vi điều khiển, dễ sử dụng và được ứng dụng trong các hệ thống an ninh hoặc điểm danh.



Hình 2.12: Cảm biến vân tay AS608



Hình 2.13: Sơ đồ chân của cảm biến

vân tay

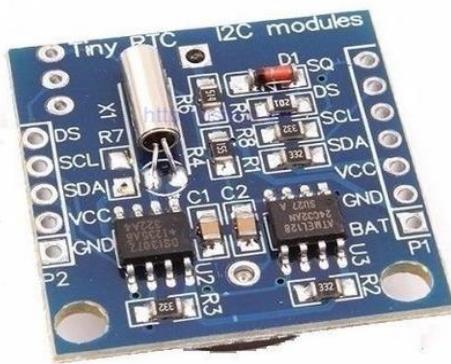
Thông số kỹ thuật [5]:

- Giao tiếp UART (TTL).
- Kích thước bề mặt chụp vân tay : 14x18 (mm).
- Kích thước mẫu vân tay: 512B.
- Khả năng lưu trữ vân tay: 128.
- Tỉ lệ từ chối sai (FRR): <1.0%.
- Tỉ lệ nhận dạng (FAR): 0.00001% (mức độ bảo mật có thể điều chỉnh).
- Môi trường làm việc: -20°C - +50°C.
- Cảm biến quang học.
- Baudrate: 57600.

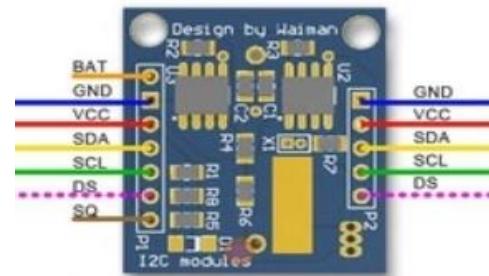
2.6.4 Module RTC DS1307

RTC - Đồng hồ thời gian thực được sử dụng để đếm và lưu các giá trị chỉ thời gian, có thể lập trình được thời gian. Sử dụng giao thức I2C để truyền dữ liệu. Module được thiết kế kèm sẵn pin để lưu trữ thông tin đã được đếm lên, lên đến 10 năm mà không cần phải cấp nguồn 5V vào từ bên ngoài.

Module được tích hợp sẵn 2 IC DS1307 giao tiếp thông qua địa chỉ 0x68 và được tích hợp thêm một bộ nhớ NVRAM cho việc lưu trữ dữ liệu. [6]



Hình 2.14: Module thời gian thực RTC DS1307



Hình 2.15: Sơ đồ chân module thời gian thực RTC DS1307

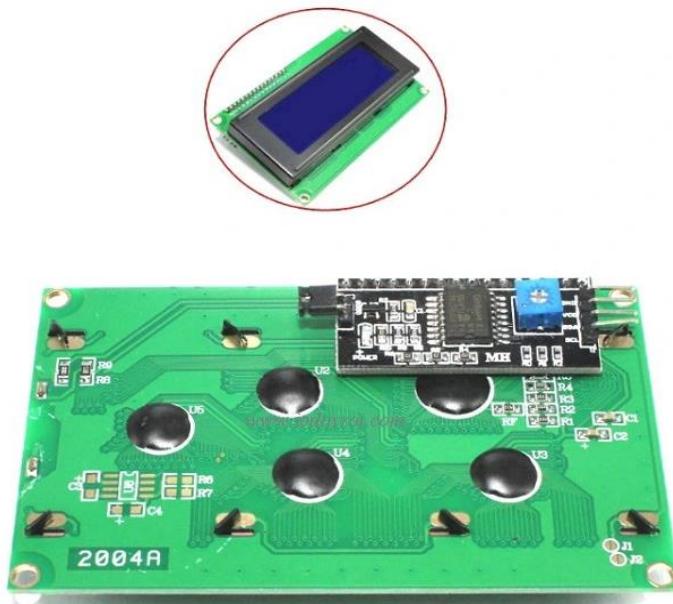
Thông số kỹ thuật:

- Điện áp làm việc: 5V.
- IC thời gian thực DS1307.
- Dùng chuẩn giao tiếp I2C (2 dây).

2.6.5 LCD 20x4 I2C

LCD 20x4 là một sản phẩm quá quen thuộc với những dự án điện tử nhỏ, LCD có rất nhiều dạng và được phân biệt với nhau bằng nhiều kích thước.

LCD được sử dụng trong đề tài là LCD 2004 gồm 4 hàng và 1 hàng có 20 ký tự và đi kèm với LCD là module I2C sử dụng driver HD44780.



Hình 2.16: Module LCD 20x04 tích hợp I2C

Thông số kỹ thuật [7]:

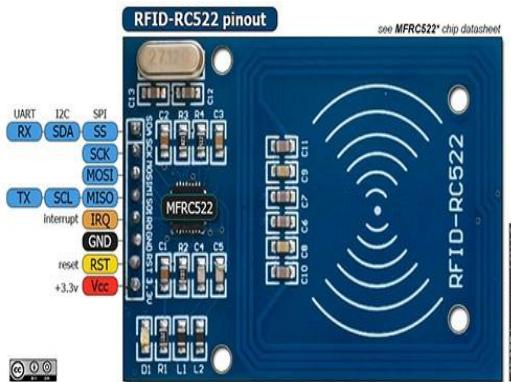
- Điện áp hoạt động: 5V.
- Kích thước: 98 x 60 x 13.5 mm.
- Giao tiếp I2C.

2.6.6 Module MFRC-522 NFC 13.56MHz

Module RFID RC522 NFC 13.56MHz dùng để đọc và ghi dữ liệu cho thẻ NFC tần số 13.56MHz. RFID – Radio Frequency Identification Detection là công nghệ nhận dạng đối tượng bằng sóng vô tuyến.



Hình 2.17: Module MFRC-522 NFC
13.56MHz



Hình 2.18 Sơ đồ chân của module
MFRC-522 NFC 13.56MHz

Thông số kỹ thuật [8]:

- Điện áp hoạt động: 3.3VDC, 13 – 26mA.
- Dòng ở chế độ chờ: 10-13mA.
- Dòng ở chế độ nghỉ: <80uA.
- Tần số sóng mang: 13.56MHz.
- Khoảng cách hoạt động: 0~60mm (mifar card).
- Giao tiếp: SPI.
- Tốc độ truyền dữ liệu: tối đa 10Mbit/s.
- Nhiệt độ hoạt động: -20 đến 80°C.
- Hỗ trợ: ISO / IEC 14443A /MIFAR.

2.6.7 Còi buzzer thụ động 5V

Buzzer được sử dụng có tác dụng báo hiệu cho các ứng dụng khác nhau. Được thiết kế nhỏ gọn, dễ sử dụng và thường được tích hợp trong các thiết bị điện tử hoặc hệ thống điều khiển.



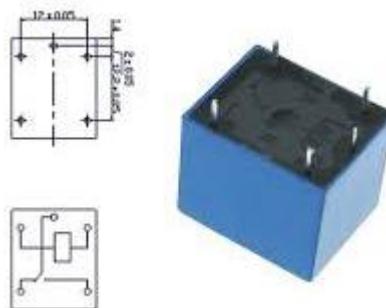
Hình 2.19: Buzzer thụ động 5V

Thông số kỹ thuật [9]:

- Điện áp hoạt động: 5VDC.
- Tần số hoạt động: 2KHz - 5KHz.
- Dòng điện tiêu thụ: <25mA.
- Biên độ âm thanh: >80 dB.
- Nhiệt độ hoạt động:-20 °C đến +70 °C.

2.6.8 Relay SRD-12VDC-SL-C

Relay hay còn được gọi là rơ – le là tên gọi theo tiếng Pháp. Nó là một công tắc (khóa K) điện từ và được vận hành bởi một dòng điện tương đối nhỏ có thể bật hoặc tắt một dòng điện lớn hơn rất nhiều. Được ứng dụng để điều khiển các thiết bị như động cơ, motor, điện



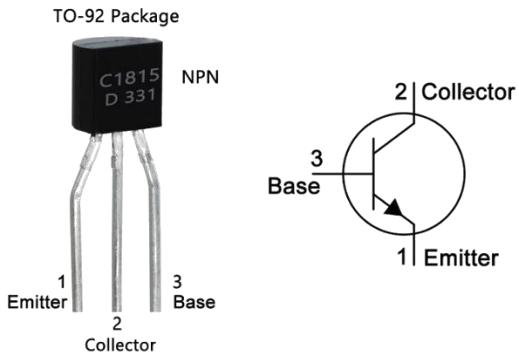
Hình 2.20: Relay SRD-12VDC-SL-C

Thông số kỹ thuật [10]:

- 10A - 250VAC: Cường độ dòng điện tối đa qua các tiếp điểm của rơ-le với hiệu điện thế $\leq 250V$ (AC) là 10A. Tương tự với các thông số trên thân.
- Dòng điện kích relay: 30mA.
- Thời gian tác động: 10ms.
- Thời gian nhả hẫm: 5ms.
- Nhiệt độ hoạt động: -45°C - 75°C.

2.6.9 Transistor NPN C1815

Transistor BJT là linh kiện được dùng để khuếch đại dòng điện. Do dòng ra ở trên chân vi điều khiển là khá nhỏ nên ta cần phải khuếch đại dòng điện để kích relay và các thiết bị khác.



Hình 2.21: Transistor NPN C1815

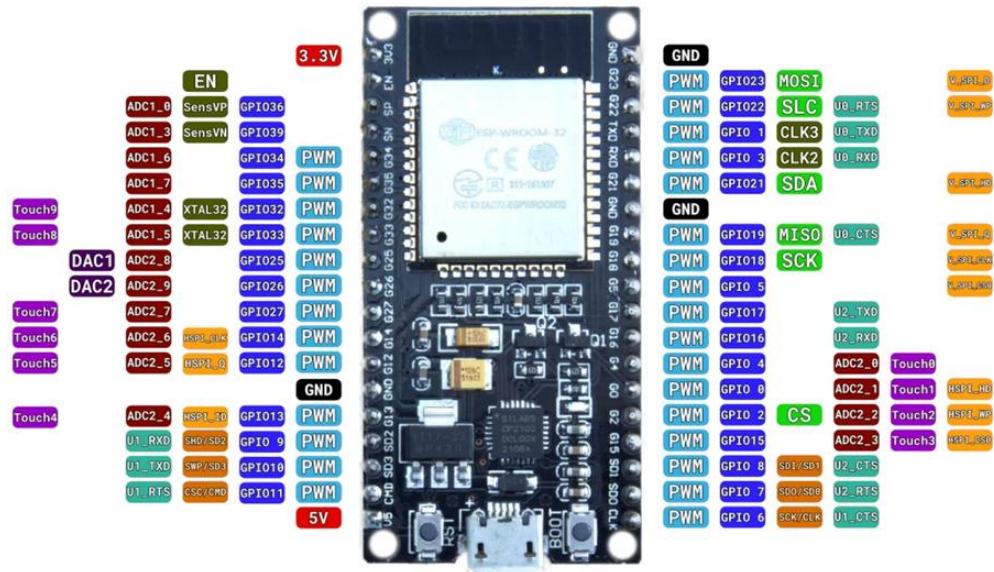
Thông số kỹ thuật [11]:

- Family: NPN.
- V_{CBOmax} : 60V
- V_{CEOmax} : 50V
- h_{fe} : 100-700.
- I_{Cmax} : 150mA.

2.6.10 Vi điều khiển ESP32 NodeMCU

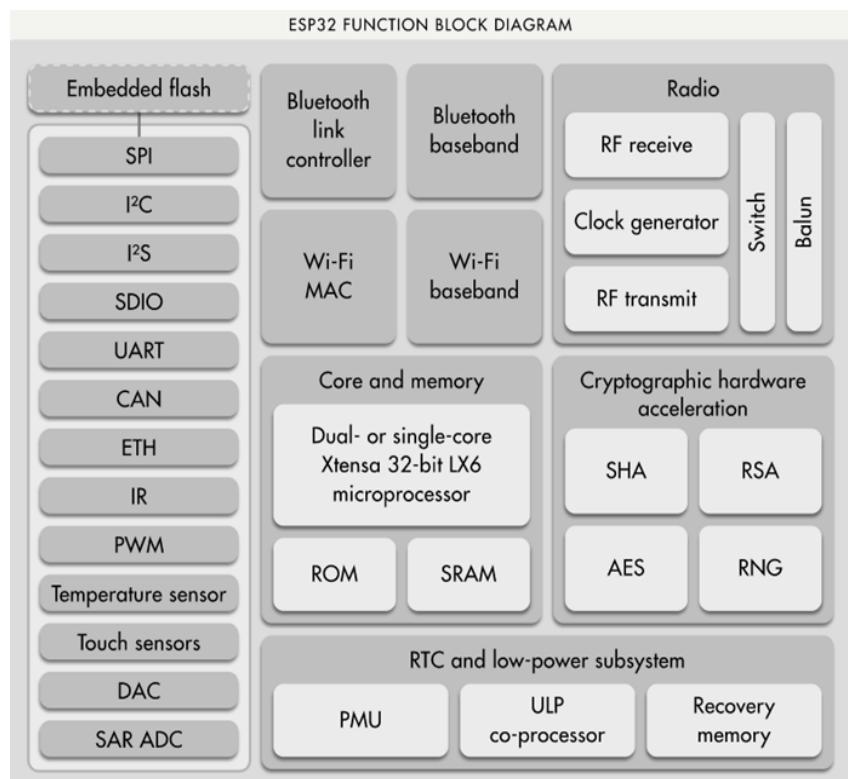
ESP32 là một series các vi điều khiển trên một vi mạch giá rẻ, năng lượng thấp có hỗ trợ WiFi và dual-mode Bluetooth (tạm dịch: Bluetooth chế độ kép). Dòng ESP32 sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng. ESP32 được chế tạo và phát triển bởi Espressif Systems, một công ty Trung Quốc có trụ sở tại Thượng Hải, và được sản xuất bởi TSMC bằng cách sử dụng công nghệ 40 nm. Nó được thiết kế để đạt được công suất và hiệu suất RF tốt nhất, thể hiện sự mạnh mẽ, tính linh hoạt và độ tin cậy trong nhiều ứng dụng và kịch bản năng lượng.

ESP32 là sản phẩm kế thừa từ vi điều khiển ESP8266. Dòng chip ESP32 bao gồm: ESP32-D0WDQ6, ESP32-D0WD, ESP32-D2WD, và ESP32-S0WD. [12]



Hình 2.22: Sơ đồ chân của module ESP32 NodeMCU

Kiến trúc vi điều khiển ESP32:



Hình 2.23: Block Diagram của ESP32 NodeMCU

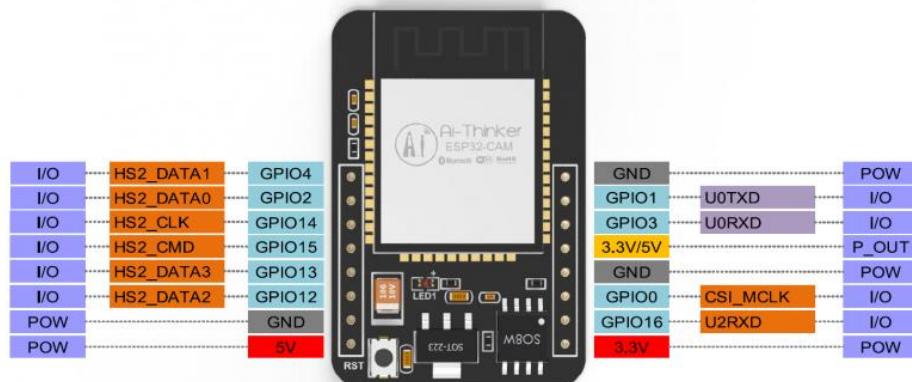
Thông số kỹ thuật:

Bảng 2.2: Thông số kỹ thuật của ESP32 NodeMCU

Điện áp hoạt động	5V (qua MicroUSB hoặc Vin)
Số chân	38
Bộ nhớ	RAM: 520KBytes ROM: 448KBytes
Dòng điện các chân IO	20mA (max)
Kiến trúc vi điều khiển	32 bits
GPIO giao tiếp mức	3.3V
Các ngoại vi	I2C, SPI, UART, PWM, DAC (8-bits), ADC(12-bits)...
Chuẩn WiFi	802.11b/g/n/e/i

2.6.11 Vi điều khiển ESP32-CAM

ESP32-CAM là một module camera kích thước nhỏ được phát hành bởi Essence. ESP32-CAM được sử dụng rộng rãi trong các ứng dụng IoT, thích hợp với các thiết bị dành cho smart home, và các ứng dụng IoT khác. Cung cấp các phương thức kết nối có độ tin cậy cao. [13]



Hình 2.24: Sơ đồ chân của module ESP32-CAM

Thông số kỹ thuật: [13]

Bảng 2.3: Bảng thông số kỹ thuật Module ESP32-CAM

Điện áp cung cấp	3.3V-5V
RAM	520KB SRAM và 4MB PSRAM
Chuẩn WiFi	802.11 b/g/n
Ngoại vi	UART, SPI, I2C, PWM
Nhiệt độ hoạt động	-20 °C ~ 85 °C
Chân IO giao tiếp mức	3.3V
Chuẩn WiFi	802.11b/g/n/e/i

CHƯƠNG 3:

THIẾT KẾ HỆ THỐNG

3.1 GIỚI THIỆU VÀ TÓM TẮT

3.1.1 Yêu cầu chung của hệ thống

Với tiêu chí thiết kế hệ thống khóa cửa an ninh cung cấp sự tiện lợi, an toàn cũng như kịp thời phát hiện những mối đe dọa sớm với tài sản của người sử dụng. Hệ thống có các chức năng sau:

- Cửa có thể được mở bằng nhiều sự lựa chọn: mật khẩu, vân tay, RFID.
- Có LCD giúp người dùng dễ dàng tương tác và thực hiện các tính năng hệ thống.
- Có khả năng thêm xóa vân tay, RFID và reset mật khẩu về giá trị mặc định trong những trường hợp khách quan và chủ quan.
- Tính năng cảnh báo thông qua email.
- Website cung cấp tính năng ghi lại lịch sử ra vào cửa và có khả năng giám sát camera.

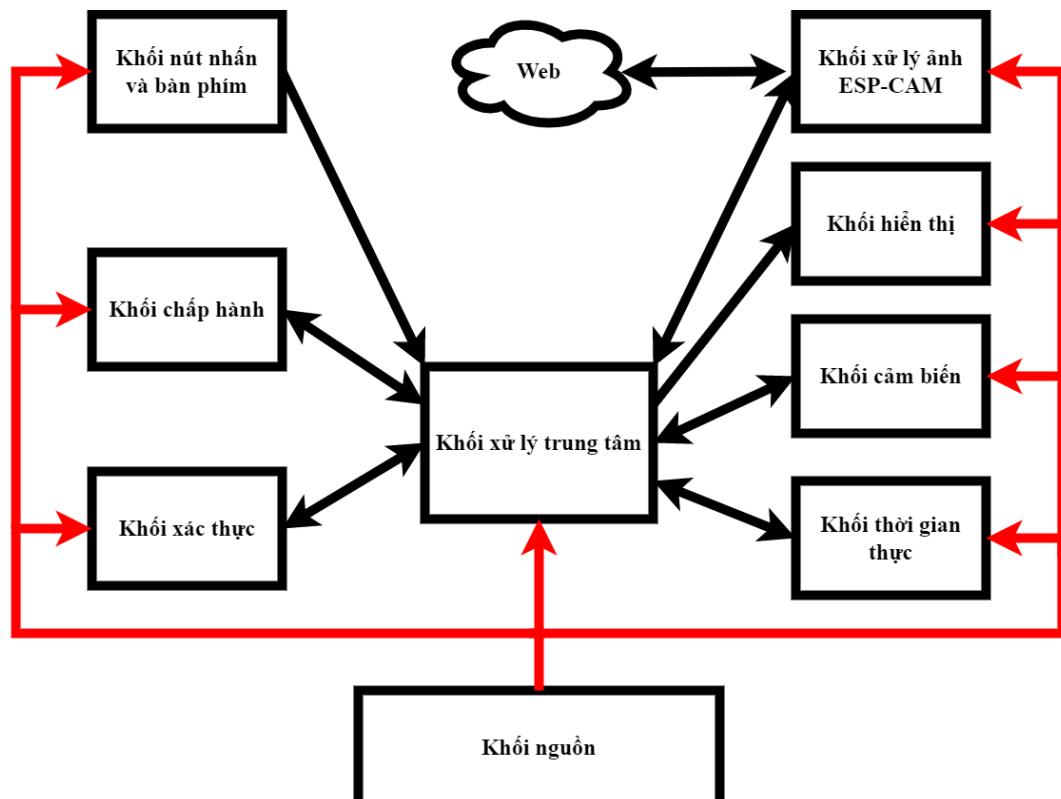
3.1.2 Phương án thiết kế

Về phần cứng, với các yêu cầu trên, nhóm đã đề ra phương án thiết kế cho phần cứng như sau:

- Khối xử lý: Nhóm sử dụng ESP32 NodeMCU.
- Khối camera: Nhóm sử dụng module ESP32-CAM.
- Khối xác thực: Nhóm sử dụng module RFID MFRC522 NFC 13.56MHz và module AS608.
- Khối bàn phím: Nhóm sử dụng keypad 4x4.
- Khối hiển thị: Sử dụng LCD 20x4 có tích hợp module I2C.
- Khối cảm biến: Cảm biến từ MC-38.

- Khối thực thi: Khóa chốt điện DC12V LY-03 và RELAY 12V và Buzzer 5V.
 - Khối nguồn: Sử dụng kết hợp 2 nguồn cung cấp 12V (cung cấp cho chốt điện) và 5V.
 - Nhóm thực hiện thiết kế mạch in để kết nối các module thiết bị lại với nhau.
- Về phần mềm, nhóm đề ra phương án thiết kế như sau:
- Sử dụng ngôn ngữ lập trình C/C++ để xây dựng chương trình chính của hệ thống.
 - Sử dụng ngôn ngữ lập trình HTML,CSS để xây dựng, thiết kế web tĩnh.
 - Sử dụng ngôn ngữ lập trình PHP để xây dựng và giao tiếp với cơ sở dữ liệu và server.
 - Sử dụng Visual Studio Code để lập trình cho trang web và Arduino IDE cho việc lập trình cho toàn hệ thống.
 - MySQL và phpMyadmin được nhóm sử dụng để làm CSDL cho hệ thống.

3.2 SƠ ĐỒ KHỐI CỦA HỆ THỐNG



Hình 3.1: Sơ đồ khối của hệ thống.

Khối xử lý trung tâm: đảm nhiệm chức năng xử lý và điều khiển toàn mạch.

Khối xử lý ảnh: thu thập hình ảnh khi có người tương tác với sản phẩm và gửi dữ liệu về hình ảnh lên server, đồng thời cũng có chức năng tương tác với server.

Khối nút nhấn và bàn phím: điều hướng cho menu chức năng và nhập mật khẩu kiểu kí tự.

Khối xác thực: Đọc và gửi dữ liệu về RFID, vân tay về khối xử lý trung tâm.

Khối hiển thị: Hiển thị menu điều khiển, thời gian, trạng thái khóa và thông báo khi người dùng tương tác với sản phẩm.

Khối thời gian thực: hoạt động như 1 đồng hồ dùng để hiển thị và lưu lại giá trị thời gian.

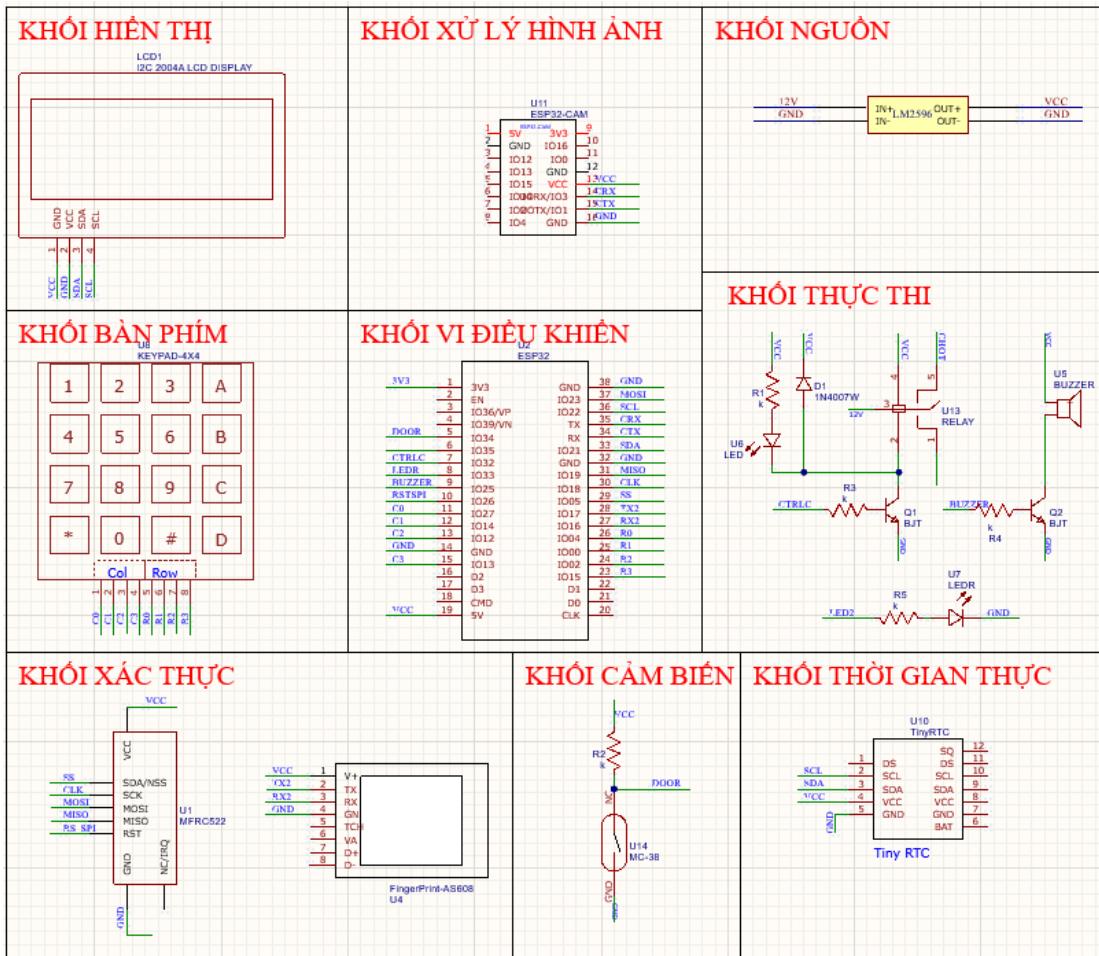
Khối cảm biến: nhận biết trạng thái khóa của cửa.

Khối thực thi: khôi này thực hiện đóng mở khóa và cảnh báo.

Khối nguồn: cung cấp năng lượng cho toàn mạch.

3.3 THIẾT KẾ CHI TIẾT KHÓA CỦA AN NINH

Sau khi xác định được các khôi chính trong sơ đồ khôi và phần cứng của hệ thống, nhóm thực hiện đã tiến hành thiết kế sơ đồ nguyên lý của hệ thống như sau:



Hình 3.2: Sơ đồ nguyên lý toàn hệ thống

Thiết kế chi tiết của từng khối như sau:

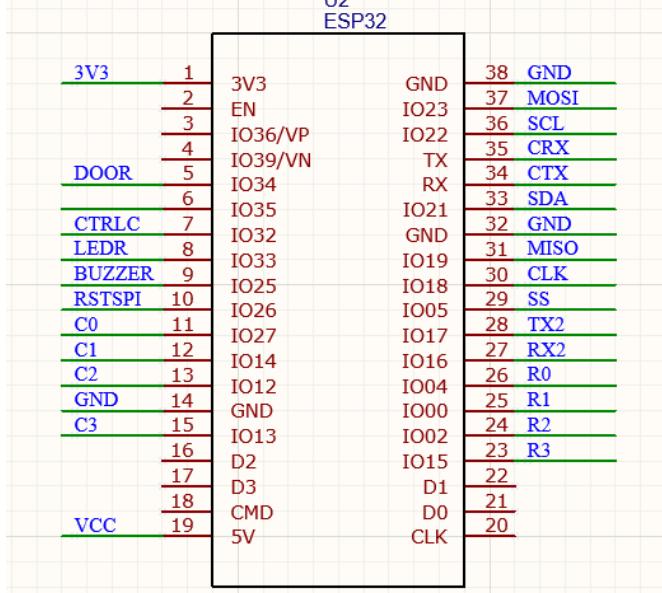
3.3.1 Khối xử lý trung tâm

Nhóm chọn vi điều khiển ESP32 được tính hợp trong module ESP32 NodeMCU LuaNode32 nhờ vào các ưu điểm như sau:

- Đây là 1 vi điều khiển phổ biến, dễ tìm hiểu, tiếp cận và cộng đồng hỗ trợ lớn.
- Được trang bị công nghệ WiFi và bluetooth, phù hợp cho việc thu thập và truyền tải dữ liệu qua internet nhờ kết nối WiFi.
- Các chân I/O đầy đủ, có khả năng giao tiếp I2C, SPI, UART, có khả năng giao tiếp với nhiều loại thiết bị.

Sau khi tính toán, nhóm đã tiến hành chọn lựa các GPIO để kết nối với các thiết bị ngoại vi như sau:

KHỐI VI ĐIỀU KHIỂN



Hình 3.3: Sơ đồ nguyên lý ESP32 và kết nối ngoại vi

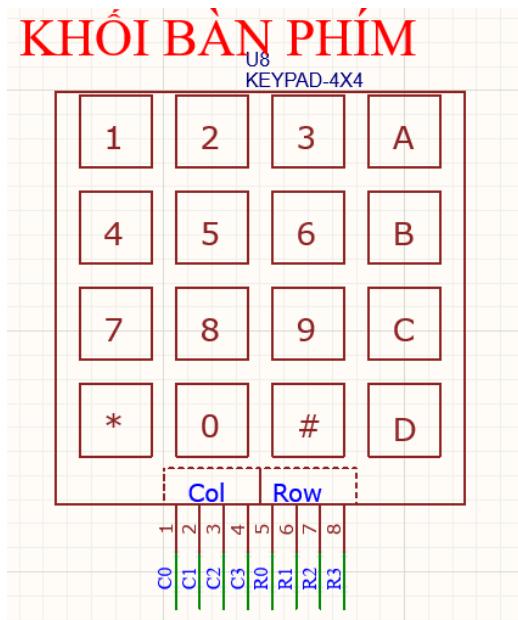
Bảng bên dưới sẽ mô tả rõ hơn về các kết nối mà nhóm thực hiện thiết kế.

Bảng 3.1: Bảng mô tả kết nối với khối xử lý trung tâm

GPIO	Thiết bị kết nối	Ghi chú
0, 2, 4, 12, 13, 14, 15, 27	Keypad	
5, 18, 19, 23, 26	MFRC522	Các chân này là chân được thiết lập chuẩn SPI của ESP32.
21, 22	LCD và Module DS1307	Các chân này được thiết lập chuẩn I2C của ESP 32.
1, 3	ESP32 - CAM	Các chân này được thiết lập chuẩn giao tiếp UART của ESP32.
16, 17	Cảm biến vân tay AS-608	Các chân này được thiết lập chuẩn giao tiếp UART thứ 2 của ESP32.
33	LED RED	
32	Chân điều khiển đóng mở chốt	
34	DOOR sensor	
25	Buzzer	

3.3.2 Khối bàn phím

Trong khói này, nhóm sử dụng module keypad 4x4 làm phương tiện để nhập mật khẩu và điều hướng chính cho toàn hệ thống.



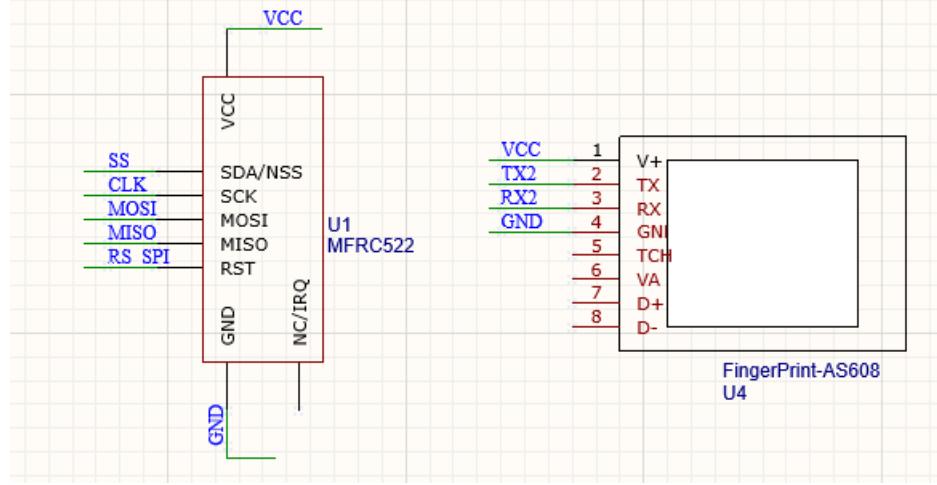
Hình 3.4: Sơ đồ nguyên lý của module keypad 4x4 và kết nối với khối xử lý

Về nguyên lý hoạt động: Nhóm thực hiện sử dụng phương pháp “quét” để xác định phím nào được sử dụng. Cụ thể như sau: Đầu tiên ta cho các cột của bàn phím được kéo lên mức cao, các hàng của bàn phím sẽ được cài đặt ở mức thấp. Khi có 1 phím nào đó từ 1 cột được nhấn thì các phím trong cột đó sẽ bị kéo xuống mức thấp. Lúc này ta tiến hành cho các hàng của bàn phím lên mức 1 và tiến hành gán mức 0 cho lần lượt các hàng. Nếu hàng nào làm cho cột chứa kí tự được nhấn đều bằng 0 thì ta có thể xác định được phím nào được nhấn.

3.3.3 Khối xác thực

Trong khói này nhóm thực hiện lựa chọn module RFID MFRC-522 NFC 13.56MHz và module cảm biến vân tay AS608. Lý do nhóm chọn 2 module này là vì nó được sử dụng rộng rãi nhờ tính tiện dụng, được sử dụng phổ biến cho các ứng dụng về bảo mật. Đặc tính dễ dàng giao tiếp cũng là lý do nhóm thực hiện chọn lựa 2 module này cho thiết kế.

KHỐI XÁC THỰC

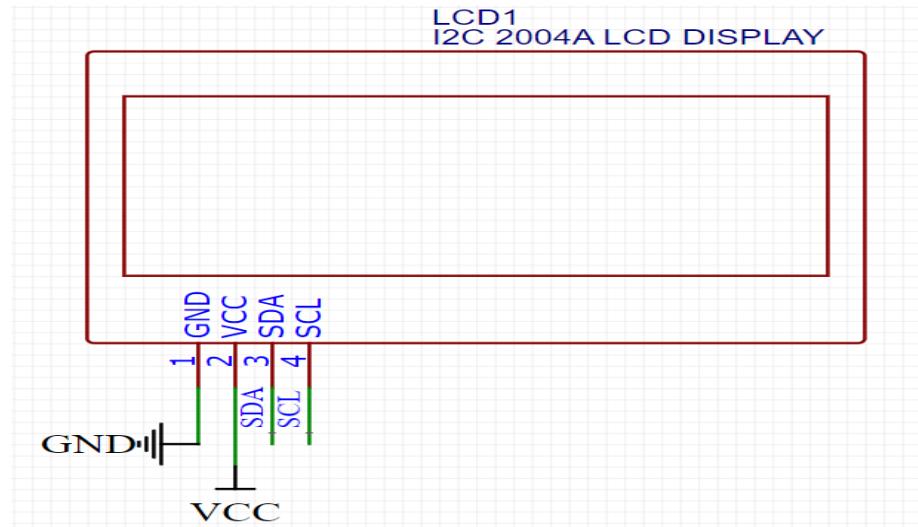


Hình 3.5: Sơ đồ nguyên lý của MFRC522, AS608 và kết nối khối xử lý
Về nguyên lý hoạt động:

- MFRC522 hoạt động như một đầu ghi dữ liệu, khi hoạt động, nó phát ra tần số 13,56MHz. Các thẻ RFID trong vùng phủ sóng của module khi cảm nhận được tín hiệu phát ra sẽ phản hồi về cho module biết về các thông số được lưu trong thẻ, thường là 1 mã số gồm có các kí tự chữ và số để thực hiện việc định danh cho thẻ, và các mã số này sẽ là đặc trưng riêng cho từng thẻ. Và module này sử dụng chuẩn SPI để giao tiếp với vi điều khiển.
- AS608: Khi đặt tay vào trong cảm biến, cảm biến sẽ tiến hành quét và xử lý vân tay, sau đó chuyển đổi dữ liệu thu được thành 1 chuỗi dữ liệu và truyền về vi điều khiển thông qua chuẩn UART để xử lý. Và module này sử dụng UART để giao tiếp với vi điều khiển.

3.3.4 Khối hiển thị

Khối này nhôm sử dụng LCD 20 x 4 kết hợp với module I2C nhằm mục đích hạn chế sử dụng nhiều GPIO và thuận tiện để giao tiếp với vi điều khiển thông qua chuẩn I2C được trang bị trên ESP32. Khối này được kết nối với vi điều khiển như sau:

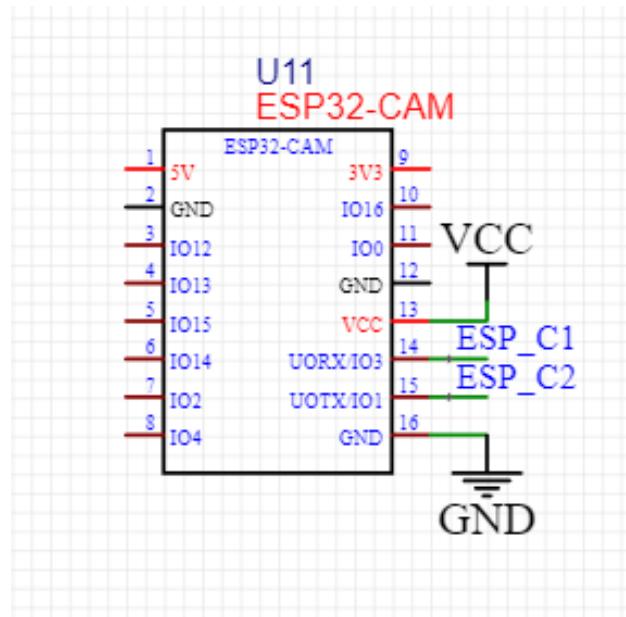


Hình 3.6: Sơ đồ nguyên lý của LCD và kết nối khối xử lý

Các chân SDA và SCL được kết nối với chân SDA và SCL của vi điều khiển để có thể giao tiếp I2C.

3.3.5 Khối xử lý ảnh ESP32-CAM

Khối này nhóm sử dụng module ESP32-CAM, đây là một phiên bản của ESP32. Tuy nhiên, phiên bản này được tùy biến nhằm mục đích chụp, ghi lại hình ảnh. Đây là 1 module phổ biến và cũng phù hợp với nhiều mô hình với nhiều khả năng vượt trội. ESP32-CAM sẽ đảm nhiệm chức năng kết nối Internet để giao tiếp với server.

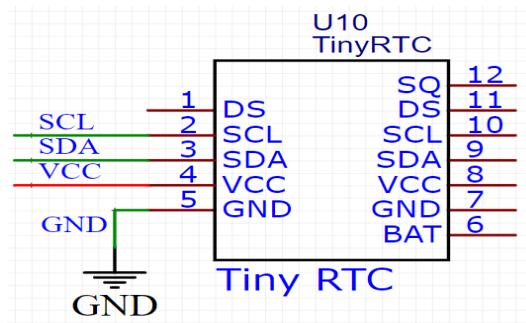


Hình 3.7: Sơ đồ nguyên lý ESP32-CAM và kết nối khối xử lý

ESP32-CAM có thể giao tiếp với vi điều khiển khác thông qua chuẩn UART, vì vậy nhóm thực hiện chọn UART thứ nhất được trang bị trên ESP32 để giao tiếp.

3.3.6 Khối thời gian thực

Trong khối này, nhóm sử dụng TinyRTC DS1307. Đây là một module phổ biến và được sử dụng rộng rãi trong các ứng dụng về thời gian thực.

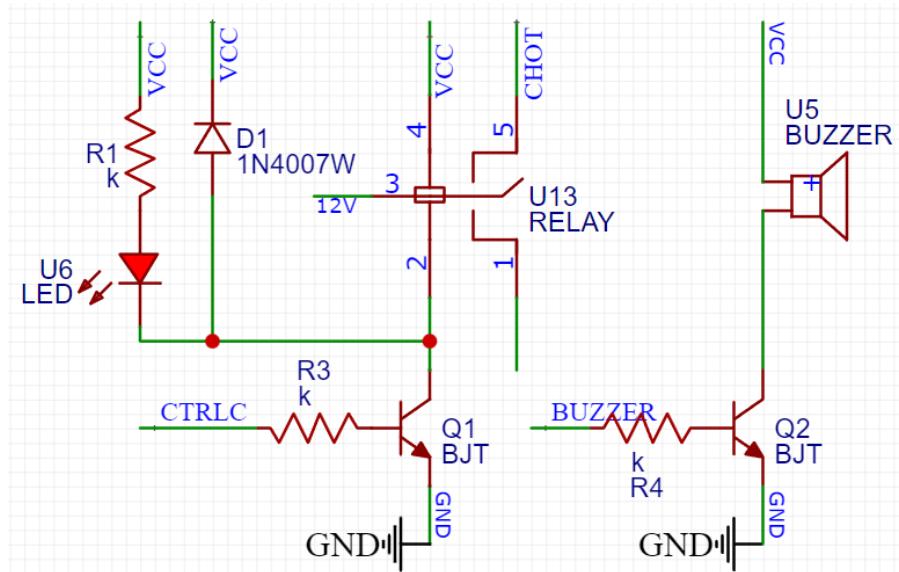


Hình 3.8: Sơ đồ nguyên lý của module RTC DS1307 và kết nối khối xử lý

Module này có thể giao tiếp được bằng chuẩn I2C. Và chuẩn I2C giao tiếp với các thiết bị thông qua địa chỉ nên nhóm thực hiện đã kết nối các module có cùng chuẩn giao tiếp này lại với nhau.

3.3.7 Khối thực thi

Khối này có chức năng thực hiện điều khiển các thiết bị dựa trên tín hiệu vi điều khiển gửi đến.



Hình 3.9: Sơ đồ nguyên lý khối thực thi và kết nối khối xử lý

* Tính toán thiết kế:

- Tính toán điện trở phân cực kích ro-le:
- Mạch làm việc ở chế độ bão hòa nên ta có: $V_{BE} = 1V$, $V_{BE} = 0.25V$ và $h_{fe} = 200$.
- Áp dụng Krichoff 1, ta có: $12V - V_{CEsat} - I_C \cdot R_{RL} = 0$
 $\Rightarrow I_C = \frac{12 - V_{CEsat}}{R_{RL}} = \frac{12 - 0.25}{400} = 29.375(mA)$
- Mà $I_C = \beta I_B \Rightarrow I_B = \frac{I_C}{\beta} = \frac{29.375}{200} = 0.146(mA)$ [14]
- Vậy để mạch phân cực ở chế độ bão hòa ta phải tìm R thỏa mãn $R_B \leq \frac{(V_{OH} - V_{BEsat})}{I_B} \Rightarrow R_B \leq \frac{(3.3 - 1)}{0.146} \approx 15.75(k\Omega)$ [14]
 \Rightarrow Chọn điện trở gần với giá trị $15.75k\Omega$. Chọn $R_B = 10k\Omega$.

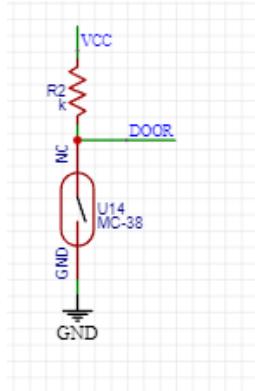
- Tính toán phân cực cho buzzer 5V:

- Vì dòng qua buzzer phải nhỏ hơn $30mA$ nên ta chọn $I_{Buzzer} = 26mA$
- Khi mạch làm việc ở chế độ khuếch đại ta có: $h_{fe} = 100$
- Áp dụng Krichoff 1, ta có: $R_B = \frac{(3.3 - 0.7) \cdot 100}{26} = 10k\Omega$
- Tính toán dòng cho LED hoạt động:
- Chọn dòng qua LED là $10mA$ và LED bật khi có tín hiệu mức cao ($3.3V$)
- Áp dụng Krichoff 1 ta có: $R = \frac{V_{OH} - V_{LED}}{I_{LED}} = \frac{3.3 - 2.2}{10mA} = 110\Omega$
 \Rightarrow Chọn điện trở $R = 100\Omega$.

* Diode mắc song song với relay: khi thay đổi điện áp đột ngột dòng điện trong cuộn dây của relay biến thiên nhanh đi qua cuộn dây gây ra hiện tượng dòng điện ngược với điện áp lớn không thoát được (thoát chậm) ở cực Collector của transistor, điện áp này có thể đánh thủng môi trường của transistor làm hỏng transistor hoặc gây giảm tuổi thọ của transistor. Diode ở đây có vai trò đưa dòng ngược đó thoát nguồn để triệt tiêu không gây ảnh hưởng cho transistor.

3.3.8 Khối cảm biến

Khối này gồm có cảm biến MC-38, nhò vào sự thay đổi trạng thái của cảm biến này, ta dễ dàng biết được cửa có được đóng, mở hay không.



Hình 3.10: Sơ đồ nguyên lý của khói cảm biến và kết nối khói xử lý

Nguyên lý hoạt động: Door sensor hoạt động như 1 công tắc. Khi 2 phần của Door sensor tiến lại gần nhau, công tắc sẽ được bật, tín hiệu từ vi điều khiển sẽ thông với đất. Lúc này, vi điều khiển sẽ dựa vào mức logic được nhận về từ cảm biến door mà xác định được trạng thái đóng mở của cửa.

3.3.9 Khối nguồn

- Khối nguồn được dùng để cung cấp điện áp cho toàn mạch hoạt động, ta cần tính toán để cho thiết bị xài nguồn được hoạt động ổn định trong thời gian dài và không bị hỏng hóc.

- Vì có nguồn điện để chốt điện hoạt động là 12V và điện áp hoạt động của các thiết bị khác là 3.3V/5V nên nhóm phải sử dụng kết hợp nguồn điện áp: điện áp cung cấp (chưa chuyển đổi) 12V, điện áp mạch hoạt động (cung cấp từ mạch giảm áp LM2596) 5V.



Hình 3.11: Mạch giảm áp LM2596

* Thông số kỹ thuật của mạch chuyển đổi:

- Điện áp đầu vào: Từ 3V đến 30V.
- Điện áp đầu ra: Điều chỉnh được trong khoảng 1.5V đến 30V.
- Dòng đáp ứng tối đa là 3A.
- Hiệu suất : 92%.
- Công suất : 15W.

* Tính toán dòng điện toàn mạch (dựa trên datasheet)

Bảng 3.2: Bảng tính toán công suất mạch

STT	Tên linh kiện	Điện áp cung cấp	Dòng điện hoạt động (max)	Công suất
1	Buzzer	5V	26mA	0.13W
2	Thẻ đọc MFRC522	3.3V	13mA	0.04W
3	ESP32	5V	240mA	1.2W
4	ESP32-CAM	3V	180mA	0.54W
5	Chốt điện	12V	800mA	9.6W
6	Cảm biến từ MC-38	3.3V(mức cao)	100mA	0.33W
7	Relay 12V-SL-C	12V	30mA	0.36W
8	Cảm biến vân tay AS608	3.3V	120mA	0.36W
			Tổng	13.16W

- Dựa vào bảng trên, để hệ thống hoạt động ổn định với thời gian dài nhóm chọn nguồn có thông số như sau:
 - Điện áp cung cấp: 12V
 - Dòng điện cung cấp: 2A
 => Nhóm thực hiện chọn nguồn sử dụng Adapter 12V-2A để cung cấp điện áp cho toàn mạch hoạt động.

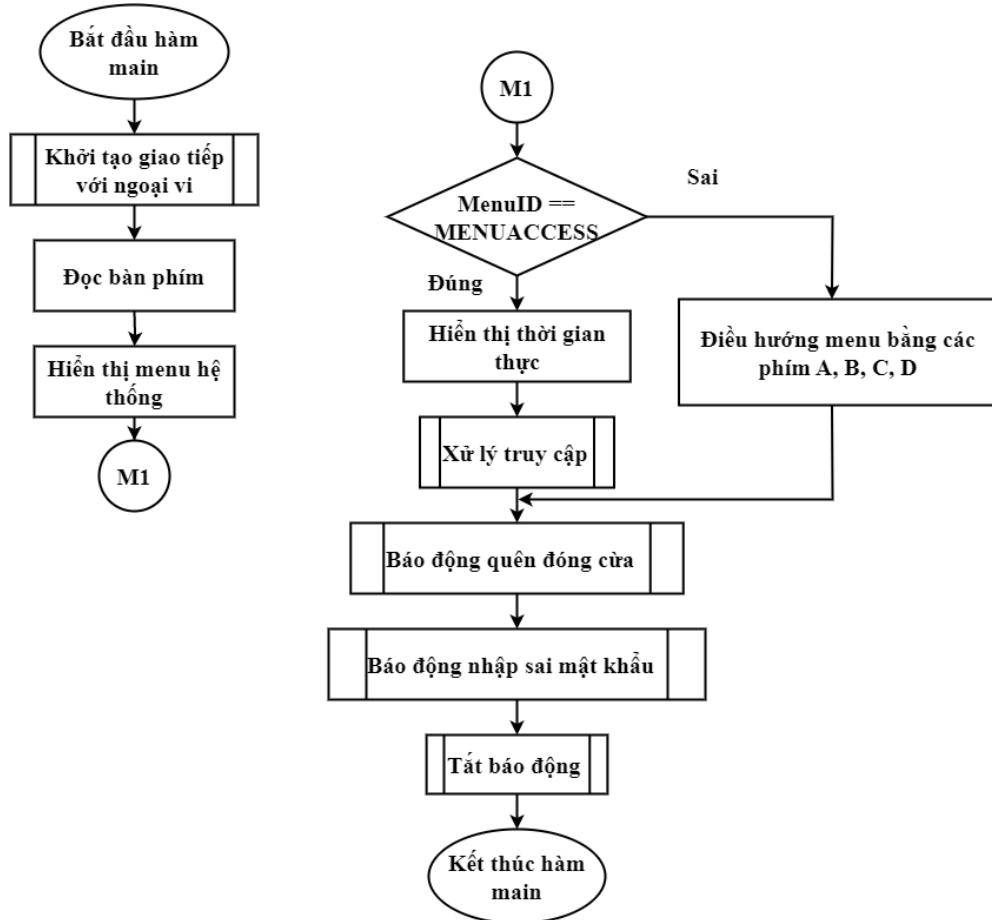


Hình 3.12: Adapter 12V-2A

3.4 LUU ĐO HOAT ĐONG CỦA HỆ THỐNG

Với việc sử dụng nhiều module khác nhau để sử dụng cho thiết kế, nên nhóm thực hiện bước đầu xác định cần chia nhỏ các phần thiết kế phần mềm thành các đoạn thư viện được thiết kế riêng cho việc giao tiếp với các module và sau cùng là gọi vào hàm main để thực hiện xử lý.

Lưu đồ của chương trình chính được nhóm thực hiện thiết kế như sau:



Hình 3.13: Lưu đồ giải thuật chương trình chính

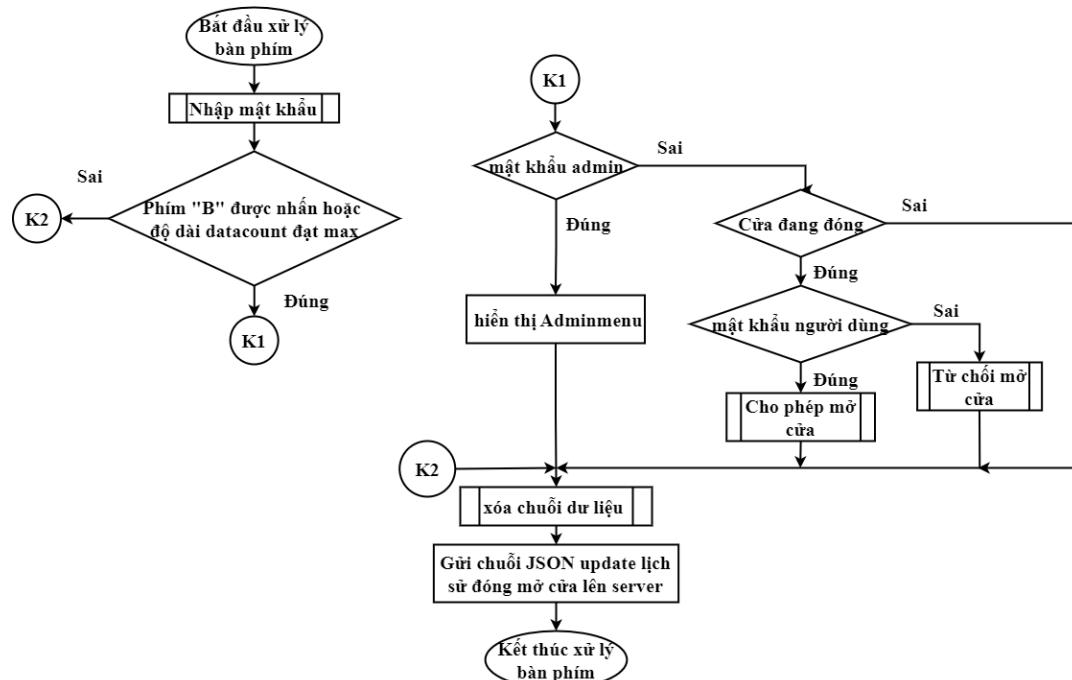
Khi bắt đầu hoạt động, chương trình chính sẽ gọi các hàm khởi tạo để bắt đầu giao tiếp với ngoại vi. Tiếp theo, menu chính của hệ thống sẽ được hiển thị lên màn hình LCD, và có thể được chuyển sang các menu hiển thị khác trên màn hình khi người dùng sử dụng phím A, B, C, D để điều hướng. Khi hệ thống đang ở menu chính, hàm xử lý truy cập sẽ được gọi và chờ người dùng nhập mật khẩu, RFID hoặc vân tay để tiến hành xử lý. Sau khi quá trình xử lý hoàn tất, hệ thống sẽ tiến hành thực hiện các hàm báo động nếu có sự sai sót nào đó đến từ phía người dùng.

3.4.1 Xử lý truy cập

Trong phần này, nhóm thực hiện chia thành 3 chương trình nhỏ gồm các hàm xử lý cho bàn phím 4x4, cho RFID và vân tay.

3.4.1.1 Xử lý cho bàn phím 4x4

Lưu đồ cho hàm này được nhóm thiết kế như sau:



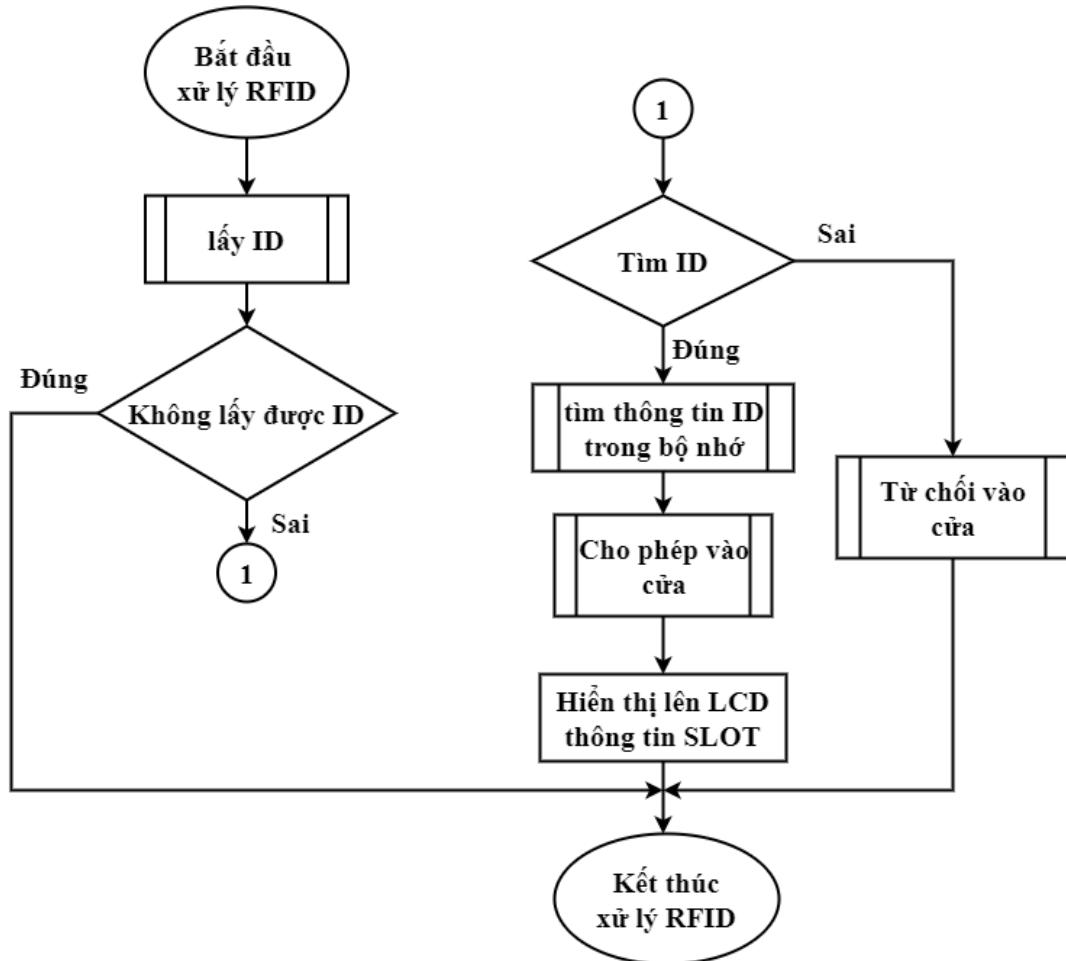
Hình 3.14: Lưu đồ giải thuật hàm xử lý xác nhận keypad

Khi bắt đầu chương trình, khi có người nhấn 1 phím bất kỳ từ bàn phím, độ dài chuỗi dữ liệu datacount sẽ tăng lên 1 và có thể tăng đến khi đạt giá trị tối đa là 19. Sau khi nhập xong chuỗi ký tự muốn nhập, có thể là mật khẩu admin, user hoặc 1 chuỗi số bất kỳ, người dùng cần nhấn phím B để tiến hành xác nhận. Lúc này,

chương trình sẽ tiến hành so sánh với các chuỗi dữ liệu đã có đang được lưu trong bộ nhớ. Nếu mật khẩu người dùng nhập là mật khẩu admin, hệ thống sẽ tiến hành đưa người dùng đến admin menu và tiến hành xử lý. Nếu người dùng nhập mật khẩu user, hệ thống sẽ tiến hành mở cửa. Nếu người dùng nhập sai mật khẩu thì cửa sẽ vẫn đóng và sau cùng là xóa chuỗi data đã được nhập.

3.4.1.2 Xử lý cho RFID.

Lưu đồ cho hàm này được nhóm thiết kế như sau:

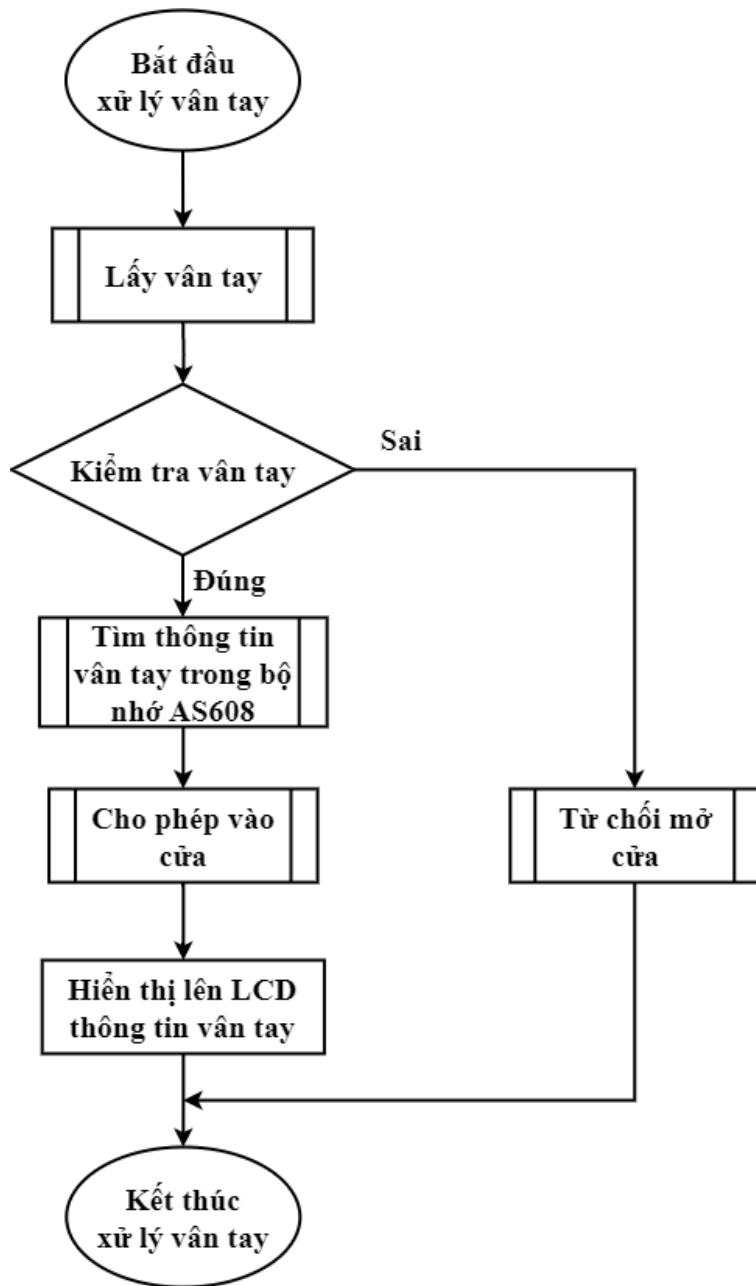


Hình 3.15: Lưu đồ giải thuật hàm xử lý xác nhận RFID

Bắt đầu chương trình, hàm lấy ID sẽ được gọi và nếu như có ID được lấy thì hệ thống sẽ bắt đầu tìm ID trong bộ nhớ, nếu ID có tồn tại, hệ thống sẽ tiến hành duyệt thông tin của ID, mở cửa và hiển thị thông tin ID lên LCD, nếu không có hệ thống sẽ từ chối mở cửa và kết thúc chương trình.

3.4.1.3 Hàm xử lý vân tay.

Lưu đồ cho hàm này được nhóm thiết kế như sau:



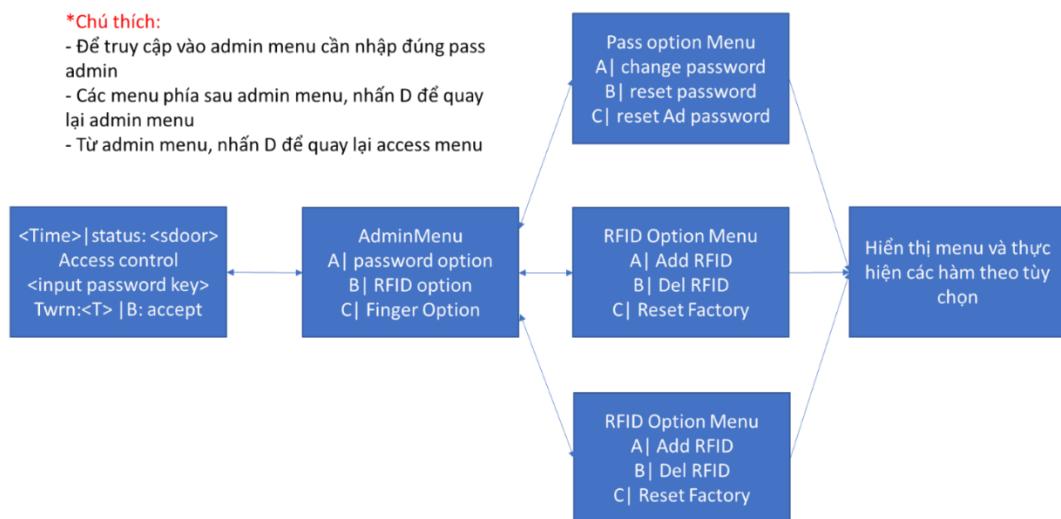
Hình 3.16: Lưu đồ giải thuật hàm xử lý xác nhận vân tay

Bắt đầu chương trình, hệ thống sẽ tiến hành quét vân tay, nếu có vân tay được quét và có vân tay đó tồn tại trong bộ nhớ, thông tin về vân tay sẽ được hiển thị lên LCD và cửa sẽ được mở, ngược lại cửa sẽ đóng và kết thúc chương trình.

3.4.2 Các menu điều hướng

Các menu này giúp người dùng có được cái nhìn trực quan về các tính năng cơ bản mà một khóa cửa an ninh có thể đem lại bằng cách nhập mật khẩu Admin vào hệ thống, menu sau đây sẽ hiển thị để người dùng thực hiện quản lý:

- Quản lý RFID.
- Quản lý vân tay.
- Quản lý mật khẩu.



Hình 3.17: Các menu điều hướng của hệ thống

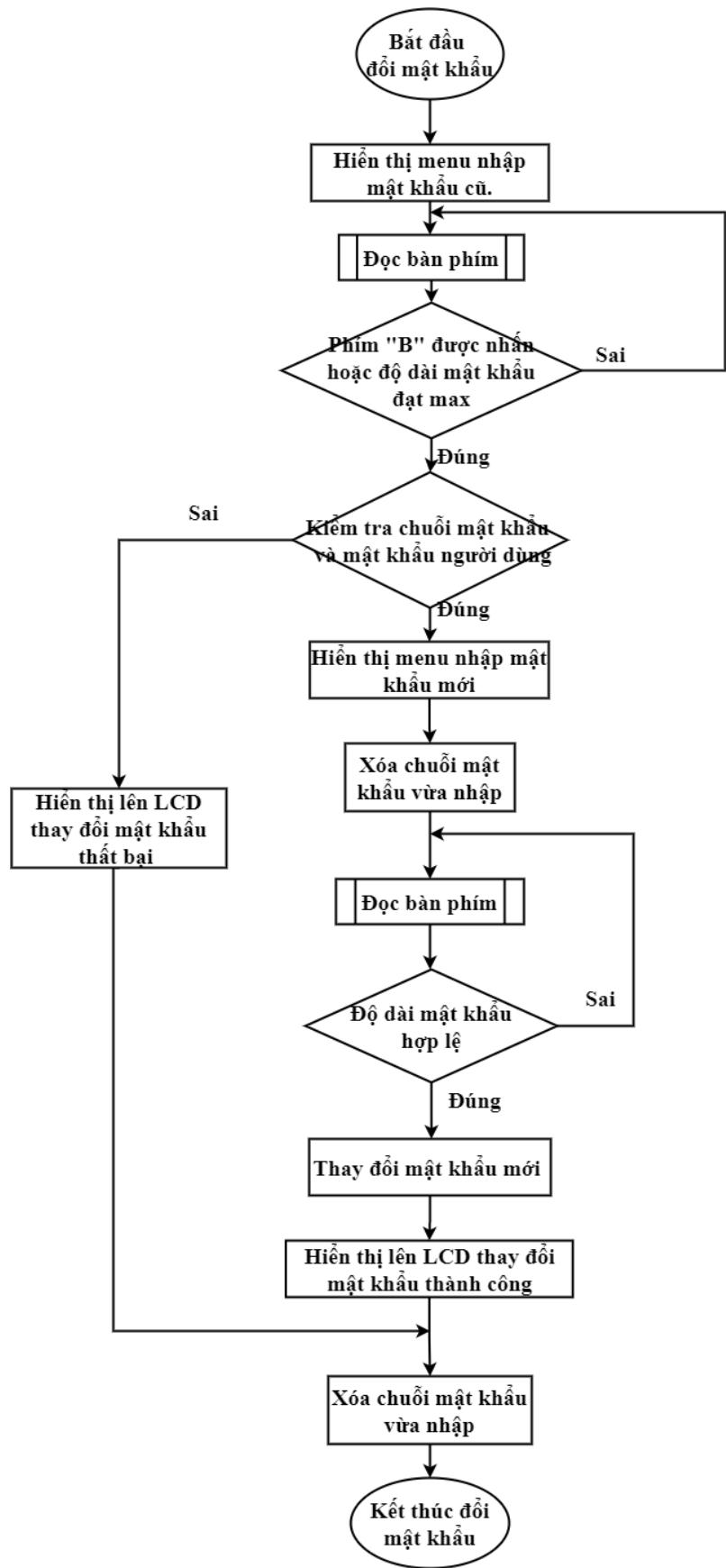
3.4.2.1 Menu quản lý mật khẩu.

Trong menu này, người dùng có thể thực hiện các chức năng cơ bản với mật khẩu như:

- Đổi mật khẩu user.
- Reset mật khẩu.
- Đổi mật khẩu admin.

a) Đổi mật khẩu user.

Giải thuật thiết kế cho hàm con thực hiện chức năng này được nhóm thiết kế như sau:



Hình 3.18: Lưu đồ giải thuật cho hàm đổi mật khẩu keypad người dùng

Khi bắt đầu, người dùng cần nhập lại mật khẩu cũ và nhấn phím B để xác nhận mật khẩu vừa nhập. Lúc này hệ thống sẽ tiến hành so sánh, nếu mật khẩu người dùng đúng, hệ thống sẽ yêu cầu người dùng nhập mật khẩu mới, ngược lại việc thay đổi mật khẩu sẽ không thành công.

b) Đổi mật khẩu Admin.

Phần này tương tự như đổi mật khẩu user, nhưng mật khẩu được đem ra so sánh là mật khẩu Admin.

c) Reset toàn bộ hệ thống.

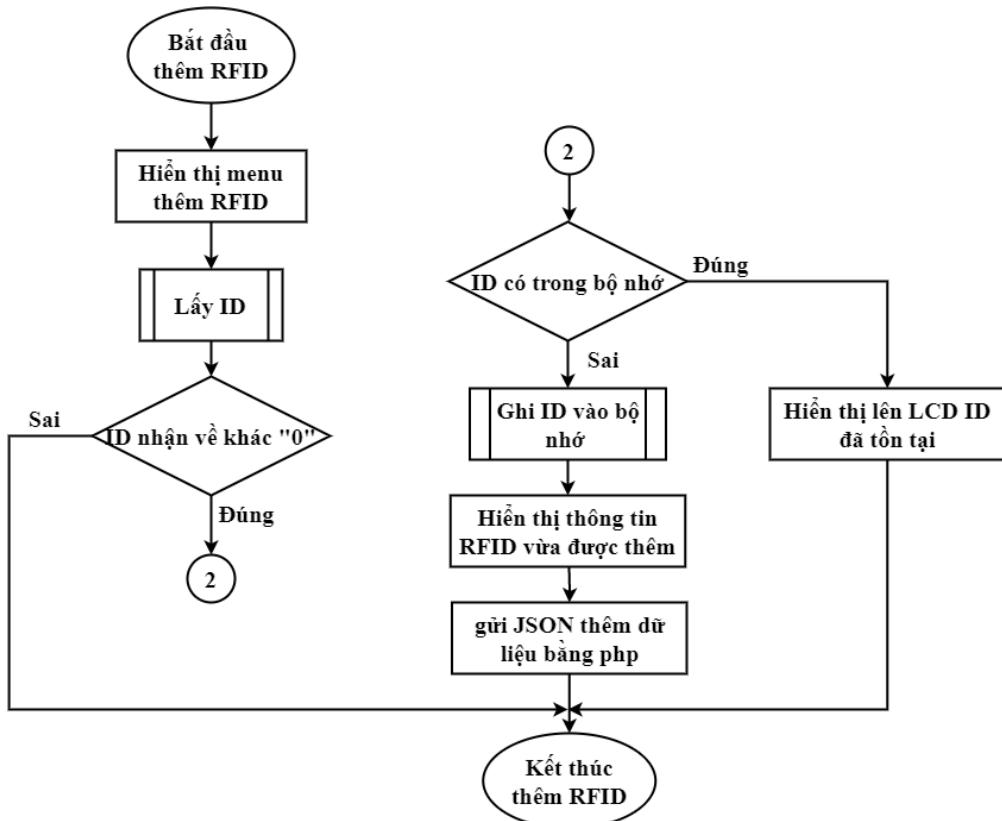
Dùng để xóa hết bộ nhớ trong EEPROM và đưa về mật khẩu admin và mật khẩu mở khóa mặc định khi chạy lại hệ thống lần tiếp theo.

3.4.2.2 Quản lý RFID

Menu này gồm có các lựa chọn:

- Thêm RFID.
- Xóa RFID.

a) Thêm RFID

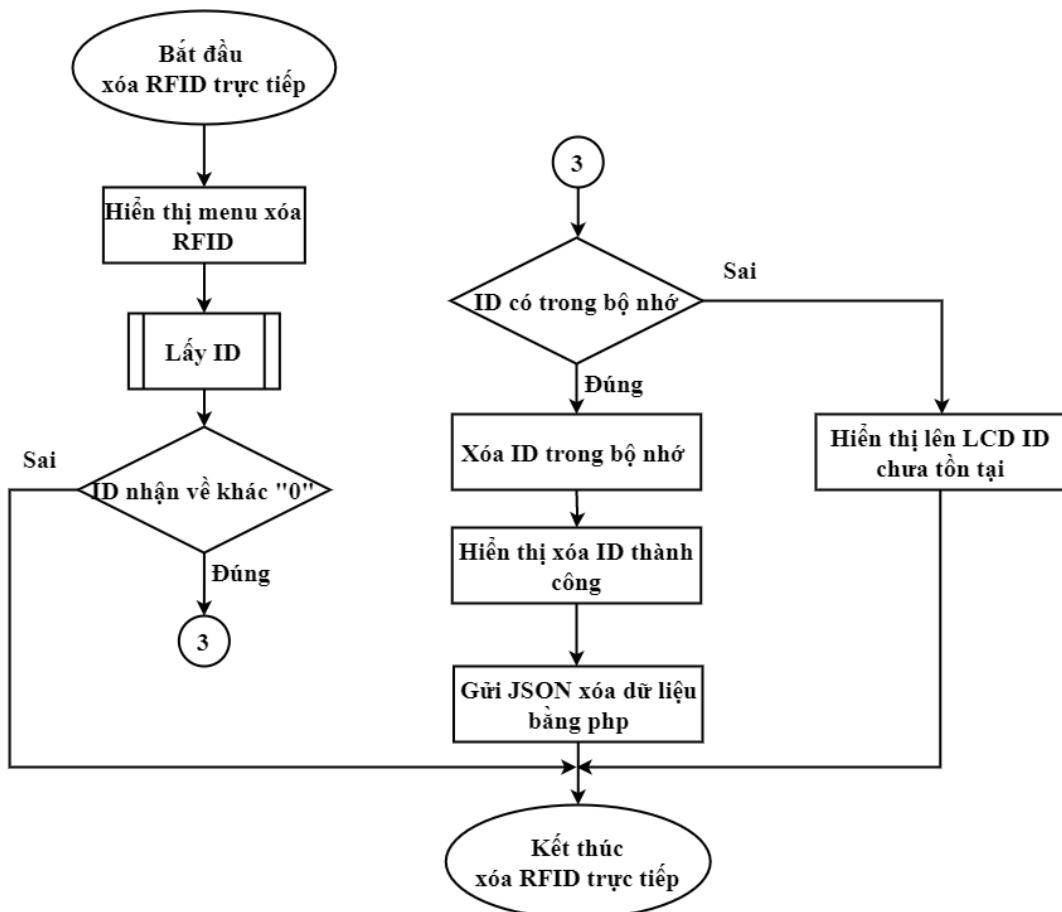


Hình 3.19: Lưu đồ giải thuật cho hàm thêm thẻ RFID

Khi bắt đầu, hệ thống sẽ quét ID, nếu quét được ID thì sẽ bắt đầu tìm ID đó trong bộ nhớ, nếu có ID, hệ thống sẽ hiển thị lên LCD là ID đã tồn tại và kết thúc chương trình, ngược lại sẽ ghi ID vào trong hệ thống và kết thúc chương trình.

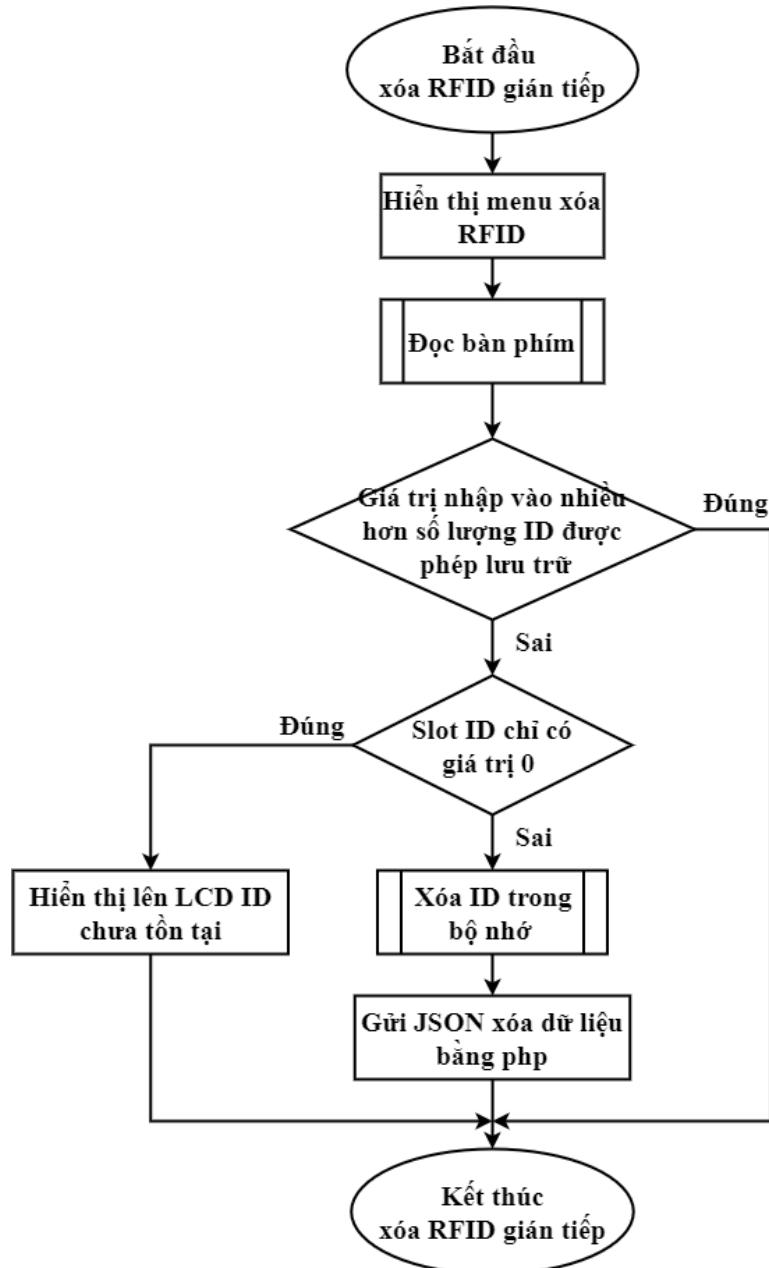
b) Xóa RFID

Trong phần này, nhóm lựa chọn 2 phương án để xóa RFID là xóa theo slot(số thứ tự của ID trong bộ nhớ) hoặc xóa theo ID.



Hình 3.20: Lưu đồ giải thuật cho hàm xóa RFID bằng mã định danh RFID

Khi bắt đầu, hệ thống sẽ quét ID, nếu quét được ID thì sẽ bắt đầu tìm ID đó trong bộ nhớ, nếu không có ID, hệ thống sẽ hiển thị lên LCD là ID chưa tồn tại và kết thúc chương trình, ngược lại sẽ xóa ID khỏi hệ thống và kết thúc chương trình.



Hình 3.21: Lưu đồ giải thuật cho hàm xóa RFID bằng vị trí

Khi bắt đầu, hệ thống yêu cầu nhập slot ID muốn xóa, nếu slot ID bé hơn số slot ID được phép lưu trữ thì sẽ bắt đầu tìm slot ID đó trong bộ nhớ, nếu không có thông tin về ID, hệ thống sẽ hiển thị lên LCD là ID chưa tồn tại và kết thúc chương trình, ngược lại sẽ xóa ID khỏi hệ thống và kết thúc chương trình.

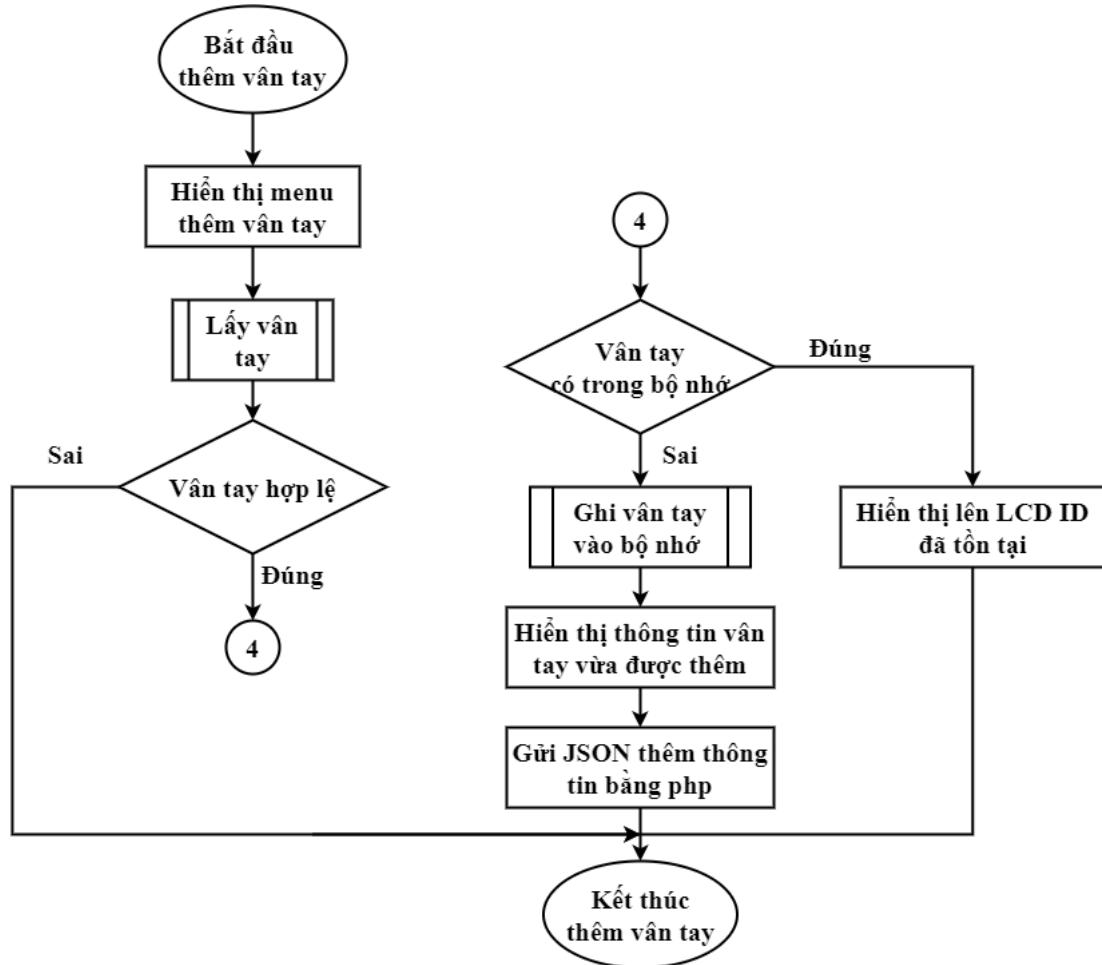
3.4.2.3 Quản lý vân tay

Menu này gồm có các lựa chọn:

- Thêm vân tay.

- Xóa vân tay.

a) Thêm vân tay.

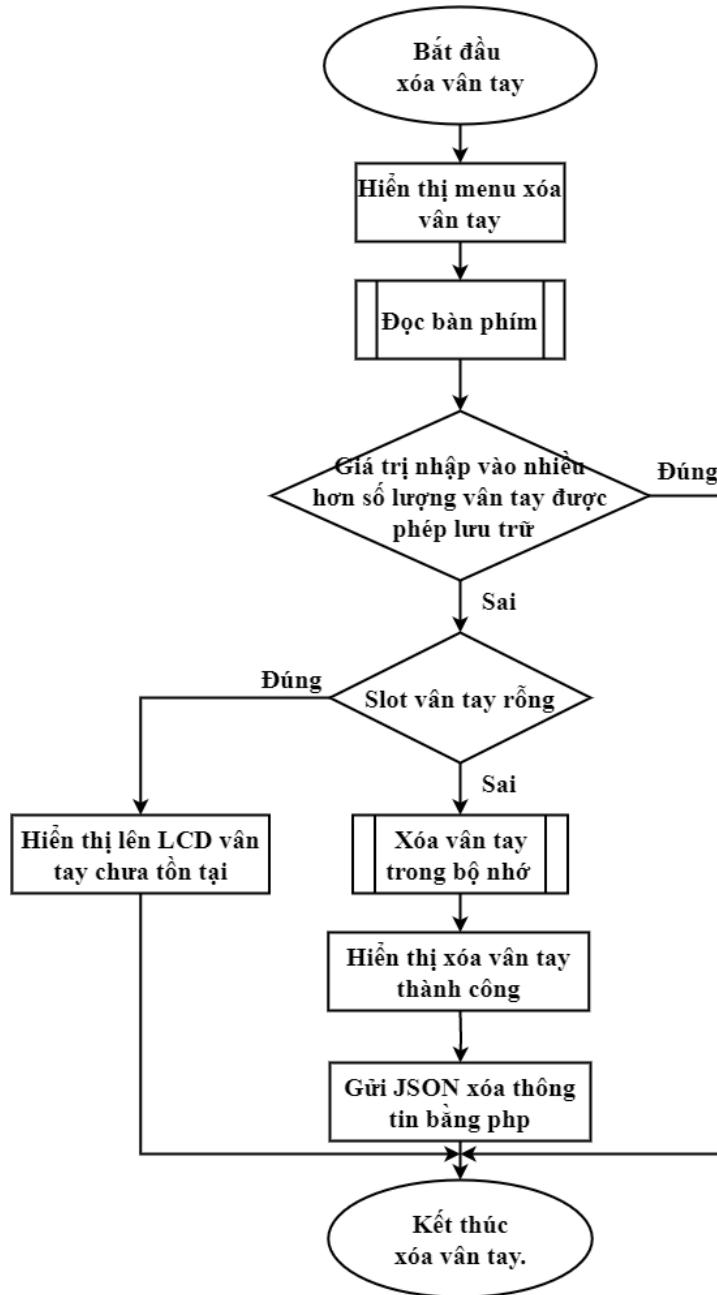


Hình 3.22: Lưu đồ giải thuật cho hàm thêm vân tay

Khi bắt đầu, hệ thống sẽ quét vân tay, nếu cảm biến vân tay trả về giá trị hợp lệ thì sẽ bắt đầu tìm vân tay đó trong bộ nhớ, nếu có vân tay, hệ thống sẽ hiển thị lên LCD là vân tay đã tồn tại và kết thúc chương trình, ngược lại sẽ ghi vân tay vào trong hệ thống và kết thúc chương trình.

b) Xóa vân tay.

Vân tay được nhóm quản lý theo mã ID được gắn cho vân tay.

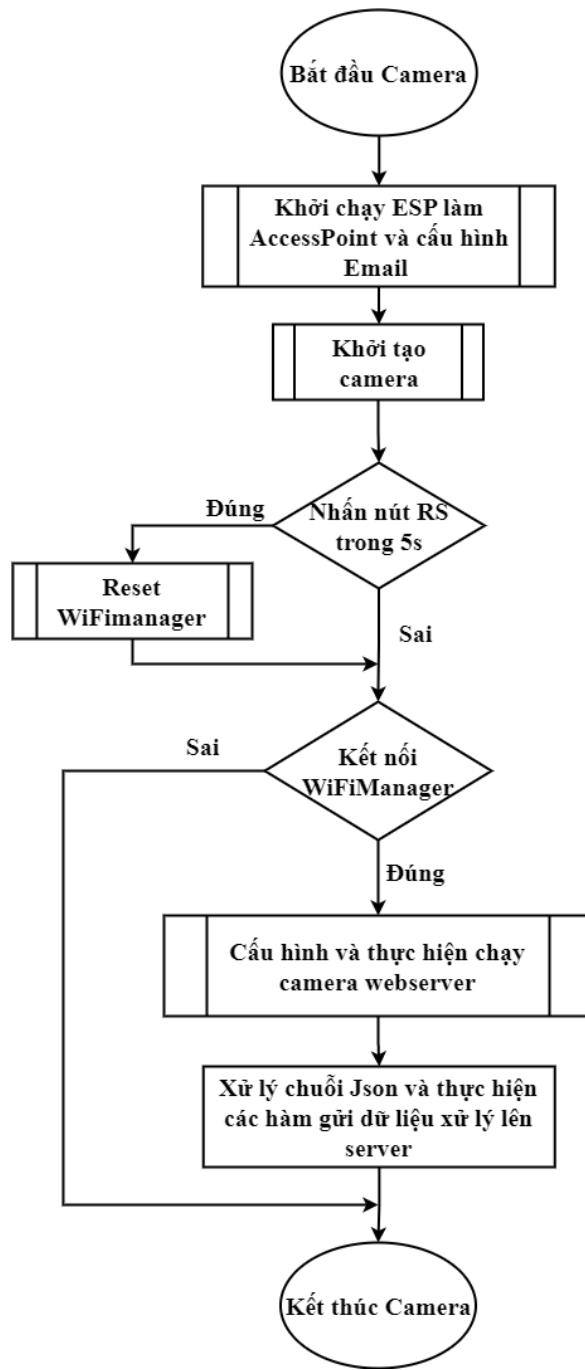


Hình 3.23: Lưu đồ giải thuật cho hàm xóa vân tay

Khi bắt đầu, hệ thống yêu cầu nhập slot vân tay muốn xóa, nếu slot vân tay bé hơn số slot vân tay được phép lưu trữ thì sẽ bắt đầu tìm slot vân tay đó trong bộ nhớ, nếu không có thông tin về vân tay, hệ thống sẽ hiển thị lên LCD là vân tay chưa tồn tại và kết thúc chương trình, ngược lại sẽ xóa vân tay khỏi hệ thống và kết thúc chương trình.

3.4.3 Camera

Việc giao tiếp với camera được nhóm thiết kế như sau:

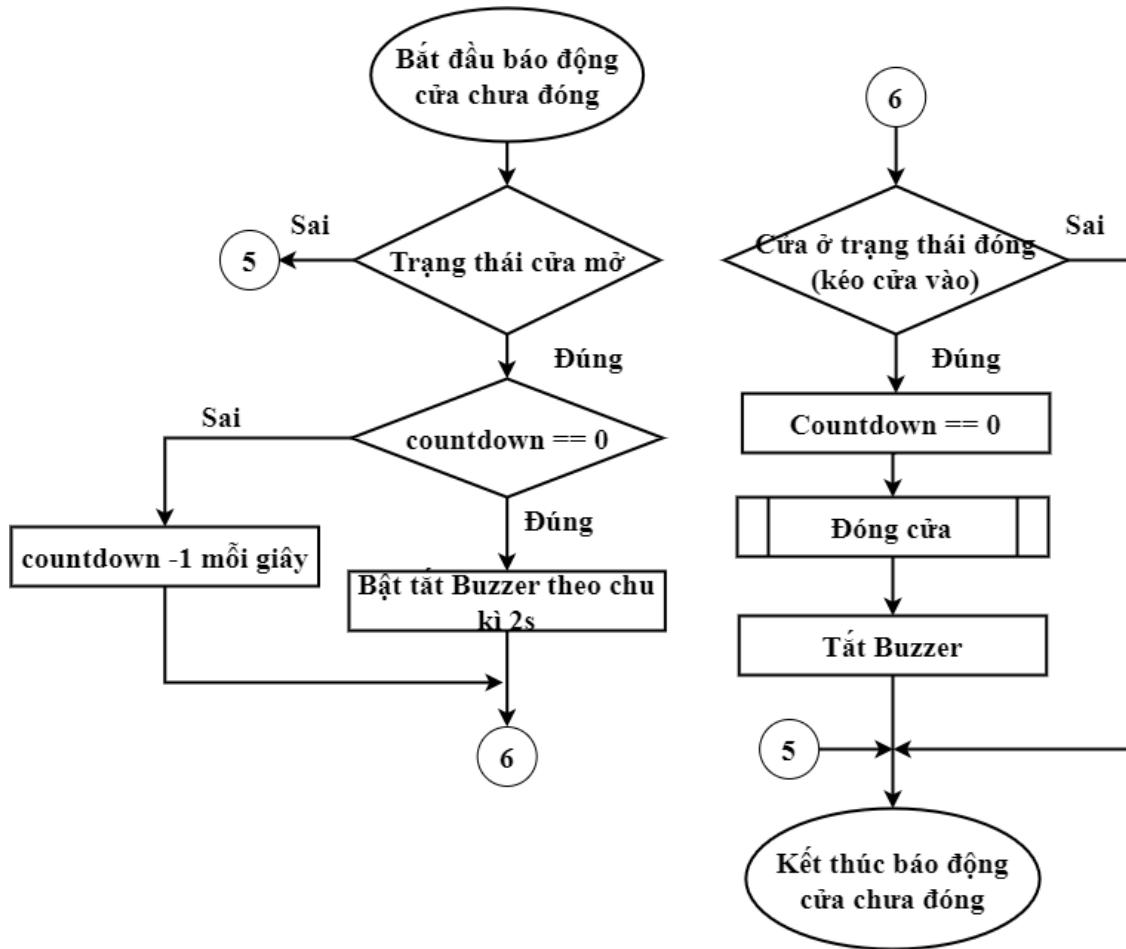


Hình 3.24: Lưu đồ giải thuật xử lý trên ESP32-CAM

Đầu tiên, ESP32 được khởi tạo chế độ Access point và bắt đầu thực hiện lệnh cấu hình cho gmail. Sau đó camera sẽ được khởi tạo, hệ thống sẽ bắt đầu kiểm tra xem nút nhấn Reset WiFi có được nhấn hay không, nếu có thì WiFiManager sẽ bắt đầu reset, nếu không, ESP32-CAM sẽ bắt đầu kết nối WiFi được cấu hình và thực hiện chạy và cấu hình streaming trên webserver. Cuối cùng, hệ thống sẽ bắt đầu xử lý chuỗi JSON và bắt đầu thực hiện gửi dữ liệu lên server.

3.4.4 Báo động quên đóng cửa

Lưu đồ báo động quên đóng cửa được nhóm thiết kế như sau:

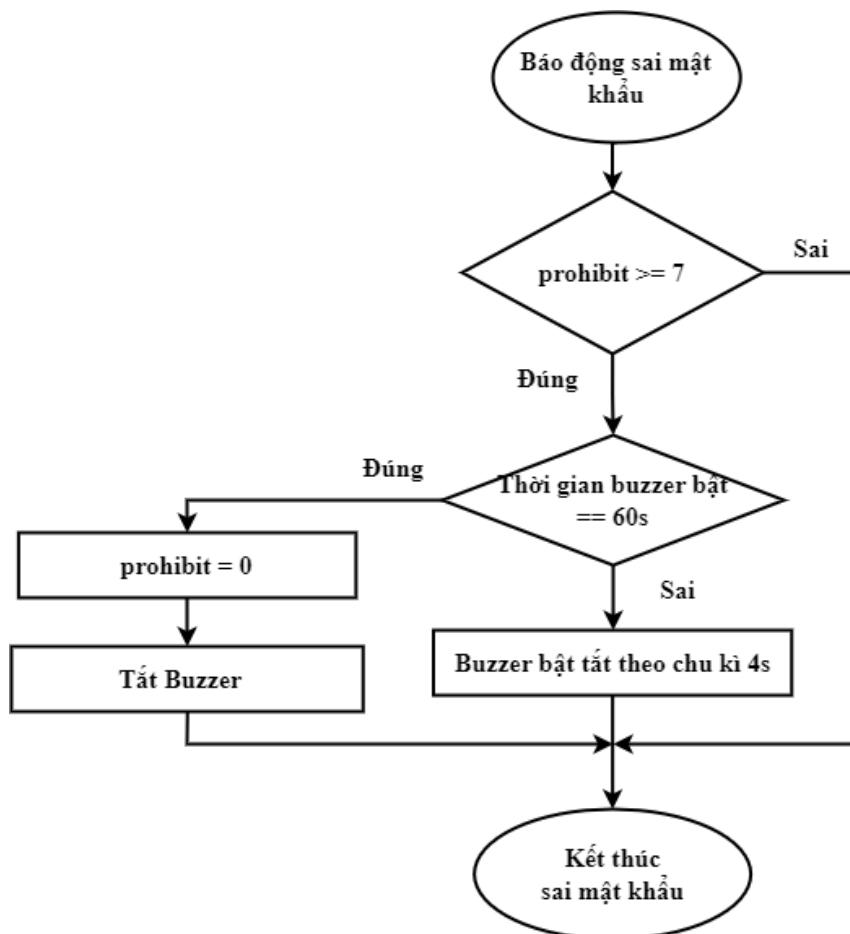


Hình 3.25: Lưu đồ báo động quên đóng cửa

Ban đầu hàm sẽ kiểm tra trạng thái cửa, nếu cửa chưa đóng và thời gian countdown đã đếm về 0 thì loa sẽ bắt đầu kêu theo chu kỳ 2s 1 lần và tiếp tục cho đến khi cửa đóng. Nếu thời gian countdown chưa bằng 0 thì countdown sẽ được trừ đi 1 mỗi 1s cho đến khi bằng 0. Nếu cửa đã đóng thì chương trình sẽ bỏ qua các xử lý bên trong và kết thúc chương trình.

3.4.5 Báo động sai mật khẩu.

Lưu đồ báo động sai mật khẩu được nhóm thiết kế như sau:



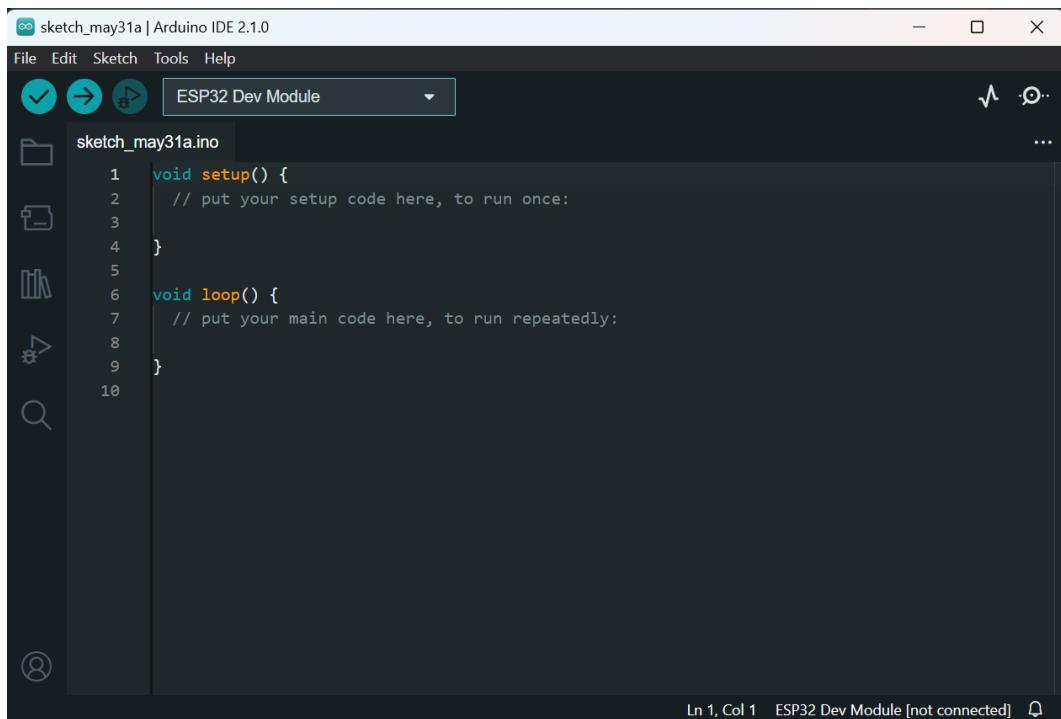
Hình 3.26: Lưu đồ báo động sai mật khẩu

Khi mật khẩu nhập sai quá 7 lần Loa sẽ bật báo động trong 60s với chu kì 4s mỗi lần. Nếu thời gian báo động vượt quá 60s, báo động sẽ tắt và reset lại trạng thái nhập sai mật khẩu (tức là prohibit =0).

3.5 LẬP TRÌNH CHO ESP32

3.5.1 Công cụ lập trình

Để lập trình cho ESP32, nhóm sử dụng phần mềm trình biên dịch ARDUINO IDE. Đây là một phần mềm lập trình cho các kit arduino với giao diện trực quan, dễ dùng và hoàn toàn miễn phí.



Hình 3.27: Giao diện cửa sổ lập trình trên Arduino IDE

Nhóm sử dụng phần mềm Arduino IDE phiên bản 2.1.0 để thực hiện lập trình cho hệ thống điều khiển.

Cần cài đặt board ESP32 để lập trình được trên Arduino:

- Ở giao diện của Arduino ta chọn File > Preferences.
- Tại Additional Board Manager URLs, ta nhập:

https://dl.espressif.com/dl/package_esp32_index.json.

- Chọn vào Tool > Board > Board Manager.
- Cửa sổ hiện lên và tìm kiếm với từ khóa “ESP32”, trong danh sách kết quả chọn “ESP32 by Espressif System” và nhấn Install.
- Chọn Board và thực hiện lập trình cho ESP32.

3.5.2 Một số xử lý cho ESP32 và ESP32-CAM

a. Ghi, đọc vào EEPROM.

Sử dụng thư viện EEPROM hỗ trợ trên Arduino để thực hiện giao tiếp, một số câu lệnh sử dụng như sau:

- EEPROM.read(addr): đọc nội dung ở địa chỉ addr.
- EEPROM.write(addr, value): ghi value vào địa chỉ addr.

- EEPROM.begin(size): khởi tạo EEPROM với kích thước size.
- EEPROM.commit(): xác nhận thực hiện thay đổi khi dùng EEPROM.write.

b. Chương trình thời gian thực:

- Sử dụng module DS1307 để lưu thời gian thực.
- Sử dụng thư viện <RTClib.h> để giao tiếp với module.
- Một số lệnh được sử dụng để giao tiếp: rtc.adjust và rtc.now() điều chỉnh và đọc thời gian hiện tại của rtc DS1307.

c. Giao tiếp, thực hiện hàm qua UART sử dụng ArduinoJson

- Sử dụng thư viện ArduinoJson để chuỗi json.
- Giao tiếp Json thực hiện hàm:
 - **ESP32-CAM**
- {"InitWiFi": "1"}: ESP32-CAM gửi chuỗi báo hiệu cho ESP32 kết nối wifi thành công.
- {"Del_RFID": "<id>"}: ESP32-CAM gửi chuỗi báo hiệu xóa thẻ RFID từ trang web với id được cung cấp thông qua MQTT.
- {"Del_FingerID": "<id>"}: ESP32-CAM gửi chuỗi báo hiệu xóa vân tay từ trang web với id được cung cấp thông qua MQTT.
- {"name": "<tên thay đổi>", "ch_RFID": "<id>"}: Chuỗi nhận từ MQTT và về ESP32-CAM sau đó gửi đến ESP32 để thay đổi tên cho thẻ RFID được lưu trong bộ nhớ.
- {"name": "<tên thay đổi>", "ch_FingerID": "<id>"}: Chuỗi nhận từ MQTT và về ESP32-CAM sau đó gửi đến ESP32 để thay đổi tên cho vân tay được lưu trong bộ nhớ.
- **ESP32**
- {"Warning": "1"}: chuỗi gửi từ Esp32 tới ESP32-CAM thực hiện gửi hình ảnh qua mail thông qua method POST và hàm mail().
- {"message": "<message>", "time": "<time>"}: Gửi message và time thêm lên giá trị cho bảng trong MySQL.

- {“RFID”: [[<4 số RFID1>],[<4 số RFID2>]], “slot”:[<vị trí 1> ,<vị trí 2>], “Name”: [<tên 1>,<tên 2>]}: chuỗi thêm RFID giá trị khởi tạo (gửi dữ liệu lên server bằng method POST).
- {“id”:[<id 1>,<id 2>], “Name”:[<tên 1>,<tên 2>]}: chuỗi thêm Finger giá trị khởi tạo (gửi dữ liệu lên server bằng method POST)
- {“DelFinger”:“1”}: Xóa dữ liệu bằng khi không có RFID nào trong vùng nhớ EEPROM.
- {“DelRFID”:“1”}: Xóa dữ liệu bằng khi không có vân tay nào trong vùng nhớ EEPROM.
- {“type”: <vân tay/ RFID>, “action”: <del/add>, “id”: <id/slot>, “Name”: <tên>, “RFID_code”: [<4 số RFID>]}: Thêm/xóa vân tay/thẻ RFID trên bảng CSDL, nếu có thêm RFID_code thì thêm thẻ RFID nếu không xác định phần tử RFID_code thì thêm vân tay.

3.5.3 Vùng nhớ EEPROM sử dụng

Hệ thống sử dụng vùng nhớ EEPROM để lưu trữ những thông tin cần thiết của hệ thống mật khẩu như các tên, mã thẻ RFID, vùng lưu trữ ID vân tay đã được sử dụng,... Nhóm thực hiện chia các vùng nhớ trong EEPROM của ESP32 theo cấu trúc để lưu thông tin như sau:

Bảng 3.3: Bảng địa chỉ lưu trữ thông tin vùng nhớ EEPROM

Mục đích	Địa chỉ bắt đầu	Số lượng phần tử	Kích thước mỗi phần tử	số lượng dùng	Địa chỉ kết thúc
Kt mật khẩu mặc định được ghi	0	1	1	1	0
Kt mật khẩu admin mặc định được ghi	1	1	1	1	1
Số lượng RFID	2	1	1	1	2
Số lượng vân tay	3	1	1	1	3
Chuỗi mật khẩu	4	1	6	6	9
Chuỗi mật khẩu admin	10	1	8	8	17
Lưu ID thẻ	20	20	4	80	99
Lưu tên thẻ	100	20	10	200	299
Lưu tên vân tay	300	20	10	200	499
Lưu vị trí map với ID vân tay	500	20	1	20	519
Kt đã thiết lập thời gian hệ thống	520	1	1	1	520

CHƯƠNG 4: THI CÔNG HỆ THỐNG

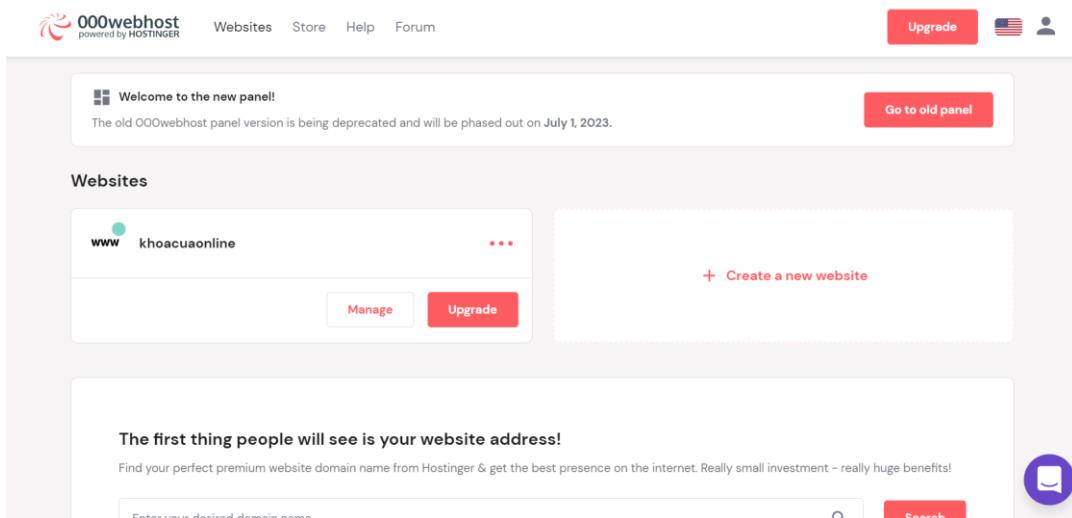
4.1 THIẾT KẾ WEBSITE

4.1.1 Tạo tên miền

Để trang web được hiển thị trên Internet, điều đầu tiên cần phải làm là đăng ký một tên miền cho trang web của mình. Ở trên Internet có nhiều trang web hỗ trợ tạo một trang web từ đơn giản (HTML, CSS) đến nâng cao (database, PHP) dùng cho mục đích cá nhân. Để tạo được trang web quản lý, nhóm chọn trang web cung cấp tên miền miễn phí trên <https://000webhost.com>.

000webhost cung cấp nhiều tính năng mạnh mẽ cho bản free như:

- 10GB băng thông.
- Dung lượng đĩa được cấp: 1GB.
- Số inode được cấp: 20000.
- Quota gửi email hàng ngày: 50.



Hình 4.1: Trang quản lý website trên 000webhost

4.1.2 Thiết kế cơ sở dữ liệu

Cơ sở dữ liệu là nơi lưu trữ các thông tin, dữ liệu của hệ thống. Một cơ sở dữ liệu có thể gồm nhiều bảng lưu trữ.

Truy cập vào quản lý database trên webhost bằng cách: Dashboard -> MySQL Database, ở đây chúng ta tạo một cơ sở dữ liệu để lưu trữ và xử lý thông tin cho hệ thống.

The screenshot shows the 'Databases' section of the phpMyAdmin interface. At the top, there's a 'Create New Database' button with a red circular icon containing a white database symbol. Below it, a text box says: 'Create a database where you can store a lot of information, such as user accounts, posts, comments, and other data that the website needs to function properly.' A table titled 'Manage Databases' lists one database entry:

Name	User	Host	Tables	Size	...
id20314831_dbcuaanninh	id20314831_admin	localhost	15	0MB	•••

Hình 4.2: Trang quản lý database MySQL trên 000WebHost

Nhập vào dấu “...” và chọn truy cập vào phpmyAdmin và tạo các bảng thông tin.

Bảng “refTable”: bảng này sẽ lưu các thông tin về tên các bảng và link sẽ tham chiếu để đọc dữ liệu lên web.

Bảng 4.1: Bảng lưu trữ dữ liệu “ref_table” trên MySQL

Bảng “ref_table”			
Tên cột	Kiểu dữ liệu	Độ dài dữ liệu	Chức năng
id	Int	11	Phân biệt các thông tin tham vấn
device_code	VARCHAR	255	Lưu mã bảo mật của thiết bị
ref_history	VARCHAR	255	Lưu tên bảng sẽ được tham chiếu ứng với mã thiết bị
ref_Finger	VARCHAR	255	Lưu tên bảng sẽ được tham chiếu ứng với mã thiết bị
ref_RFID	VARCHAR	255	Lưu tên bảng sẽ được tham chiếu ứng với mã thiết bị
ref_stream	VARCHAR	255	Lưu lại WiFi IP Address của ESP32-CAM

Bảng “user_infor”: Lưu lại thông tin tài khoản đã được đăng ký

Bảng 4.2: Bảng lưu trữ dữ liệu “user_infor” trên MySQL

Bảng “user_info”			
Tên cột	Kiểu dữ liệu	Độ dài dữ liệu	Chức năng
id	Int	11	Tự động đánh thứ tự phân biệt các hàng với nhau
name	VARCHAR	255	Lưu tên của người tạo tài khoản
username	VARCHAR	255	Lưu tên đăng nhập
password	VARCHAR	255	Lưu mật khẩu đăng nhập
email	VARCHAR	255	Lưu thông tin email
device	VARCHAR	255	Thông tin thiết bị được quản lý bởi tài khoản

Bảng “history<x>”: Lưu lại các hoạt động đóng/mở cửa và các phản thông
điệp của hệ thống mỗi thiết bị sẽ được đánh số x theo id (bảng device_table).

Bảng 4.3: Bảng lưu trữ dữ liệu “History<x>” trên MySQL

Bảng “history<x>”			
Tên cột	Kiểu dữ liệu	Độ dài dữ liệu	Chức năng
id	Int	11	Tự động đánh thứ tự phân biệt các hàng với nhau
Message	VARCHAR	255	Lưu lại thông điệp của hệ thống gửi lên
Timestamp	datetime	*	Lưu lại dấu thời gian message gửi đến (đọc từ RTC)

Bảng RFID<x>: Lưu lại thông tin mở khóa của RFID trên hệ thống. Mỗi thiết bị sẽ được đánh số x theo id (bảng device_table)

Bảng 4.4: Bảng lưu trữ dữ liệu RFID<x> trên MySQL

Bảng RFID<x>			
Tên cột	Kiểu dữ liệu	Độ dài dữ liệu	Chức năng
Slot	Int	11	Lưu thông tin slot RFID
RFID_name	VARCHAR	10	Lưu thông tin tên RFID
RFID_code	Long text	*	Lưu lại 4 chữ số định danh của thẻ RFID

Bảng Finger<x>: Lưu lại thông tin vân tay mở khóa trên hệ thống bao gồm tên và id. Mỗi thiết bị sẽ được đánh số x theo id (bảng device_table).

Bảng 4.5: Bảng lưu trữ dữ liệu Finger<x> trên MySQL

Bảng Finger<x>			
Tên cột	Kiểu dữ liệu	Độ dài dữ liệu	Chức năng
Slot	Int	11	Lưu thông tin slot vân tay
RFID_name	VARCHAR	10	Lưu thông tin tên vân tay

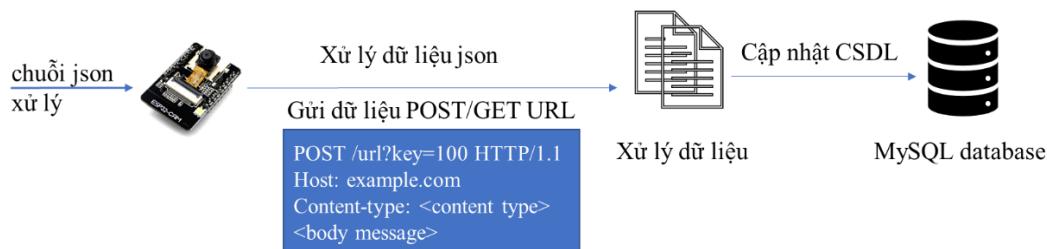
Bảng “device_table”: Với mỗi thiết bị nhóm tạo ra sẽ có một mã thiết bị khác. Bảng này dùng nhằm phục vụ cho việc quản lý mã thiết bị đã được dùng.

Bảng 4.6: Bảng lưu trữ dữ liệu “device_table” trên MySQL

Bảng “device_table”			
Tên cột	Kiểu dữ liệu	Độ dài dữ liệu	Chức năng
id	Int	11	Tự động đánh thứ tự phân biệt các hàng với nhau
device_code	VARCHAR	255	Lưu mã thiết bị có thể được sử dụng
Available	Int	1	Lưu trạng thái đã được sử dụng: “0” đã được sử dụng “1” khả dụng

4.1.3 Lập trình các file xử lý dữ liệu bằng PHP

Các file xử lý dữ liệu bằng PHP sẽ sử dụng các phương thức GET và POST request đã được giới thiệu trong chương 2. Dữ liệu sẽ được gửi lên web server thực hiện xử lý và tác động lên database MySQL.



Hình 4.3: Hoạt động cập nhật dữ liệu lên MySQL

Trước khi xử lý một database ta cần có một kết nối với nó, để kết nối với database ta có thể dùng lệnh:

- mysqli_connect(<sv_host>,<sv_username>,<sv_password>,<database>);

Ví dụ về một đoạn code gửi dữ liệu web bằng phương thức POST kết hợp với GET request cho gửi thông điệp từ ESP32 lên MySQL.

```

<?php
error_reporting(0);
header("Cache-control: no-store");
header("Content-type: text/event-stream");
require_once ('../connection.php');
$p = '';

```

```

$respone = array();
if(isset($_GET['ser_code']))
{
    $sercuritycode = $_GET['ser_code'];
    $sql = "select * from ref_table where device_code = '$sercuritycode'";
    $result = mysqli_query($conn,$sql);
    if($result->num_rows>0)
    {
        $row = mysqli_fetch_array($result);
        $ref_history = $row['ref_history'];
        if(isset($_POST['NoMessage']))
        {
            $NoMes = $_POST['NoMessage'];
            if($NoMes != '')
                $result = $conn->query("select * from $ref_history ORDER BY Timestamp DESC limit $NoMes");
            else
                $result = $conn->query("select * from $ref_history ORDER BY Timestamp DESC");
        }
        else
            $result = $conn->query("select * from $ref_history ORDER BY Timestamp DESC");
        $r = array();
        if($result->num_rows > 0)
        {
            while($row = $result->fetch_assoc())
            {
                $param = array();
                $param['message'] = $row['message'];
                $param['Timestamp'] = $row['Timestamp'];
                array_push($r,$param);
            }
            $respone['success'] = 1;
            $respone['receive'] = array();
            $respone['receive'] = $r;
            $n = json_encode($respone);
            echo $n . "\n\n";
        }
        else
        {
            $respone['success'] = 0;
            $respone['message'] = 'The table is wrong or not exist';
            echo json_encode($respone);
        }
    }
    else
    {
        $respone['success'] = 0;
        $respone['message'] = 'sercurity code is wrong or not exist';
        echo json_encode($respone);
    }
}
else
{
    $respone['success'] = 0;
    $respone['message'] = 'Not enough param to query';
}

```

```
        echo json_encode($respone);  
    }  
?>
```

4.1.4 Thiết kế giao diện website

a) Công cụ quản lý file

Có thể sử dụng trang web của 000webhost để quản lý file của bạn, bằng cách truy cập công cụ “Quản lý file” hoặc bằng công cụ FTP (File Transfer Protocol).

b) Thiết kế trang web đăng ký tài khoản

Đây là trang giúp người sử dụng thiết bị có thể tạo tài khoản để có quyền truy cập xem thông tin giám sát thiết bị. Link truy cập:

<https://khoacuaonline.000webhostapp.com/register>, giao diện được thiết kế như sau:

Trang chủ Quản lý khóa cửa bạn Giới thiệu Liên hệ

Đăng ký tài khoản thiết bị của bạn

Account name*

type your name

Your email

Please type your email

Username*

6-10 characters

Password*

8-16 characters

Retype your password*

8-16 characters

Type your id prod*

code on the product

* is required

Đăng ký

Hình 4.4: Giao diện trang đăng ký tài khoản trên website

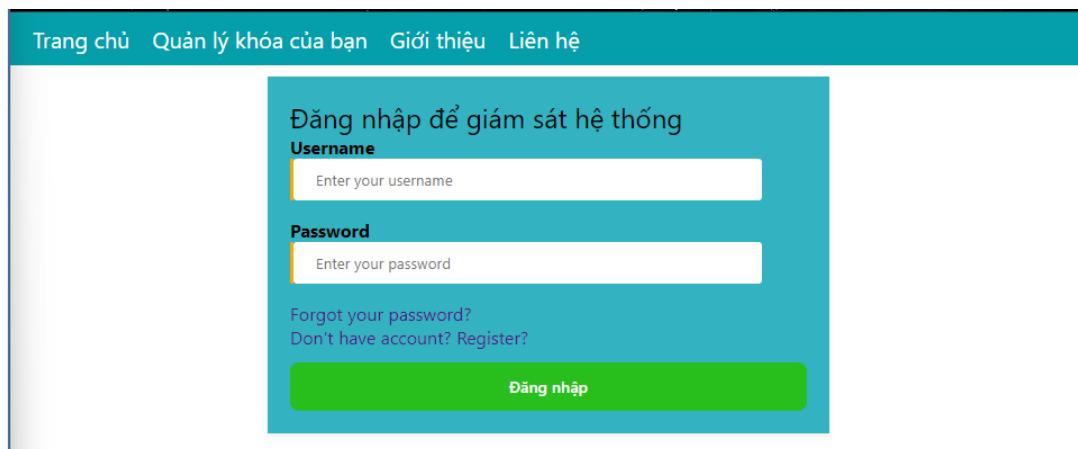
Thuật toán xử lý:

- Khi nhập đầy đủ thông tin và nhấn nút “Đăng ký”. Các thông tin sẽ được update lên bảng user_infor và lưu lại.
- Kiểm tra tên đăng nhập có bị trùng với tên đăng nhập đã được đăng ký trước đó hay không. Nếu trùng thông tin tài khoản hoặc mã được sử dụng, thông báo và

không thực hiện đăng ký tài khoản, ngược lại sẽ đăng ký tài khoản và báo đăng ký thành công.

c) *Thiết kế trang đăng nhập.*

Đây là trang web sẽ giúp người dùng đăng nhập vào để giám sát hoạt động của thiết bị của mình. Link truy cập: <https://khoacuaonline.000webhostapp.com/login>, giao diện được thiết kế như sau:



Hình 4.5: Giao diện trang đăng nhập trên website

Thuật toán xử lý:

- Khi nhập đầy đủ thông tin và nhấn nút “Đăng nhập”, các thông tin sẽ được kiểm tra tại trang web hiện tại.
- Trang sẽ kiểm tra xem có tồn tại tài khoản trong database hay không, nếu không sẽ có thông báo tại trang hiện tại.
- Khi đăng nhập thành công các thông tin sẽ được lưu vào session các biến bao gồm: name, device, username trong bảng “user_infor”. Session là 1 cách để lưu lại phiên đăng nhập người dùng cho phép truy cập trang web đó mà không cần phải đăng nhập lại nhiều lần.

d) *Thiết kế trang đăng xuất*

- Trang này không có giao diện hiển thị. File code chính có tên là logout.php được lưu trong thư mục “login”.

Link truy cập: <https://khoacuaonline.000webhostapp.com/login/logout.php>

- File này dùng để thực hiện xóa các session hiện tại trên trình duyệt và quay trở lại trang đăng nhập khi xóa xong.

e) *Thanh chức năng*

- Thanh chức năng này sẽ được hiển thị đối với các trang giám sát hệ thống.
- Thanh chức năng có chức năng điều hướng ra các trang quản lý khác.
- Khi ở màn hình lớn hơn 580px chiều rộng thanh chức năng sẽ được ở phía trên. Ngược lại, thanh chức năng sẽ hiển thị ở cuối trang. Thanh chức năng luôn hiển thị cả khi cuộn trang.



Hình 4.6: Thanh chức năng được thiết kế trên trang web

- Các chức năng các nút được liệt kê từ trái sang phải như sau:
 - + Nút 1: Nhảy lên đầu trang hiện tại.
 - + Nút 2: Quay về trang thông tin lịch sử cửa.
 - + Nút 3: Xem thông tin RFID, vân tay có trên hệ thống.
 - + Nút 4: Tải lại trang hiện tại để cập nhật thông tin.
 - + Nút 5: Xem hình ảnh được chụp lưu trên hệ thống.
 - + Nút 6: Giám sát camera hệ thống.
- f) Thiết kế trang lịch sử cửa:
 - Để bảo mật cho trang, tránh người dùng khác điều khiển thiết bị của nhau, trước khi vào được trang web sẽ được chèn lệnh để kiểm tra các biến session đang được lưu trên server. Nếu chưa đăng nhập mà truy cập vào link sẽ được chuyển sang trang đăng nhập. Ngược lại, nếu đăng nhập rồi thì sẽ quản lý được thiết bị của người dùng.
 - Trang web được sử dụng để giám sát hoạt động ra vào cửa. Link truy cập: https://khoacuaonline.000webhostapp.com/history_door, Giao diện được thiết kế như sau:

Hình 4.7: Giao diện trang xem lịch sử ra vào cửa

Thuật toán xử lý:

- Trong trang này sẽ thấy một bảng chứa các cột thông tin mà thiết bị đã gửi lên.
- Trang ban đầu sẽ gửi hiển thị 25 lịch sử gần nhất được lưu trong bảng trên CSDL.
- Có thể lọc các giá trị lịch sử như 25, 50, 100, và all (tất cả dữ liệu lưu trong bảng).
- Giá trị được cập nhật lên cơ sở dữ liệu sẽ được hiển thị cập nhật liên tục (chỉ kí 5 giây).

g) Thiết kế trang xem thông tin RFID, vân tay hệ thống.

- Trang này được bảo mật như trang lịch sử cửa. Chỉ người dùng đã đăng nhập mới có thể xem thông tin.
- Trang web dùng để xem các thông tin về RFID và vân tay lưu trữ trong hệ thống. Link truy cập: <https://khoacuaonline.000webhostapp.com/monitor/>, Giao diện được thiết kế như sau:

The screenshot shows a web application interface with the following components:

- Header:** Trang chủ, Quản lý khóa cửa bạn, Giới thiệu, Liên hệ, Xin chào Vũ Trần Hoàng Sơn.
- Toolbar:** Includes icons for back, forward, search, and refresh.
- Form (Top Left):** Cập nhật tên người dùng (Update user name). It has a dropdown menu for 'RFID', an input field for 'id' (value: 1), an input field for 'name' (placeholder: name), and a 'Cập nhật' (Update) button.
- Table 1 (Top Right):** Bảng thông tin RFID lưu trữ (Storage RFID information table). It contains three rows of data:

slot	Name	RFID Card	Xóa
1	Khiem01	[163,28,146,21]	Xóa
2	Son	[210,67,188,46]	Xóa
3	Sonvu	[44,218,130,24]	Xóa
- Table 2 (Bottom Left):** Bảng thông tin vân tay lưu trữ (Storage fingerprint information table). It contains three rows of data:

id	Name	Xóa
1	Finger01	Xóa
2	Khiem	Xóa
3	F_Son	Xóa
- Footer:** Powered by 000webhost.

Hình 4.8: Giao diện trang quản lý RFID và vân tay

- Dữ liệu từ trang web đọc từ cơ sở dữ liệu MySQL bao gồm: slot Name và mã thẻ RFID cho thông tin RFID và id, Name cho thông tin vân tay.
- Dữ liệu được đọc từ 2 bảng RFIDX và Fingerx với x là số id của thiết bị.
- Phần cập nhật tên người dùng sẽ gửi phương thức đến file PHP để cập nhật giá trị cho các topic thuộc feed có tên là chuỗi api tương ứng của thiết bị. Với các

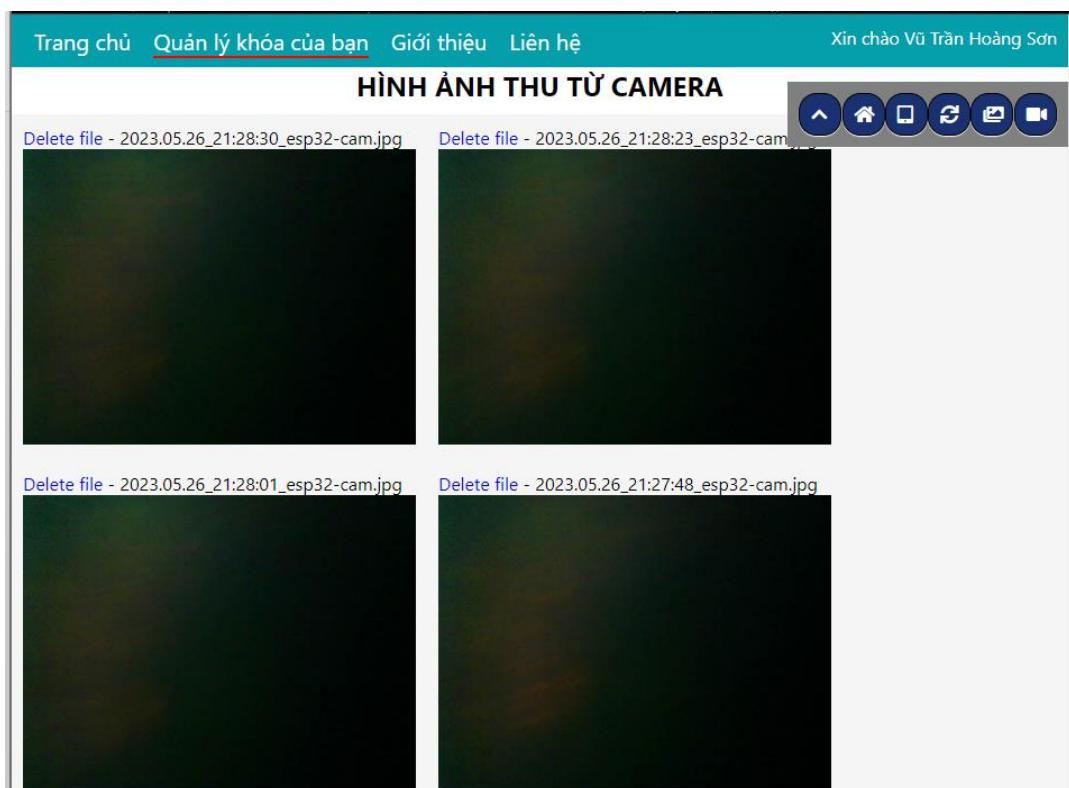
topic được cập nhật giá trị có tên: chNameFinger, chNameRFID. Và nội dung cập nhật là chuỗi json để gửi về ESP32 gọi hàm thay đổi tên.

- Các liên kết xóa ở trên trang web sẽ gửi phương thức đến file PHP để cập nhật các topic thuộc feed có tên là chuỗi api tương ứng. Với các topic được cập nhật giá trị có tên: DelFinger, DelRFID. Và nội dung cập nhật là các số ID muốn xóa.

h) Trang quản lý xem ảnh từ camera.

- Trang này được bảo mật như trang lịch sử cửa. Chỉ người dùng đã đăng nhập mới có thể xem thông tin.

- Trang web dùng để xem các ảnh được chụp từ camera (Khi mở cửa). Link truy cập: <https://khoacuaonline.000webhostapp.com/api/>, giao diện được thiết kế như sau:



Hình 4.9: Giao diện trang quản lý hình ảnh được gửi từ ESP32-CAM

Thuật toán xử lý:

- Các ảnh sẽ được đăng lên thư mục đăng tải ảnh của từng thiết bị (qua file, các thư mục này sẽ có 6 tự cuối để phân biệt với từng thư mục. Thư mục sẽ được tạo khi đăng ký thiết bị thành công.

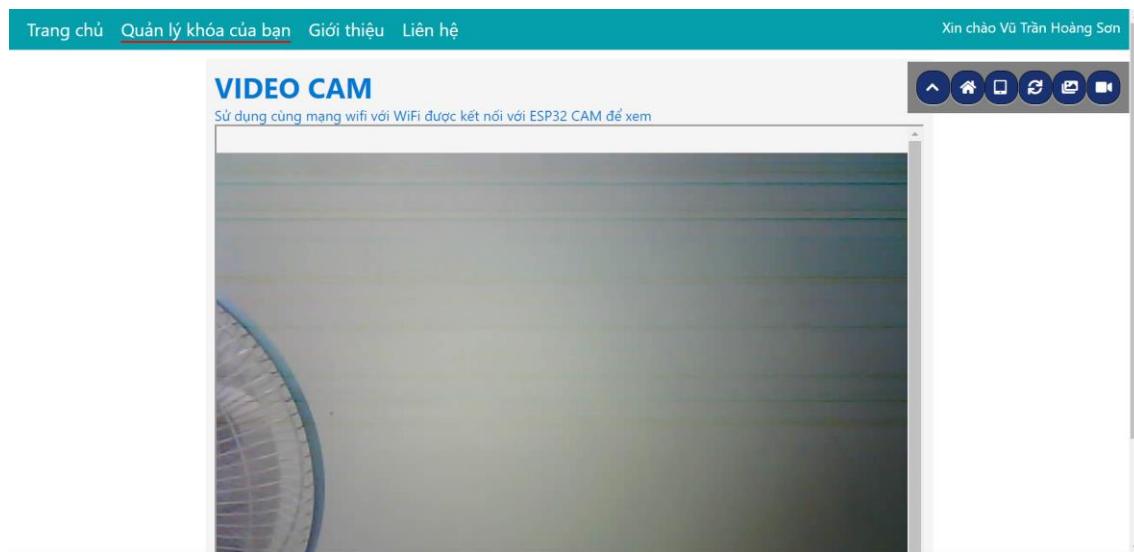
- Có thể xóa ảnh bằng nút nhấn delete file hiển thị trên trang web.
- Sử dụng 1 file php để nhận và chuyển các file hình ảnh vào đúng thư mục cần thiết thông qua giao thức post.

h) Trang truy cập giám sát video camera stream.

- Trang này được bảo mật như trang lịch sử cửa. Chỉ người dùng đã đăng nhập mới có thể xem thông tin.

- Trang web dùng để xem giám sát camera trực tiếp dưới dạng stream video.

Link truy cập: http://khoacuaonline.000webhostapp.com/video_stream/, Giao diện được thiết kế như sau:



Hình 4.10: Giao diện trang giám sát video camera

i) Các trang khác hỗ trợ tài khoản:

Trang đổi mật khẩu: Giúp người dùng đổi mật khẩu mới. Điều kiện để đổi mật khẩu là nhập đúng mật khẩu cũ. Trang chỉ có thể truy cập vào khi đăng nhập được vào tài khoản. Trang truy cập:

<https://khoacuaonline.000webhostapp.com/modify-pass>, giao diện như sau:

Hình 4.11: Giao diện trang hỗ trợ đổi mật khẩu tài khoản

Trang khôi phục tài khoản: Giúp người dùng nếu quên mật khẩu tài khoản có thể truy cập để tạo mật khẩu mới. Điều kiện để lấy lại mật khẩu là người dùng phải điền đúng tên đăng nhập và mã sản phẩm. Giao diện như sau:

Hình 4.12: Giao diện trang khôi phục mật khẩu cho tài khoản

4.2 HƯỚNG DẪN SỬ DỤNG VÀ THAO TÁC

4.2.1 Tài liệu hướng dẫn kết nối

Bước 1: Kết nối các thiết bị và cấp điện cho mạch hoạt động.

Bước 2: Khi thực hiện cấp nguồn xong kiểm tra ESP32-CAM có hiện đèn LED phía sau hay chưa, nếu chưa thì nhấn nút RST trên ESP32-CAM để restart lại.

Bước 3: Thiết lập WiFi bằng cách truy cập vào WiFi có SSID: ESP32-CAM Config (nếu không hiện SSID nhấn RST pin để restart lại ESP32-CAM cho đến khi thấy SSID hiển thị).



Hình 4.13: Kết nối với access point để thiết lập WIFI cho hệ thống

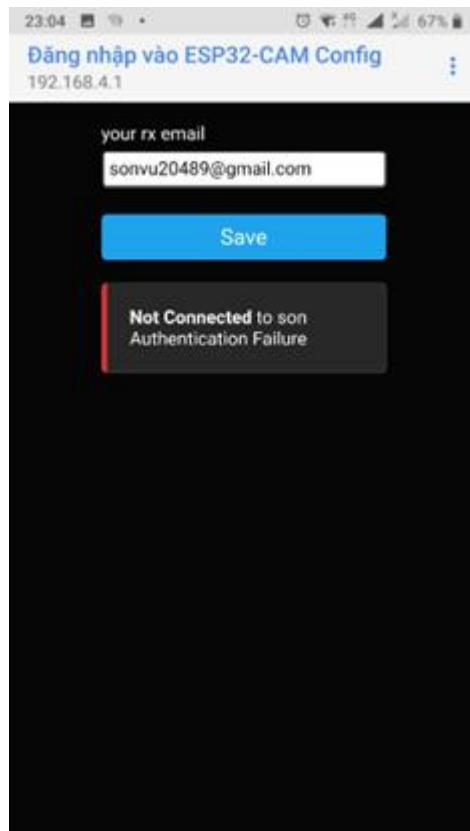
Lưu ý: Khi restart ESP32-CAM nhiều lần mà cũng không hiện SSID kiểm tra lại kết nối camera.

Bước 4: Thực hiện cấu hình:

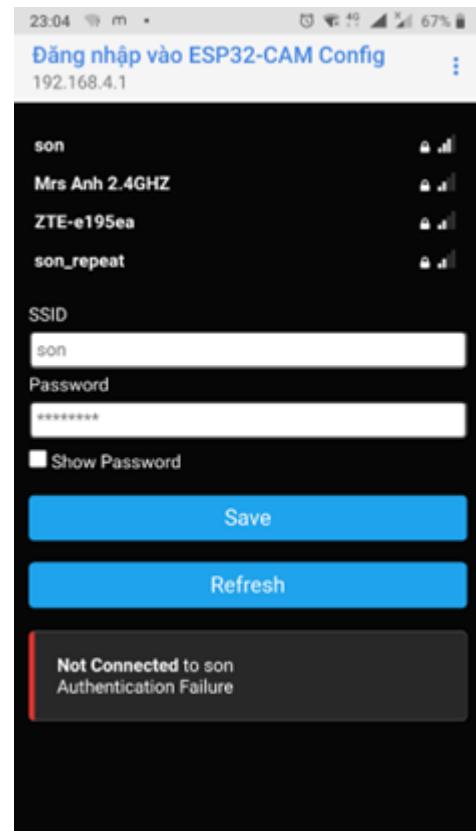
- Cấu hình email nhận cảnh báo (Setup) sau đó nhấn save rồi quay trở lại menu chính.
- Tiếp theo, cấu hình WiFi cho ESP32-CAM và nhấn save. Khi kết nối WiFi thành công ESP32-CAM sẽ chớp tắt Flash 2 lần, nếu không thành công truy cập lại configuration WiFi để thực hiện cấu hình lại.



Hình 4.14: Mục chọn cấu hình



Hình 4.15: Cấu hình email cảnh báo



Hình 4.16: Cấu hình wifi kết nối

Bước 5: Thực hiện đăng ký tài khoản để giám sát hệ thống trên web.

Bước 6: Nhập mật khẩu admin: “12345678” để truy cập vào menu ủy quyền để đổi mật khẩu mặc định “123456” (mật khẩu mở cửa), thêm vân tay, RFID,...

Bước 7: Khi thêm các mật khẩu xong ta có thể dùng các mật khẩu để mở cửa.

4.2.2 Thao tác thực hiện đăng ký tài khoản để giám sát hệ thống

Bước 1: Quét mã QR được dán trên thiết bị để lấy mã thiết bị đăng ký tài khoản.

Bước 2: Tạo tài khoản cho thiết bị bằng cách truy cập website: <https://khoacuaonline.000webhostapp.com/register>.

Diền các thông tin cho tài khoản và mã sản phẩm trên thiết bị, sau đó nhấn nút đăng ký “Đăng ký ngay”.

Bước 3: Đăng nhập hệ thống giám sát qua trang web:

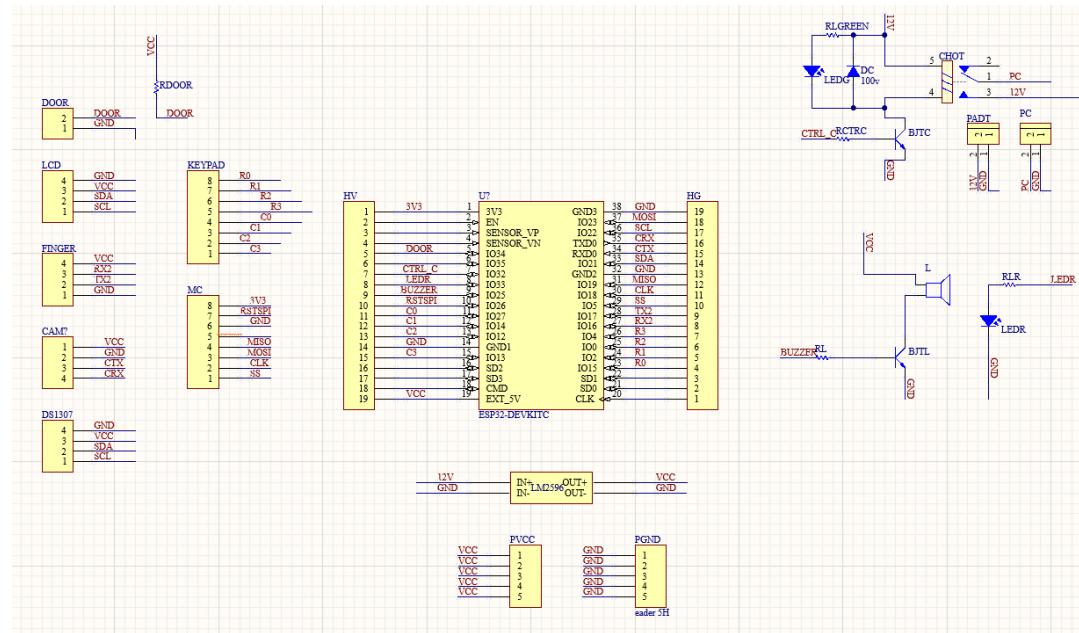
<https://khoacuaonline.000webhostapp.com/login> hoặc nhấn vào “Quản lý khóa của bạn” trên mục điều hướng website để tiến hành đăng nhập.

Bước 4: Thực hiện giám sát hệ thống trên trang web.

4.3 THIẾT KẾ BOARD MẠCH

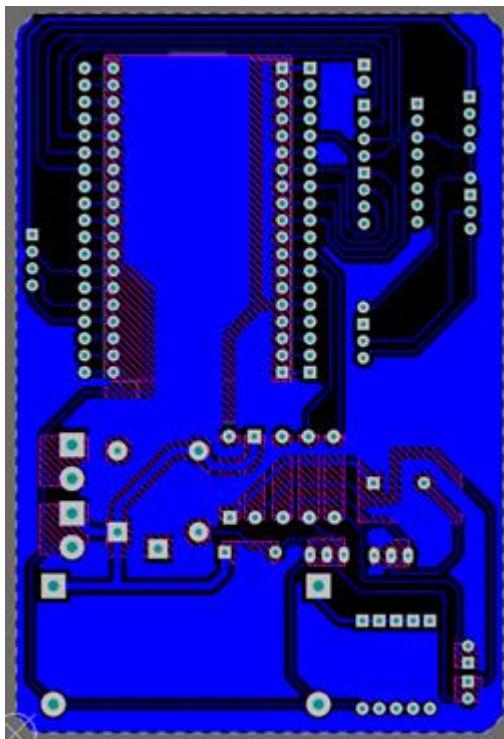
Do tính chất sử dụng nhiều module để thực hiện thiết kế nên phần mạch in của nhóm chủ yếu sẽ bao gồm các headers được nhóm sắp xếp theo thứ tự chân của phần cứng và kết nối với vi điều khiển.

Vì tính chất sử dụng nhiều module nên phần lớn các ngoại vi của mạch đều được nhóm thay thế bằng header để có thể dễ dàng kết nối, nhờ đó tạo thêm tính linh động để có thể dễ dàng thay mới, kiểm tra nếu có lỗi xảy ra từ module.

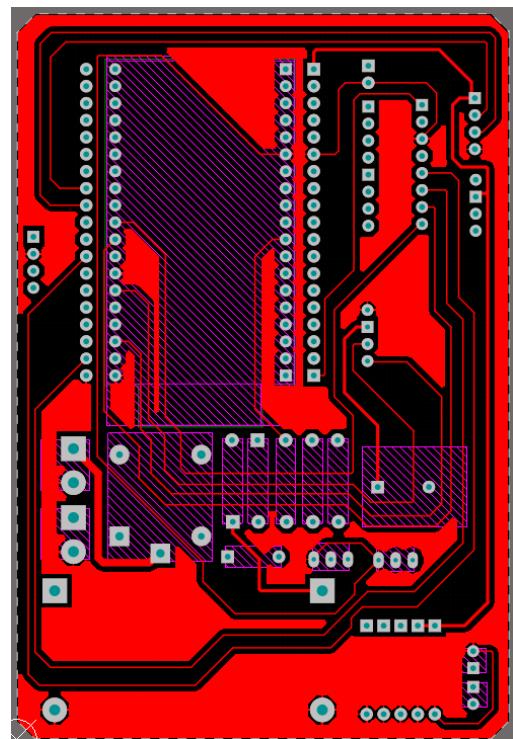


Hình 4.17: Sơ đồ nguyên lý của thiết kế phần cứng của nhóm

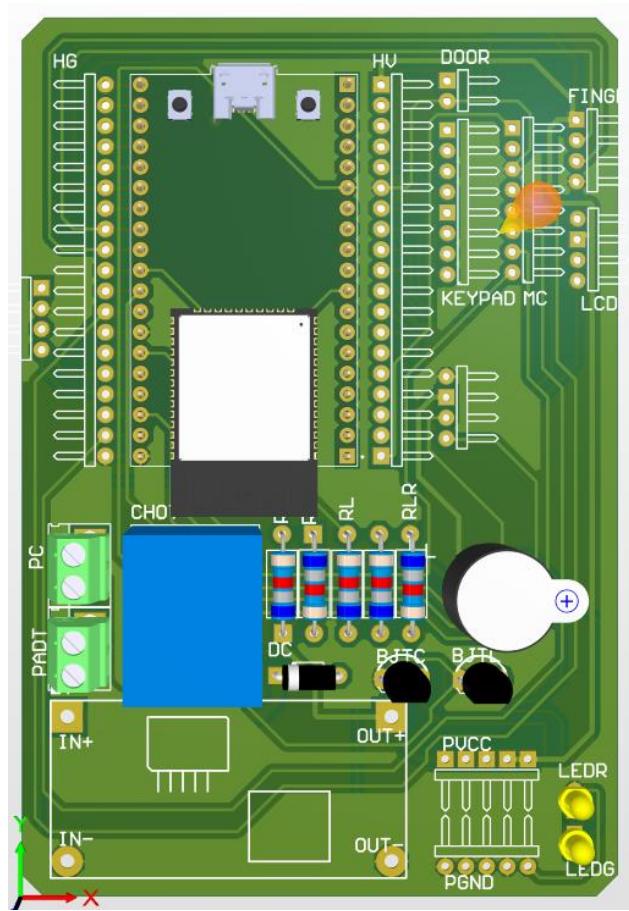
Với tính chất là 1 thiết bị có thẻ tích hợp trên khóa cửa, nên nhóm thực hiện đã tối ưu thiết kế của mình bằng cách đi dây 2 lớp.



Hình 4.18: Sơ đồ dây ở mặt dưới



Hình 4.19: Sơ đồ dây ở mặt trên



Hình 4.20: Thiết kế mạch nhìn từ góc độ 3D

CHƯƠNG 5:

KẾT QUẢ

5.1 KẾT QUẢ ĐẠT ĐƯỢC

Sau quá trình thực hiện đề tài nhóm đã vận dụng được các kiến thức đã học để nghiên cứu và ứng dụng vào thực tiễn. Cũng có một số khó khăn nhất định trong việc thi công và thiết kế hệ thống. Nhưng nhóm đã thực hiện và đạt được một số kết quả sau:

- Tìm hiểu được việc thiết kế hệ thống IoT và lập trình vi điều khiển ESP32.
- Kết hợp các thiết bị module tương tác điều khiển thông qua các thiết bị với nhau.
- Giao tiếp được 2 ESP32 với nhau thông qua chuẩn giao tiếp UART.
- Lập trình cho ESP32-CAM kết nối và tương tác với server và MQTT.
- Xử lý tương tác giữa hệ thống với người dùng thông qua trang web.
- Biết được thêm các giao thức giao tiếp với webserver.
- Suy nghĩ, tìm hiểu các hướng để giải quyết vấn đề đặt ra trong hệ thống.

5.2 KẾT QUẢ MÔ HÌNH THI CÔNG

5.2.1 Mạch in thực tế của nhóm.

Sau khi hoàn thành thiết kế trên phần mềm ở mục 4.3 nhóm đã tiến hành thi công mạch in thực tế.



Hình 5.1: Mạch thực tế của nhóm sau khi thi công

Cụm header bên trái sẽ được nhóm sử dụng để kết nối với ESP32-CAM, các cụm header bên trái được nhóm thiết kế để kết nối với các thiết bị, cụ thể như sau: theo thứ tự từ trái qua phải và từ trên xuống dưới, các thiết bị được kết nối lần lượt là module keypad, MFRC-522 NFC 13.56MHz, cảm biến vân tay AS608 và LCD 20x4 I2C. Relay dùng để điều khiển chốt điện, mạch nguồn và buzzer được nhóm thiết kế để hàn kết nối trực tiếp vào mạch in. Mạch có thêm 2 LED để báo trạng thái cửa đóng và mở.

5.2.2 Mô hình sản phẩm của nhóm.

Mô hình sản phẩm được nhóm thực hiện như hình sau:



Hình 5.2: Mô hình sản phẩm của nhóm

Trong đó khối LCD, keypad và vân tay cảm ứng có sự tương tác trực tiếp nên được nhóm thiết kế ở bên ngoài vỏ hộp, RFID được nhóm ẩn bên trong hộp sản phẩm. ESP32-CAM được đặt trước cửa nhằm mục đích mô tả lại hành động quan sát các hành vi ở trước cửa.

5.2.3 Kết quả của sản phẩm.

Các tính năng chính của sản phẩm được nhóm thử nghiệm và ghi lại kết quả như sau:

- **Màn hình hiển thị khi bắt đầu hoạt động.**

Đây cũng là màn hình hiển thị. Menu truy cập (menu chờ xác minh mật khẩu).



Hình 5.3: Menu truy cập chờ quét KEY xác nhận

Menu truy cập hiển thị các thông tin gồm có các thông tin về thời gian, trạng thái cửa đang đóng hoặc mở, khoảng thời gian báo động sẽ bắt đầu.

- **Menu Admin thực hiện các chức năng xử lý về KEY của sản phẩm**



Hình 5.4: Menu Admin điều hướng tùy chọn

Để truy cập được menu này, ở menu truy cập, người dùng cần nhập đúng mật khẩu Admin và nhấn “B” để xác nhận. Trong menu này, người dùng có thể lựa chọn các loại KEY mình muốn xử lý thông qua các phím A, B, C.

- **Menu Password Option**



Hình 5.5: Menu Password Option

Menu này cho phép người dùng thực hiện các lựa chọn: thay đổi password người dùng (lựa chọn “A”), đặt mật khẩu về mật khẩu mặc định (lựa chọn “B”), thay đổi mật khẩu Admin (lựa chọn “C”).

Khi thực hiện thay đổi mật khẩu người dùng, người dùng cần nhập lại chính xác mật khẩu của mình và nhấn “B” để xác nhận, điều này cũng tương tự khi tiến hành thay đổi mật khẩu Admin.



Hình 5.6: Màn hình yêu cầu nhập mật khẩu cũ

Sau khi xác nhận mật khẩu, nếu chính xác, hệ thống sẽ yêu cầu người dùng nhập thêm mật khẩu mới. Khi nhập đủ độ dài của mật khẩu, cụ thể là 6 kí tự đối với user và 8 kí tự đối với Admin, mật khẩu sẽ tự động được lưu vào bộ nhớ của thiết bị.



Hình 5.7: Màn hình yêu cầu nhập mật khẩu mới

- **Menu RFID Option**

Menu này được nhóm thiết kế để sử dụng cho việc xử lý một số chức năng liên quan đến RFID và xử lý xóa hết hệ thống về ban đầu.



Hình 5.8: Menu RFID Option

Menu này thực hiện các chức năng: thêm thẻ NFC mới (lựa chọn “A”), xóa thẻ NFC (lựa chọn “B”), đặt lại toàn bộ hệ thống (lựa chọn “C”).

Để thêm thẻ NFC mới, ta tiến hành nhập lựa chọn “A” màn hình hiển thị sẽ yêu cầu thêm thẻ mới vào hệ thống, ta tiến hành thêm bằng cách đưa thẻ mới lại gần khu vực bàn phím của hệ thống.



Hình 5.9: Thêm thẻ RFID mới

Trong màn hình hiển thị này, số ở góc trên bên phải biểu thị cho số thẻ NFC đã tồn tại trong bộ nhớ.

Khi tiến hành xóa thẻ NFC, người dùng có thể tiến hành xóa NFC theo 2 cách, đưa thẻ muốn xóa lại gần khu vực muốn xóa hoặc có thể nhập từ bàn phím số thứ tự của thẻ NFC muốn xóa.



Hình 5.10: Màn hình xóa thẻ NFC

- **Menu Finger Option**

Menu này thực hiện một số chức năng liên quan đến vân tay.



Hình 5.11: Menu Finger Option

Menu này thực hiện một số tính năng: thêm vân tay (lựa chọn “A”), xóa vân tay (lựa chọn “B”) và lựa chọn “C” hiện tại chưa có chức năng.

Để thêm vân tay mới, ta tiến hành nhấn phím “A” và làm theo các hướng dẫn hiển thị trên màn hình.



Hình 5.12: Màn hình thêm vân tay

Màn hình này sẽ hiển thị số ID vân tay hiện có ở góc trên bên trái của màn hình, và màn hình hiển thị chính sẽ thông báo ID mới của vân tay tiếp theo sẽ được thêm.

Để xóa vân tay, ta nhấn “B” và nhập ID của vân tay muốn xóa vào và nhấn “B” để xác nhận xóa.



Hình 5.13: Màn hình xóa vân tay

5.2.4 Kết quả phần mềm

Giao diện trang web giám sát được thiết kế dễ dàng sử dụng đối với người dùng ở trên website sẽ có các trang khác như trang chủ, giới thiệu, liên hệ để xem thông tin về đề tài và sinh viên thực hiện...

- **Trang lịch sử hệ thống**

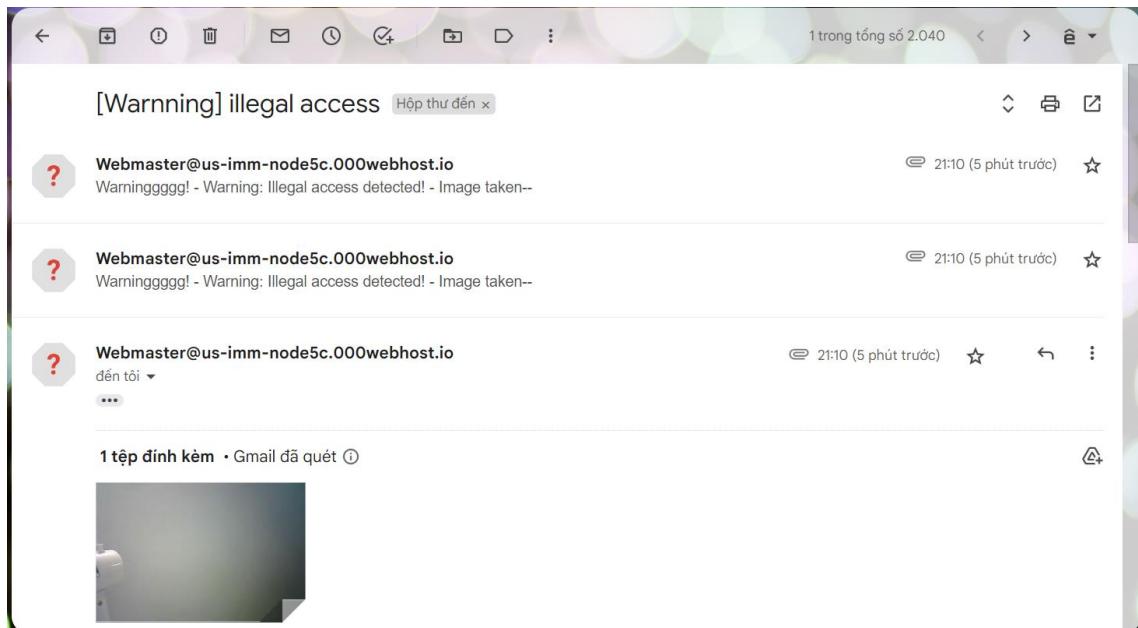
Các tin nhắn của hệ thống được gửi lên (hoạt động/mở cửa, cảnh báo đột nhập trái phép) có các chế độ lọc và thuật toán xử lý được đề cập ở chương trước (phần 4.1.4f).

LỊCH SỬ RA VÀO CỬA	
Time	Message
2023-06-20 21:17:40	Door is closed
2023-06-20 21:17:22	Door is opened by RBaoTruong
2023-06-20 21:14:38	Door is closed
2023-06-20 21:14:29	Door is opened by RFID01
2023-06-20 21:09:45	Illegal access warning!
2023-06-20 21:07:59	Door is closed
2023-06-20 21:07:45	Door is opened by RFID02
2023-06-20 21:07:21	Door is closed
2023-06-20 21:07:07	Door is opened by RFID01
2023-06-20 21:06:36	Door is closed
2023-06-20 21:06:08	Door is opened by RFID01
2023-06-20 00:20:26	Door is closed
2023-06-20 00:20:18	Door is opened by Finger01
2023-06-20 00:19:22	Door is closed
2023-06-20 00:19:16	Door is opened by Son
2023-06-20 00:18:05	Door is closed

Hình 5.14: Kết quả hiển thị 25 thông điệp trên trang web

- **Email cảnh báo**

Khi có cảnh báo hệ thống sẽ gửi một thông điệp lưu lại trên trang lịch sử cửa. Và lúc này mỗi 10 giây hệ thống sẽ gửi email cảnh báo đến email được cấu hình trong menu thiết đặt wifi hệ thống (phần 4.2.2).



Hình 5.15: Email cảnh báo được gửi từ hệ thống

- **Trang thông tin khóa truy cập**

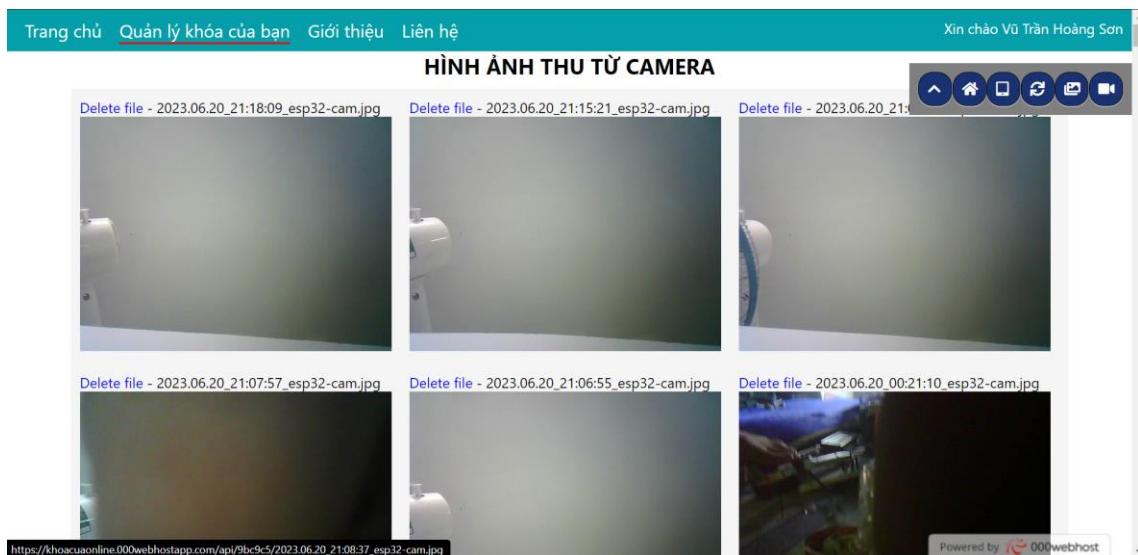
Trang thông tin về KEY truy cập. Có thể cập nhật tên cho các KEY người dùng và xóa KEY trên web.



Hình 5.16: Bảng thông tin KEY lưu trữ và cập nhật

- **Trang quản lý, xem những hình ảnh của hệ thống**

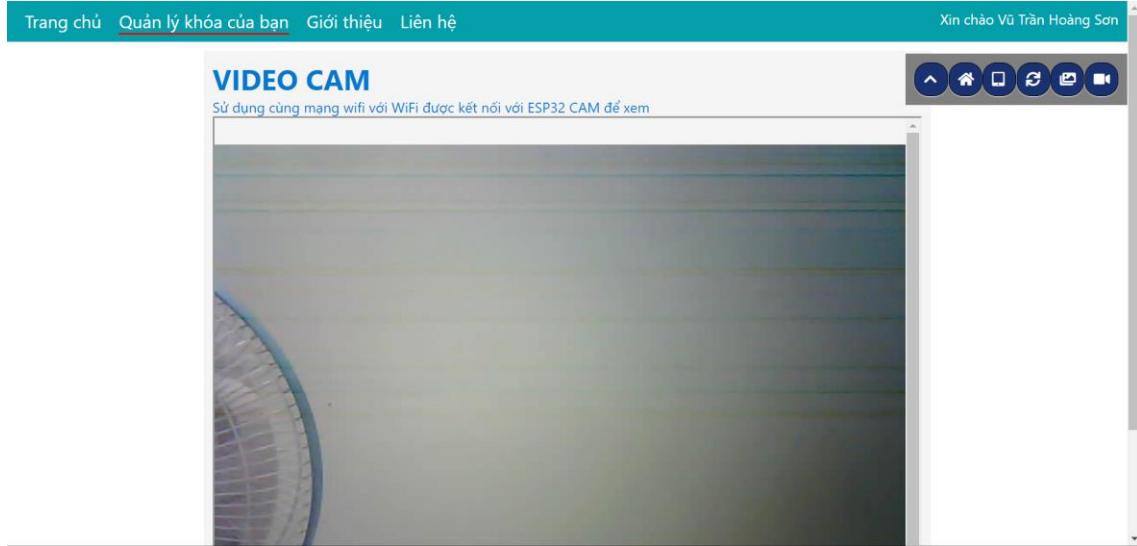
Các hình ảnh về hoạt động mở cửa sẽ được hệ thống chụp và lưu trên các thư mục của trang web. Giúp giám sát được hoạt động vào cửa.



Hình 5.17: Lưu trữ các hình ảnh mở cửa trên trang web

- **Trang giám sát hoạt động của camera**

Trang hỗ trợ giám sát hoạt động khi có cảnh báo về đột nhập, phá khóa. Đồng thời giúp tránh việc di chuyển lên để giám sát ai đang phá cửa nhà của mình.



Hình 5.18: Video stream giám sát hệ thống thông qua camera

- **Tính năng thay đổi tên người dùng**

Tính năng này sẽ được thực hiện trên web của sản phẩm. Ban đầu, khi thêm KEY mới, KEY sẽ được đặt tên theo mặc định của hệ thống, nếu muốn thay đổi tên người dùng cần truy cập vào website của sản phẩm và tiến hành lựa chọn KEY muốn thay đổi tên.



Hình 5.19: Tên người dùng trước và sau khi đổi

5.3 NHẬN XÉT VÀ ĐÁNH GIÁ

Sau thời gian nghiên cứu, thực hiện và kiểm tra, sản phẩm của nhóm đã đáp ứng được các yêu cầu cần thiết của một thiết bị khóa cửa an ninh.

- **Về phần cứng**

- Mạch nhỏ gọn, hoạt động ổn định.
- Tốc độ truyền gửi nhận dữ liệu nhanh.
- Có buzzer báo động.
- Các KEY được xác nhận nhanh và chính xác.
- Hệ thống được thiết kế dễ dàng sử dụng.

- **Về phần mềm**

- Giao diện hiển thị trên LCD thân thiết, đầy đủ các thông tin cần thiết.
- Giao diện web dễ dàng sử dụng và tương tác.
- Có nút nhấn giúp tắt buzzer cảnh báo quên đóng cửa trong những trường hợp cần cửa luôn mở.
- Số lượng vân tay và thẻ NFC lưu trữ được lên đến 20.
- Tự động gửi mail cảnh báo khi nhập sai mật khẩu nhiều lần.

- **Những vấn đề chưa giải quyết:**

- Chưa có hộp bảo vệ phần cứng.
- Có độ trễ xảy ra giữa việc cập nhật dữ liệu (1-7 giây) lên server.
- Vấn đề về bất đồng bộ thông tin giữa trang web và gấp sự cố về internet (cập nhật dữ liệu).
- Giám sát camera cần truy cập cùng mạng với hệ thống kết nối.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 KẾT LUẬN

Sau thời gian tìm hiểu, nghiên cứu các tài liệu, tổng hợp các kiến thức đã học và thực hiện đề tài cũng như được sự hướng dẫn của GVHD. Nhóm đã hoàn thành đồ án tốt nghiệp với đề tài “**THIẾT KẾ VÀ THI CÔNG HỆ THỐNG KHÓA CỦA AN NINH**”.

Mạch hoạt động đã đạt được yêu cầu đặt ra như:

- Hoạt động mở khóa với 3 loại bảo mật: RFID, vân tay, nhập mật khẩu.
- Hệ thống có khả năng báo động bằng buzzer.
- Cài đặt được email nhận cảnh báo.
- Cảnh báo thông qua email trên server.
- Gửi nhận dữ liệu cần giám sát lên CSDL của MySQL.
- Có thể cập nhật các thông tin của thẻ, vân tay thông qua trang web.
- Khi không có kết nối mạng, phần mở đóng cửa, thêm/xóa mật khẩu xác thực, reset... vẫn hoạt động bình thường. Phần kết nối, xử lý internet không thể hoạt động (các xử lý bên ESP32-CAM).
- Hệ thống hoạt động ổn định sau thời gian hoạt động liên tục.
- Có thể xem được hình ảnh, video được quay trực tiếp và theo dõi lịch sử ra vào cửa trên trang web.

Tuy nhiên, thiết bị vẫn còn một số nhược điểm vẫn chưa được khắc phục như không thể xem thông tin, giám sát, các thông tin cập nhật của hệ thống khi không có kết nối internet.

6.2 HƯỚNG PHÁT TRIỂN

Vì thời gian thực hiện đề tài có giới hạn nên vẫn còn nhiều chức năng được nhóm phát triển chưa tròn vẹn dẫn đến hệ thống còn nhiều khuyết điểm cần sửa đổi, nâng cấp:

- Thiết kế hộp bảo vệ sản phẩm.
- Giải pháp đồng bộ dữ liệu giữa sản phẩm và cơ sở dữ liệu.
- Tối ưu tính độ trễ khi giao tiếp giữa server và sản phẩm.
- Thay đổi chuyển đổi từ gửi mail trên host thành gửi mail thông qua smtp mail trên ESP32 (tối ưu bộ nhớ heap).
- Giám sát camera tích hợp từ xa thông qua internet bằng phương pháp stream video server hoặc NAT port.
- Thêm, xóa mật khẩu user, thay đổi mật khẩu admin từ trang web của sản phẩm.
- Gửi tín hiệu điều khiển mở cửa từ trang web và theo dõi được trạng thái cửa từ trang web của sản phẩm.

PHỤ LỤC

Link github code của hệ thống:

https://github.com/sonvu20489/DATN_Sonyth_Khiemhdt/tree/main/Code

Code Arduino:

- Code chương trình chính trên ESP32:

```
#include <Arduino.h>
#include <EEPROM.h>
#include "MyLCD.h"
#include "MyKeypad.h"
#include "MyWarning.h"
#include "MyRFID.h"
#include "Myfinger.h"
#include "global.h"
#include <LiquidCrystal_I2C.h>
#include <RTClib.h>
#define EEPROMSIZE 525
#define SERIAL_BAUD 115200
#define ADD_SETTIME 520
bool is_door_close = 1;
uint8_t prohibit = 0;
char dateT [20];
RTC_DS1307 rtc;
bool InitWifi = 0;
LiquidCrystal_I2C lcd(0x27, 20, 4);
void setup() {
    // put your setup code here, to run once:
    rtc.begin();
    Serial.begin(SERIAL_BAUD);
    Serial.setRxBufferSize(1024);
    Serial.print("Hello this is LVTN");

    EEPROM.begin(EEPROMSIZE);
    if(EEPROM.read(ADD_SETTIME) != 127)
    {
        rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
        EEPROM.write(ADD_SETTIME,127);
        EEPROM.commit();
    }
    Serial.println(EEPROM.read(520));
    init_LCD();
    init_keypadcheck();

    lcd.clear();
    while(InitWifi == 0)
    {
        customkey = customKeypad.getKey();
        if(customkey == 'D')
        {
            InitWifi = 1;
        }
        ReadJson();
        PrintL("Wait for init WiFi",1,0);
        PrintL(" ESP32-CAM Config ",2,0);
        PrintL(" Press D to skip ",3,0);
    }
}
```

```

    }

    MenuDisplay(menu);
    init_RFID();
    init_warning();
    finger_init();

}

void loop() {
    // put your main code here, to run repeatedly:
    customkey = customKeypad.getKey();
    lastdoorstate = doorstate;
    ReadJson();
    doorstate = digitalRead(DOOR_SEN_PIN);
    if(customkey == 'A' && menu->MenuID != ACCMENU)
    {
        if(menu->Activation1 != NULL)
            menu->Activation1(menu->MenuID,OPTION1);
        if(ret_fr_fcntp != 1)
        {
            menu = (menu->Menulist1==NULL)?menu:menu->Menulist1;
            MenuDisplay(menu);

        }
        ret_fr_fcntp = 0;
    }
    else if(customkey == 'B' && menu->MenuID != ACCMENU)
    {
        if(menu->Activation2 != NULL)
            menu->Activation2(menu->MenuID,OPTION2);
        if(ret_fr_fcntp != 1)
        {
            menu = (menu->Menulist2==NULL)?menu:menu->Menulist2;
            MenuDisplay(menu);
        }
        ret_fr_fcntp = 0;
    }
    else if(customkey == 'C' && menu->MenuID != ACCMENU)
    {

        if(menu->Activation3 != NULL)
            menu->Activation3(menu->MenuID,OPTION3);
        if(ret_fr_fcntp != 1)
        {
            menu = (menu->Menulist3==NULL)?menu:menu->Menulist3;
            MenuDisplay(menu);
        }
        ret_fr_fcntp = 0;
    }
    else if(customkey == 'D')
    {

        if(menu->MenuID!= ADMINMENU && menu->MenuID!= ACCMENU)
        {
            menu = &AdminMenu;
            MenuDisplay(menu);
        }
        else if (menu->MenuID == ADMINMENU)
        {
            menu = menu->Menulist0;
        }
    }
}

```

```

    MenuDisplay(menu);
    lcd.setCursor(14,0);
    if(is_door_close == 0)
    {
        lcd.print("|OPEN ");
    }
    else
        lcd.print("|CLOSE");
    }

}

if(menu->MenuID == ACCMENU)
{
    DateTime now = rtc.now();
    sprintf(dateT,"%02d/%02d %02d:%02d:%02d",
    now.day(),
    now.month(),
    now.hour(),
    now.minute(),
    now.second());
    lcd.setCursor(0,0);
    lcd.print(dateT);
    handle_key();
    handle_RFID();
    handle_finger();
}
emergency_btn();
Warning_door_open();
Warning_wrongPass();

}

void ReadJson()
{
    while(Serial.available())
    {
        DynamicJsonDocument json(1024);
        DeserializationError error = deserializeJson(json, Serial);
        if(error)
        {
            return;
        }
        else
        {
            if(json["InitWiFi"] == "1")
            {
                InitWifi = 1;
            }
            if(json["Del_RFID"])
            {
                int ID_DEL = json["Del_RFID"];
                deleteRFID_ID(ID_DEL);
                MenuDisplay(menu);
            }
            if(json["Del_FingerID"])
            {
                int ID_DEL = json["Del_FingerID"];
                deleteFingerprint(ID_DEL);
                MenuDisplay(menu);
            }
        }
    }
}

```

```

    if(json["name"] && json["ch_RFID"])
    {
        const char * name = json["name"];
        int ch_id = json["ch_RFID"];
        RFID_changename(name, ch_id);
        MenuDisplay(menu);
    }
    if(json["name"] && json["ch_FingerID"])
    {
        const char * name = json["name"];
        int ch_id = json["ch_FingerID"];
        Finger_changename(name, ch_id);
        MenuDisplay(menu);
    }
}
}

```

- Code chương trình chính trên ESP32-CAM:

```

#include "MQTT.h"
#include "esp_camera.h"
#include <FS.h>
#include <driver/gpio.h>
#include <WiFi.h>
#include <WiFiManager.h>
#include <ArduinoJson.h>
#include <SPIFFS.h>
#include "esp_timer.h"
#include "img_converters.h"

#include "Arduino.h"
#include "fb_gfx.h"
#include "soc/soc.h" //disable brownout problems
#include "soc/rtc_cntl_reg.h" //disable brownout problems
#include "esp_http_server.h"
#include "app_httpd1.h"
#include <pgmspace.h>

#define BTN_PIN 2
#define FLASH_PIN 4
#define LED_BUILTIN_PIN 33
char email[100];
bool shouldSaveConfig = false;
const char* configPath = "/config.json";
bool Photo = false;

WiFiManager wifiManager;

#define PART_BOUNDARY "12345678900000000000987654321"

#define CAMERA_MODEL_AI_THINKER
#include "esp32cam_pin.h"
void setup() {
    WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // disable brownout
detector
    /*setting IO port*/
    Serial.begin(115200);
    Serial.setRxBufferSize(1024);
}

```

```

pinMode(LED_BUILTIN_PIN, OUTPUT);
digitalWrite(LED_BUILTIN_PIN, LOW);
initSPIFFS();
pinMode(BTN_PIN, INPUT);
pinMode(FLASH_PIN, OUTPUT);
digitalWrite(FLASH_PIN, LOW);
Serial.setDebugOutput(true);
/*setting to init camera*/
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sscb_sda = SIOD_GPIO_NUM;
config.pin_sscb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;

if(psramFound()){
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_CIF;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

/* Camera init*/
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}
delay(500);
if(digitalRead(BTN_PIN) == LOW)
{
    Serial.println("This will reset your WiFi Config");
    Serial.println("You have 5s to Cancel");
    Serial.println("Unpress to Cancel");
    bool buttonState = monitorWipeButton(5000);
    if (buttonState == true && digitalRead(BTN_PIN) == LOW)
    {
        wifiManager.resetSettings();
        Serial.println("Reset Done");
        ESP.restart();
    }
    else

```

```

    {
        Serial.println("Reset Config Cancel");
    }
}

/*WiFi Connection*/
WiFiManagerParameter custom_email("email","your rx email",
email,100);
//set config save notify callback
wifiManager.setSaveConfigCallback(saveConfigCallback);
wifiManager.addParameter(&custom_email);
std::vector<const char *> menu =
{"wifi","info","param","sep","restart","exit"};
wifiManager.setMenu(menu);
wifiManager.setClass("invert");
// fetches ssid and pass from eeprom and tries to connect
// if it does not connect it starts an access point with the
specified name
// here "AutoConnectAP"
// and goes into a blocking loop awaiting configuration
bool res;
res = wifiManager.autoConnect("ESP32-CAM Config","88888888");
/*save email receive warning to variable email*/
strcpy(email, custom_email.getValue());
/*save config to file from SPIFF file*/
if(shouldSaveConfig)
{
    Serial.println("Saving config");
    DynamicJsonDocument doc(200);
    doc["email"] = email;
    File configFile = SPIFFS.open(configPath,FILE_WRITE);
    if(!configFile)
    {
        Serial.println("Failed to open config file for writing");
    }
    serializeJson(doc,configFile);
    configFile.close();
}
/*connect successfully -> blink flash and show the connected IP*/
if(res)
{
    digitalWrite(FLASH_PIN,HIGH);
    delay(500);
    digitalWrite(FLASH_PIN,LOW);
    delay(500);
    digitalWrite(FLASH_PIN,HIGH);
    delay(500);
    digitalWrite(FLASH_PIN,LOW);
    delay(500);
    Serial.print("Camera Stream Ready! Go to: http://");
    Serial.println(WiFi.localIP());
    update_url_streamIpAddress2String(WiFi.localIP());
    Serial.println();
}

else
{
    Serial.println("Failed to connect or hit timeout");
}
/*setting callback MQTT subscriber*/
feed2.setCallback(callback);

```

```

feed3.setCallback(callback2);
feed4.setCallback(callback3);
feed5.setCallback(callback3);
/*setting feed subscribe*/
mqtt.subscribe(&feed5);
mqtt.subscribe(&feed4);
mqtt.subscribe(&feed3);
mqtt.subscribe(&feed2);
// Start streaming web server
startCameraServer();
Serial.println("{\"InitWiFi\":\"1\"}");
/*print heap remain heap memory*/
}

void loop() {

ReadJson();
MQTT_connect();
mqtt.processPackets(2000);
if(! mqtt.ping()) {
    mqtt.disconnect();
}
}

void saveConfigCallback() {
Serial.println("Should save config");
shouldSaveConfig = true;
}

void initSPIFFS() {
Serial.println("mounting FS...");
if(SPIFFS.begin())
{
    Serial.println("Mounted file system");
    if(SPIFFS.exists(configPath))
    {
        Serial.println("reading config file");
        File configFile = SPIFFS.open(configPath);
        if(configFile)
        {
            Serial.println("opened config file");
            size_t size = configFile.size();
            // Allocate a buffer to store contents of the file.
            std::unique_ptr<char[]> buf(new char[size]);
            configFile.readBytes(buf.get(), size);
            DynamicJsonDocument doc(200);
            DeserializationError Error = deserializeJson(doc,buf.get());
            serializeJson(doc, Serial);
            if(Error)
            {
                Serial.println("Failed load json config");
                return ;
            }
            else
            {
                Serial.println("\nParsed json");
                strcpy(email,doc["email"]);
                Serial.println(email);
            }
        }
    }
}

```

```

        }
    }
    else
    {
        Serial.println("File not Found!");
    }
}
else
{
    Serial.println("Failed to mount file FS");
    //end read
}
}

///////////////////////////////
bool monitorWipeButton(uint32_t interval) {
    unsigned long currentMillis = millis(); // grab current time
    while (millis() - currentMillis < interval) {
        int timeSpent = (millis() - currentMillis) / 1000;
        Serial.println(timeSpent);
        // check on every half a second
        if (((uint32_t)millis() % 10) == 0) {
            if (digitalRead(BTN_PIN) != LOW) {
                return false;
            }
        }
    }
    return true;
}
void ReadJson()
{
    while(Serial.available()>0)
    {
        DynamicJsonDocument json(1024);
        DeserializationError error = deserializeJson(json, Serial);
        if(error)
        {
            return;
        }
        else
        {
            /*Sendphoto using POST method*/
            Photo = json["photo"];
            if(Photo == 1)
            {
                sendPhoto();
            }
            /*sending msg and time*/
            if((json["message"])&&(json["time"]))
            {
                send_message(json["message"],json["time"]);
            }
            /*sending warning to email and */
            if(json["Warning"]=="1")
            {

                sendPhoto(email);

            }
            if((json["RFID"])&&(json["pos"])&&(json["Name"]))

```

```

{
    JSONArray RxRFID = json["RFID"];
    JSONArray Rxslot = json["pos"];
    JSONArray RxName = json["Name"];
    String sendRFID;
    String sendslot;
    String sendName;
    serializeJson(RxRFID,sendRFID);
    serializeJson(Rxslot,sendslot);
    serializeJson(RxName,sendName);
    sendReadRFID(sendRFID, sendslot, sendName);
}
if(json["Rst_WiFi"] == "1")
{
    wifiManager.resetSettings();
    Serial.println("Reset Done");
    ESP.restart();
}
if((json["id"]) && (json["Name"]))
{
    JSONArray RxFingerId = json["id"];
    JSONArray RxFingerName = json["Name"];
    String sendFingerId;
    String sendFingerName;
    serializeJson(RxFingerId,sendFingerId);
    serializeJson(RxFingerName,sendFingerName);
    sendReadFinger(sendFingerId, sendFingerName);
}
if(json["type"] && json["action"] && json["id"] && json["name"]
&& json["RFID_code"])
{
    JSONArray RFID_code = json["RFID_code"];
    String sendRFIDcode;
    serializeJson(RFID_code,sendRFIDcode);

    sendaction(json["type"],json["action"],json["id"],json["name"],
    sendRFIDcode);
}
else if(json["type"] && json["action"] && json["id"] &&
json["name"])
{
    sendaction(json["type"],json["action"],json["id"],json["name"],"NULL");
}

if(json["DelRFID"]=="1")
{
    sendReadRFID("None","NULL","NULL");
}
if(json["DelFinger"]=="1")
{
    sendReadFinger("None","NULL");
}
}
}
}

```

TÀI LIỆU THAM KHẢO

- [1] Hà Lưu Phương Lê, Phạm Thị Dung, “Thiết kế hệ thống hỗ trợ thi tập trung,” *Đồ án tốt nghiệp*, 07/2021.
- [2] xuanthuLAB, “<https://j2c.cc/a8fd3746>,” *Gửi email trong PHP*, 2020.
- [3] L. Shenzhen Global Technology Co., LY-03 DC12V Electromagnetic Lock Small Electronic, 2010-2015.
- [4] SYNACORP TRADING & Service, MC-38 Datasheet.
- [5] Rajguru Electronics (I) Pvt.Ltd, AS608 Fingerprint Reader Sensor Module With Cable.
- [6] Rajguru Electronic , DS1307 I2C Real Time Clock Module.
- [7] SHENZHEN EONE ELECTRONICS CO.,LTD, Specification For LCD Module 2004A.
- [8] Handson Technology, RFID RC522.
- [9] Pro Signal, farnell, Buzzer datasheet.
- [10] Ningbo songle relay Co.ltd, SRD SONGLE RELAY Datasheet.
- [11] Toshiba, Transistor Silicon NPN 2SC1815, 1997.
- [12] Espressif, “1. Overview,” *ESP32 Series Datasheet*, p. 8, 2021.
- [13] Ai-Thinker, “ESP32-CAM camera development board,” 2017.
- [14] PGS. TS Trần Thu Hà, Đỗ Văn Thắng, Giáo trình điện tử cơ bản, Đại Học Quốc Gia TP. HCM, 2013.