

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐIỆN ĐIỆN TỬ
BỘ MÔN KỸ THUẬT MÁY TÍNH - VIỄN THÔNG

ĐỒ ÁN MÔN HỌC 1

**XÂY DỰNG HỆ THỐNG CHẤM CÔNG TỰ ĐỘNG
ỨNG DỤNG TRONG QUẢN LÝ NHÂN SỰ**

NGÀNH HỆ THỐNG NHÚNG VÀ IOT

Sinh viên: **PHAN THANH THẢO**

MSSV: 22139062

VÕ MINH THÁI

MSSV: 22139063

Hướng dẫn ThS. **TRƯỜNG QUANG PHÚC**

TP. HỒ CHÍ MINH – 6/2025

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc đến Thầy **Trương Quang Phúc**, người đã tận tình hướng dẫn, định hướng và hỗ trợ chúng em trong suốt quá trình học tập cũng như thực hiện đề tài Đồ án 1. Chính sự tâm huyết và niềm đam mê sâu sắc của Thầy đối với lĩnh vực chuyên môn đã truyền cảm hứng mạnh mẽ, giúp chúng em thêm vững tin và quyết tâm hoàn thành tốt nhiệm vụ được giao.

Chính sự tâm huyết, tận tụy và niềm đam mê sâu sắc của Thầy đối với lĩnh vực kỹ thuật đã truyền cho chúng em nguồn cảm hứng to lớn. Qua từng buổi làm việc và trao đổi, chúng em không chỉ tiếp thu được kiến thức chuyên môn mà còn học hỏi được tinh thần nghiên cứu nghiêm túc và thái độ làm việc chuyên nghiệp – những điều mà chúng em tin rằng sẽ theo chúng em suốt chặng đường học tập và làm việc sau này.

Trong quá trình thực hiện đồ án, chúng em đã cố gắng nỗ lực hết mình, vận dụng những kiến thức đã được học kết hợp với việc tự tìm hiểu tài liệu thực tế nhằm hoàn thành đề tài một cách tốt nhất. Tuy nhiên, do thời gian nghiên cứu có hạn cùng với kinh nghiệm thực hành còn chưa sâu, chúng em hiểu rằng báo cáo này vẫn còn tồn tại một số thiếu sót nhất định. Chúng em rất mong nhận được sự thông cảm và những ý kiến đóng góp quý báu từ Thầy để chúng em có thể tiếp tục hoàn thiện và nâng cao chất lượng của đề tài.

Một lần nữa, chúng em xin gửi lời cảm ơn chân thành đến Thầy. Chúng em kính chúc Thầy luôn dồi dào sức khỏe, hạnh phúc và tiếp tục truyền cảm hứng cho nhiều thế hệ sinh viên tiếp theo trên con đường học tập và nghiên cứu khoa học.

LỜI CAM ĐOAN

Nhóm em xin cam đoan đề tài: “ Xây dựng hệ thống chấm công tự động ứng dụng trong quản lý nhân sự” là do nhóm tự chủ động nghiên cứu, phát triển ý tưởng, thiết kế kỹ thuật và triển khai thực tế từ giao đoạn khởi đầu đến khi hoàn thiện. Toàn bộ quá trình thực hiện không hề có sự sao chép cấu trúc, mã nguồn, hay sử dụng bất kỳ sản phẩm của cá nhân hoặc tổ chức khác dưới mọi hình thức. Đề tài này là kết quả của quá trình tìm tòi học hỏi độc lập, kết hợp với sự hướng dẫn tận tình, góp ý chuyên môn quý báu từ thầy **Trương Quang Phúc**. Tất cả hình ảnh minh họa, sơ đồ thiết kế mạch điện, bố cục phần cứng, và giao diện điều khiển đều được nhóm tự tay xây dựng, kiểm thử và tối ưu hóa thông qua các thử nghiệm thực tế. Về tài liệu tham khảo, nhóm chỉ sử dụng các nguồn sách, bài báo khoa học, tài liệu kỹ thuật chính thống nhằm mục đích hỗ trợ nghiên cứu lý thuyết, hoàn toàn không sao chép nội dung từ báo cáo, đồ án, hay giải pháp thiết kế có sẵn của bất kỳ tác giả nào. Do đó, nhóm hoàn toàn tự chịu trách nhiệm về tính trung thực và nguyên bản của đề tài. Nếu phát hiện bất kỳ sai phạm nào liên quan đến vi phạm quyền sở hữu trí tuệ, sao chép không trích dẫn, hoặc gian lận học thuật trong báo cáo này, nhóm sẵn sàng chấp nhận mọi hình thức kỷ luật từ Bộ môn và Nhà trường theo quy định hiện hành.

Sinh viên cam đoan

Phan Thanh Thảo

Võ Minh Thái

TÓM TẮT

Hiện nay chúng ta đang sống trong thời đại công nghiệp 4.0, các công nghệ kỹ thuật số từ những thập kỷ gần đây đã đẩy lên một cấp độ hoàn toàn mới với sự trợ giúp của kết nối thông qua Internet vạn vật, truy cập và quản lý dữ liệu thời gian thực. Công nghiệp 4.0 kết nối vật lý với kỹ thuật số cho phép cộng tác và truy cập tốt hơn giữa các bộ phận, đối tác, nhà cung cấp, sản phẩm và con người. Nó trao quyền cho các chủ doanh nghiệp kiểm soát và hiểu rõ hơn mọi khía cạnh hoạt động của họ và cho họ tận dụng dữ liệu để tăng năng suất, cải thiện quy trình và thúc đẩy tăng trưởng. Sự ảnh hưởng của nền công nghiệp 4.0 len lỏi đến mọi quy trình trong doanh nghiệp từ nhỏ đến lớn. Các quy trình thủ công nhảm chán đang dần bị thay thế bởi các mô hình tự động mang lại hiệu quả cao và giúp doanh nghiệp không lãng phí nhân sự. Một trong những quy trình được tối ưu hóa để phục vụ cho doanh nghiệp gần đây là quá trình chấm công. Vì vậy nhóm em đề xuất đề tài “Xây dựng hệ thống chấm công tự động ứng dụng trong quản lý nhân sự”. Với đề tài này, nhóm em sẽ khắc phục phương pháp điểm danh thủ công truyền thống thay vào đó đưa ra một hệ thống điểm danh hiện đại điểm danh một cách minh bạch, rõ ràng, chính xác và đưa thông tin người đã điểm danh lên mạng để dễ dàng quản lý và chụp ảnh người đó để hỗ trợ việc xác minh nếu cần. Để thực hiện đề tài này, nhóm sử dụng các vi xử lý như STM32F103C8T6, ESP32, ESP32-CAM để xử lý thông tin và đưa ra kết quả phù hợp theo yêu cầu. Đề tài đã được xây dựng thành công. Các chức năng của thiết bị đều hoạt động ổn, đáp ứng được các yêu cầu đã đề ra.

MỤC LỤC

DANH MỤC HÌNH.....	ix
DANH MỤC BẢNG.....	xii
CÁC TỪ VIẾT TẮT	xii
CHƯƠNG 1. TỔNG QUAN.....	1
1.1. GIỚI THIỆU	1
1.2. TÌNH HÌNH NGHIÊN CỨU	2
1.3. PHƯƠNG PHÁP NGHIÊN CỨU	3
1.4. MỤC TIÊU ĐỀ TÀI	4
1.5. BỐ CỤC ĐỀ TÀI	4
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	6
2.1. TỔNG QUAN VỀ GOOGLE SHEETS VÀ GOOGLE APPS SCRIPT	6
2.1.1. Tổng quan về Google Sheets	6
2.1.2. Ưu điểm của Google Sheets trong hệ thống điểm danh.....	7
2.1.3. Tổng quan về Google Apps Script	7
2.1.4. Ưu điểm điểm của Google Apps Script trong hệ thống điểm danh.....	8
2.2. TỔNG QUAN VỀ CÁC GIAO THỨC TRUYỀN THÔNG.....	9
2.2.1. Giao thức UART	10
2.2.2. Giao thức SPI.....	12
2.2.3. Giao thức I2C	13
2.2.4. Giao thức Wi-Fi (Wireless Fidelity)	16

2.3. TỔNG QUAN VỀ VI ĐIỀU KHIỂN STM32F103C8T6	18
2.3.1. Giới thiệu về dòng STM32 và vi điều khiển STM32F103C8T6.....	18
2.3.2. Cấu tạo của Module STM32 Blue pill	19
2.3.3. Quản lý công suất tiêu thụ trên STM32 Blue pill	21
2.4. TỔNG QUAN VỀ VI ĐIỀU KHIỂN ESP32	23
2.4.1. Cấu tạo của Module ESP32 Devkit	24
2.4.2. Cấu tạo của Module ESP32-CAM	27
2.5. TỔNG QUAN VỀ CÁC THIẾT BỊ DÙNG TRONG HỆ THỐNG	30
2.5.1. Tổng quan về Module vân tay AS608.....	30
2.5.2. Tổng quan về Module RFID RC522.....	34
2.5.3. Tổng quan về màn hình OLED SH1106	38
CHƯƠNG 3. THIẾT KẾ HỆ THỐNG	41
3.1. YÊU CẦU HỆ THỐNG	41
3.2. SƠ ĐỒ KHỐI HỆ THỐNG	42
3.2.1. Sơ đồ khối.....	42
3.2.2. Chức năng từng khối	43
3.2.3. Hoạt động của hệ thống	44
3.3. THIẾT KẾ TÙNG KHỐI	45
3.3.1. Thiết kế khống nhận diện thẻ từ	45
3.3.2. Thiết kế khống sinh trắc học	50
3.3.3. Thiết kế khống hiển thị	53
3.3.4. Thiết kế khống vi điều khiển 1	57

3.3.5. Thiết kế khói vi điều khiển 2	62
3.3.6. Thiết kế khói vi điều khiển 3	65
3.3.7. Thiết kế khói lưu trữ.....	68
3.3.8. Thiết kế khói nhận diện	69
3.3.9. Thiết kế khói mạng.....	70
3.3.10. Thiết kế khói Google Sheets	71
3.3.11. Thiết kế khói giao diện.....	74
3.3.12. Thiết kế khói nguồn	76
3.3.13. Sơ đồ nguyên lý toàn hệ thống	79
3.4. THIẾT KẾ CHƯƠNG TRÌNH CỦA HỆ THỐNG.....	81
3.4.1. Chương trình chính của hệ thống.....	81
3.4.2. Chương trình đăng ký thông tin nhân viên mới	85
3.4.3. Chương trình đăng ký vân tay	87
CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ.....	91
4.1. KẾT QUẢ THIẾT KẾ PHẦN CỨNG.....	91
4.2. KẾT QUẢ THIẾT KẾ PHẦN MỀM.....	96
4.2.1. Kết quả giao tiếp UART trên giao diện quản trị và màn hình OLED	97
4.2.2. Kết quả ghi nhận ảnh và dữ liệu	99
4.2.3. Giao diện quản trị viên	101
4.3. ĐÁNH GIÁ	102
4.3.1. Ưu điểm	103
4.3.2. Hạn chế.....	103

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	105
 5.1. KẾT LUẬN	105
 5.2. Hướng phát triển	106
TÀI LIỆU THAM KHẢO	108

DANH MỤC HÌNH

Hình 2.1. Kiến trúc UART [16]	10
Hình 2.2. Khung dữ liệu của UART [16].....	11
Hình 2.3. Kiến trúc SPI [17]	12
Hình 2.4. Kiến trúc I2C [18]	14
Hình 2.5. Cấu trúc khung dữ liệu I2C [18]	15
Hình 2.6. Mô hình Wifi [19]	16
Hình 2.7. Sơ đồ chân của STM32 Blue pill [23].....	20
Hình 2.8. Sơ đồ chân của Module ESP32-WROOM32 Devkit [24]	25
Hình 2.9. Sơ đồ chân của Module ESP32-CAM [25]	28
Hình 2.10. Module vân tay AS608 [26]	31
Hình 2.11. Sơ đồ chân của Module AS608 [27]	31
Hình 2.12. Module RFID RC522 [28]	35
Hình 2.13. Sơ đồ chân của Module RFID RC522	36
Hình 2.14. Màn hình OLED SH1106 [29].....	38
Hình 3.1. Sơ đồ khối hệ thống.....	42
Hình 3.2. Kết nối STM32 Blue pill với module RFID RC522	47
Hình 3.3. Sơ đồ nguyên lý khối nhận diện thẻ từ.....	48
Hình 3.4. Đường dây trên mạch in	49
Hình 3.5. Vị trí đặt module RFID RC522 trên mạch in	49
Hình 3.6. Kết nối của STM32 Blue pill và module cảm biến vân tay	51
Hình 3.7. Sơ đồ nguyên lý của module vân tay AS608	52
Hình 3.8. Bố trí header cho module vân tay	53
Hình 3.9. Kết nối của màn hình OLED và STM32 Blue pill.....	55
Hình 3.10. Cách đi dây cho I2C trên OLED	56
Hình 3.11. Cách bố trí OLED trên mạch in	56
Hình 3.12. Kết nối của khối vi điều khiển 1 với các thành phần quanh nó	59
Hình 3.13. Sơ đồ nguyên lý của khối vi điều khiển 1	60
Hình 3.14. Vị trí đặt STM32 trên mạch in	62
Hình 3.15. Kết nối của khối vi điều khiển 2 với các thành phần xung quanh	63
Hình 3.16. Sơ đồ nguyên lý của khối vi điều khiển 2	64
Hình 3.17. Cách bố trí khối vi điều khiển 2 trên mạch in	65
Hình 3.18. Kết nối chân của ESP32-CAM với ESP32 Devkit	66
Hình 3.19. Sơ đồ nguyên lý của khối vi điều khiển 3	67
Hình 3.20. Cách bố trí ESP32-CAM trên mạch in.....	68

Hình 3.21. Luồng xử lý dữ liệu truyền từ ESP32-CAM đến Google Sheets	72
Hình 3.22. Các cột được tạo trong Google Sheets	72
Hình 3.23. Đoạn code JavaScript xử lý POST từ ESP32-CAM	73
Hình 3.24. Cách Deploy để ESP32-CAM có thể POST dữ liệu	73
Hình 3.25. Giao diện quản trị viên	75
Hình 3.26. Sơ đồ nguyên lý của khối nguồn	77
Hình 3.27. Sơ đồ nguyên lý toàn mạch	80
Hình 3.28. Lưu đồ thuật toán tổng quát của hệ thống	81
Hình 3.29. Lưu đồ chương trình chính	82
Hình 3.30. Lưu đồ chương trình chính (tiếp theo)	83
Hình 3.31. Lưu đồ thuật toán đăng ký thông tin nhân viên mới	86
Hình 3.32. Lưu đồ chương trình đăng ký vân tay	88
Hình 3.33. Lưu đồ chương trình đăng ký (tiếp theo)	89
Hình 4.1. Mô hình tổng thể của hệ thống	93
Hình 4.2. Mô hình hệ thống sau khi cấp nguồn	93
Hình 4.3. Mô hình của khối vi điều khiển 1 và các thiết bị	94
Hình 4.4. Mô hình của khối vi điều khiển 2 với các thành phần xung quanh	94
Hình 4.5. Mô hình của khối vi điều khiển 3 và các phần tử xung quanh	95
Hình 4.6. Mô hình của khối nguồn	96
Hình 4.7. Kết nối UART với máy tính	97
Hình 4.8. Phản hồi qua UART khi đưa thẻ đúng vào	97
Hình 4.9. Màn hình OLED hiển thị tên nhân viên	98
Hình 4.10. Phản hồi qua UART khi đưa thẻ chưa được đăng ký vào	98
Hình 4.11. Màn hình OLED hiển thị không tìm thấy	98
Hình 4.12. Phản hồi của phần cứng khi nhận được tín hiệu qua UART	99
Hình 4.13. Tên các thư mục ảnh nhân viên sẽ được lưu vào	100
Hình 4.14. Tên của các file ảnh	100
Hình 4.15. Dữ liệu được đẩy lên Google Sheets sau khi điểm danh xong	101
Hình 4.16. Giao diện quản trị hoàn chỉnh và sơ đồ chức năng của nó	102

DANH MỤC BẢNG

Bảng 2.1. Bảng so sánh một số dòng STM32	18
Bảng 2.2. Các tính năng nổi bật của STM32 Blue Pill	21
Bảng 2.3. Các chế độ năng lượng trên STM32F103C8T6.....	22
Bảng 2.4. Các tính năng nổi bật của ESP32 Devkit	26
Bảng 2.5. Các chế độ năng lượng trên ESP32-WROOM-32	27
Bảng 2.6. Các tính năng nổi bật của ESP32-CAM	29
Bảng 2.7. Các chế độ năng lượng của ESP32-CAM.....	30
Bảng 2.8. Thông số kỹ thuật của module AS608	33
Bảng 2.9. Tổng hợp mức tiêu thụ năng lượng phổ biến của module AS608	34
Bảng 2.10. Bảng mô tả chức năng các chân trên module RFID RC522	36
Bảng 2.11. Tổng hợp mức tiêu thụ năng lượng của module RFID RC522.....	37
Bảng 2.12. Thông số kỹ thuật của OLED SH1106	39
Bảng 2.13. Các chế độ năng lượng trên oled SH1106	40
Bảng 3.1. Sơ đồ kết nối STM32 Blue pill và module RFID RC522.....	47
Bảng 3.2. Sơ đồ kết nối của STM32 Blue pill và module cảm biến vân tay	52
Bảng 3.3. Sơ đồ kết nối của OLED và STM32 Blue pill	55
Bảng 3.4. Sơ đồ kết nối của khói vi điều khiển 1 với các thành phần xung quanh.....	60
Bảng 3.5. Sơ đồ kết nối của khói vi điều khiển 2 với các thành phần khác.....	64
Bảng 3.6. Sơ đồ kết nối của ESP32 Devkit và ESP32-CAM.....	67
Bảng 3.7. Công suất tiêu thụ của toàn hệ thống	78

CÁC TỪ VIẾT TẮT

Viết tắt	Ý nghĩa
ADC	Analog-to-Digital Converter (Bộ chuyển đổi tương tự – số)
DAC	Digital-to-Analog Converter (Bộ chuyển đổi số – tương tự)
GND	Ground (Mass nối đất)
GPIO	General Purpose Input/Output (Chân vào/ra mục đích chung)
IDE	Integrated Development Environment (Môi trường phát triển tích hợp)
IEEE	Institute of Electrical and Electronic Engineers (Viện kỹ sư điện và điện tử)
I/O	Input/Output (Nhập/Xuất)
IoT	Internet of Things (Internet vạn vật)
I2C	Inter-Integrated Circuit (Chuẩn giao tiếp liên mạch tích hợp)
MISO	Master Input Slave Output (Chủ nhận – Tớ phát)
MOSI	Master Output Slave Input (Chủ phát – Tớ nhận)
OLED	Organic Light-Emitting Diodes (Đi-ốt phát quang hữu cơ)
PCB	Printed Circuit Board (Mạch in)
RAM	Random Access Memory

RFID	Radio Frequency Identification (Nhận dạng qua tần số vô tuyến)
RX	Receiver (Ngõ nhận)
SPI	Serial Peripheral Interface (Giao diện ngoại vi nối tiếp)
STM	STMicroelectronics (Tên hãng sản xuất vi điều khiển)
TX	Transmitter (Ngõ phát)
UART	Universal Asynchronous Receiver-Transmitter (Giao tiếp nối tiếp bất đồng bộ toàn cục)
ESP	Espressif Systems Platform
USB	Universal Serial Bus (Chuẩn kết nối nối tiếp toàn cục)
RTC	Real-Time Clock
BLE	Bluetooth Low Energy
CPU	Central Processing Unit
Wi-Fi	Wireless Fidelity
SD	Secure Digital
DPI	Dots Per Inch
AP	Access Point
STA	Station
SS	Slave Select
CS	Chip Select
SCCB	Serial Camera Control Bus

GAS	Google Apps Script
HTTPS	HyperText Transfer Protocol Secure
SWD	Serial Wire Debug
API	Application Programming Interface

CHƯƠNG 1. TỔNG QUAN

1.1. GIỚI THIỆU

Cuộc Cách mạng Công nghiệp 4.0 đánh dấu sự hội tụ giữa các công nghệ tiên tiến như Internet of Things (IoT), Trí tuệ nhân tạo (AI), Dữ liệu lớn (Big Data) và Tự động hóa thông minh, mang lại sự chuyển đổi mạnh mẽ trong cách vận hành và quản trị của các tổ chức hiện đại [1]. Trong môi trường doanh nghiệp, những công nghệ này không chỉ giúp tối ưu hóa chi phí vận hành mà còn tăng cường năng suất lao động, độ chính xác và hiệu quả ra quyết định.

Tuy nhiên, nhiều doanh nghiệp hiện nay vẫn còn duy trì phương thức quản lý nhân sự truyền thống, đặc biệt là trong công đoạn chấm công và kiểm soát thời gian làm việc. Các hình thức như ghi sổ tay, quét thẻ từ, hoặc báo cáo thủ công bằng Excel thường xuyên gặp phải các vấn đề như gian lận giờ làm, thiếu minh bạch, mất dữ liệu, hoặc quá tải nhân sự quản lý [2]. Những hạn chế này dẫn đến việc tính lương, đánh giá hiệu suất và hoạch định nguồn nhân lực không còn chính xác, gây ảnh hưởng trực tiếp đến hiệu quả hoạt động của tổ chức.

Để giải quyết các bất cập trên, xu hướng hiện nay đang hướng đến việc tích hợp hệ thống nhúng (Embedded Systems) với IoT (Internet of Things), cho phép xây dựng giải pháp chấm công thông minh, thời gian thực, và đồng bộ dữ liệu với hệ thống trung tâm [3]. Các hệ thống này thường kết hợp vi điều khiển (như STM32, ESP32) với các module nhận diện vân tay, thẻ RFID hoặc khuôn mặt, và có thể mở rộng giao tiếp qua mạng không dây như Wi-Fi hoặc Bluetooth để đồng bộ dữ liệu với máy chủ hoặc lưu trữ đám mây [4].

Ngoài ra, khả năng phân tích dữ liệu chấm công tự động, cảnh báo sai lệch thời gian, và truy xuất báo cáo nhanh chóng giúp nhà quản lý ra quyết định dựa trên dữ liệu đáng tin

cậy, hỗ trợ xây dựng chính sách lương thưởng minh bạch, nâng cao hiệu suất làm việc và giảm thiểu rủi ro nhân sự [5].

Xuất phát từ những vấn đề thực tiễn đó, đề tài "Xây dựng hệ thống chấm công tự động ứng dụng trong quản lý nhân sự" được thực hiện nhằm hiện thực hóa một hệ thống có tính ứng dụng cao, giúp tổ chức nâng cao hiệu quả trong công tác quản lý nhân sự, hướng đến chuyển đổi số toàn diện và phù hợp với yêu cầu thực tiễn trong kỷ nguyên công nghiệp 4.0.

1.2. TÌNH HÌNH NGHIÊN CỨU

Trong bối cảnh chuyển đổi số toàn cầu, hệ thống chấm công tự động đã và đang trở thành một trong những giải pháp thiết yếu nhằm nâng cao hiệu quả quản lý nhân sự. Ở các nước phát triển, xu hướng tích hợp các công nghệ như nhận diện khuôn mặt, trí tuệ nhân tạo (AI), và Internet of Things (IoT) vào hệ thống chấm công đã được nghiên cứu và ứng dụng rộng rãi nhằm giảm thiểu gian lận, tối ưu hóa dữ liệu và tiết kiệm chi phí vận hành. Theo một nghiên cứu công bố trên ScienceDirect, việc ứng dụng thị giác máy tính vào chấm công giúp cải thiện độ chính xác và tốc độ xử lý dữ liệu trong doanh nghiệp hiện đại [6]. Tương tự, một tổng quan hệ thống từ ResearchGate đã chỉ ra rằng các hệ thống chấm công tự động hiện đại đang dần chuyển sang mô hình điện toán đám mây, đồng thời tích hợp bảo mật dữ liệu cá nhân chặt chẽ để đáp ứng yêu cầu pháp lý và thực tiễn [7].

Tại Việt Nam, mặc dù chưa phổ biến công nghệ tiên tiến như ở các nước phát triển, nhưng các phần mềm chấm công ứng dụng công nghệ IoT và nền tảng SaaS đã bắt đầu xuất hiện và được đón nhận rộng rãi. Nhiều doanh nghiệp sử dụng các giải pháp như Base Checkin, Tanca, hay Dogoo HRM để theo dõi thời gian làm việc của nhân viên thông qua các thiết bị như máy chấm công vân tay, nhận diện khuôn mặt hoặc GPS trên điện thoại di động [8][9][10]. Những phần mềm này cho phép quản lý tập trung và xuất dữ liệu thống kê nhanh chóng, hỗ trợ tính lương chính xác và minh bạch.

Tuy nhiên, có thể thấy rằng hầu hết các hệ thống trong nước vẫn ở mức độ tự động hóa cơ bản, chưa có nhiều sản phẩm tích hợp trí tuệ nhân tạo để học và dự đoán hành vi nhân sự, hay phân tích sâu dữ liệu nhằm xây dựng mô hình đánh giá hiệu quả làm việc. Do đó, nhu cầu phát triển một hệ thống chấm công tự động hiện đại, tích hợp cả phần cứng và phần mềm ứng dụng công nghệ IoT, có khả năng mở rộng và tùy biến cao, là hoàn toàn thực tiễn và có tính ứng dụng cao trong doanh nghiệp hiện nay.

1.3. PHƯƠNG PHÁP NGHIÊN CỨU

Để triển khai thành công hệ thống chấm công tự động phục vụ công tác quản lý nhân sự, đề tài áp dụng một chuỗi các phương pháp nghiên cứu có hệ thống, từ lý thuyết đến thực nghiệm. Trước hết, chúng tôi tiến hành nghiên cứu và tổng hợp tài liệu từ các nguồn học thuật và thực tiễn như IEEE Xplore, ScienceDirect, Base.vn, Tanca.io,... nhằm làm rõ thực trạng và xu hướng phát triển của các hệ thống chấm công hiện đại ứng dụng IoT. Thông qua quá trình khảo sát này, đề tài xác định được các yêu cầu chức năng cơ bản và tiêu chí kỹ thuật cần thiết cho hệ thống.

Tiếp theo, nhóm sử dụng phương pháp phân tích – thiết kế hệ thống để xây dựng kiến trúc tổng thể, bao gồm cả phần cứng và phần mềm. Trên cơ sở đó, các linh kiện như cảm biến vân tay, RFID RC522, vi điều khiển ESP32 và STM32, màn hình OLED SH1106 cùng các giao thức truyền thông như UART, SPI, I2C, Wi-fi được lựa chọn và tích hợp. Quá trình thiết kế được cụ thể hóa bằng các sơ đồ khối phần cứng, lưu đồ thuật toán và sơ đồ kết nối, đảm bảo hệ thống có cấu trúc rõ ràng và dễ triển khai.

Sau giai đoạn thiết kế, đề tài tiến hành lắp ráp mô hình thực nghiệm và viết chương trình điều khiển cho hệ thống. Các bài kiểm thử được thực hiện ở cấp độ thành phần để đảm bảo từng thiết bị hoạt động ổn định, trước khi chuyển sang tích hợp toàn bộ hệ thống. Các lỗi, sai lệch hoặc độ trễ trong quá trình vận hành được ghi nhận, phân tích và điều chỉnh nhằm tối ưu hiệu năng và đảm bảo độ chính xác trong việc ghi nhận dữ liệu chấm công.

Cuối cùng, phương pháp mô hình hóa được áp dụng để biểu diễn luồng dữ liệu và hoạt động logic của hệ thống, qua đó hỗ trợ việc lập trình, bảo trì và nâng cấp sau này. Việc kết hợp chặt chẽ giữa các phương pháp nghiên cứu lý thuyết và thực nghiệm giúp đề tài đạt được sự cân bằng giữa tính học thuật và khả năng ứng dụng thực tế, góp phần đề xuất một giải pháp chấm công tự động hiệu quả, hiện đại và phù hợp với xu thế chuyển đổi số trong công tác quản lý nhân sự hiện nay.

1.4. MỤC TIÊU ĐỀ TÀI

Đề tài hướng đến việc xây dựng một hệ thống chấm công tự động ứng dụng công nghệ IoT kết hợp với các thiết bị cảm biến, nhằm nâng cao hiệu quả trong công tác quản lý nhân sự tại doanh nghiệp. Trước hết, đề tài tập trung vào việc thiết kế và phát triển phần cứng hệ thống, bao gồm các thành phần như cảm biến vân tay, mô-đun RFID, vi điều khiển và màn hình hiển thị, bảo đảm khả năng ghi nhận chính xác thời gian làm việc của từng cá nhân. Tiếp theo, việc tích hợp công nghệ IoT giúp hệ thống có thể đồng bộ dữ liệu theo thời gian thực, tạo điều kiện để nhà quản lý giám sát thông tin chấm công từ xa một cách linh hoạt, nhanh chóng và hiệu quả. Đồng thời, hệ thống được thiết kế với cơ chế bảo mật và xác thực chặt chẽ, góp phần hạn chế tối đa các sai sót hoặc hành vi gian lận trong quá trình ghi nhận dữ liệu, từ đó nâng cao tính minh bạch và độ tin cậy. Cuối cùng, đề tài tiến hành đánh giá thực nghiệm nhằm kiểm chứng hiệu suất, độ ổn định và khả năng mở rộng của hệ thống trong điều kiện môi trường thực tế, làm cơ sở đề xuất các giải pháp cải tiến phù hợp với doanh nghiệp quy mô nhỏ và vừa, đặc biệt trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ.

1.5. BỐ CỤC ĐỀ TÀI

Chương 1 giới thiệu tổng quan về đề tài, sự phát triển của hệ thống chấm công từ truyền thống đến tự động, phương pháp nghiên cứu của đề tài.

Chương 2 giới thiệu về các công nghệ dùng trong hệ thống điểm danh tự động, các giao thức truyền thông sử dụng đến các thiết bị phần cứng và phần mềm dùng cho hệ thống.

Chương 3 trình bày các yêu cầu thiết kế hệ thống, sơ đồ khối nguyên lý, quá trình thiết kế phần cứng và phần mềm cho hệ thống.

Chương 4 nêu kết quả thực hiện được trên hệ thống, các chức năng phần mềm tích hợp đi kèm.

Chương 5 nhận xét những yêu cầu đáp ứng của hệ thống, những hạn chế và hướng phát triển hệ thống trong tương lai.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. TỔNG QUAN VỀ GOOGLE SHEETS VÀ GOOGLE APPS SCRIPT

2.1.1. Tổng quan về Google Sheets

Google Sheets là một ứng dụng bảng tính trực tuyến được phát triển bởi Google, cho phép người dùng tạo, chỉnh sửa và tương tác trên các bảng tính theo thời gian thực thông qua trình duyệt web. Khác với các phần mềm bảng tính truyền thống như Microsoft Excel, Google Sheets hoạt động hoàn toàn trên nền tảng điện toán đám mây, giúp việc lưu trữ, chia sẻ và đồng bộ dữ liệu trở nên linh hoạt và tiện lợi hơn [11].

Về mặt kỹ thuật, Google Sheets hỗ trợ đa dạng các hàm tính toán, công thức logic, biểu đồ, bảng tổng hợp, và cả khả năng liên kết dữ liệu giữa các bảng tính khác nhau hoặc từ nguồn dữ liệu bên ngoài. Một trong những ưu điểm nổi bật của Google Sheets là khả năng tích hợp với Google Workspace (trước đây là G Suite), cho phép người dùng sử dụng chung dữ liệu giữa các dịch vụ như Google Forms, Google Drive, Gmail và đặc biệt là Google Apps Script – nền tảng lập trình tích hợp để tự động hóa và mở rộng chức năng bảng tính [12].

Ngoài ra, hệ thống quyền truy cập và theo dõi lịch sử chỉnh sửa theo từng người dùng giúp Google Sheets trở thành công cụ phù hợp cho các hệ thống quản lý dữ liệu đa người dùng như quản lý chấm công, thống kê thời gian làm việc, báo cáo hiệu suất... trong môi trường doanh nghiệp hoặc tổ chức giáo dục. Với khả năng cập nhật dữ liệu theo thời gian thực và hoạt động ổn định trên hạ tầng web, Google Sheets đang ngày càng được ứng dụng rộng rãi trong các giải pháp IoT và quản trị tự động hóa hiện đại [13].

2.1.2. Ưu điểm của Google Sheets trong hệ thống điểm danh

Google Sheets không chỉ là một công cụ bảng tính trực tuyến, mà còn là một nền tảng dữ liệu mạnh mẽ phục vụ hiệu quả cho các hệ thống điểm danh tự động nhờ vào nhiều ưu điểm nổi bật về mặt kỹ thuật và triển khai.

Thứ nhất, khả năng lưu trữ và đồng bộ dữ liệu theo thời gian thực là một trong những yếu tố then chốt giúp Google Sheets phù hợp với hệ thống điểm danh. Khi thiết bị chấm công gửi dữ liệu thông qua giao thức HTTPS hoặc thông qua Apps Script, dữ liệu sẽ được cập nhật tức thì lên bảng tính, giúp nhà quản lý có thể theo dõi tình hình điểm danh của nhân viên hoặc học sinh mà không cần truy cập vào hệ thống nội bộ [11].

Thứ hai, Google Sheets hỗ trợ truy cập đa người dùng và phân quyền chi tiết, điều này rất quan trọng trong môi trường doanh nghiệp hoặc trường học, nơi nhiều bộ phận cần sử dụng cùng một nguồn dữ liệu nhưng với quyền truy cập khác nhau. Ví dụ, phòng nhân sự có thể chỉnh sửa dữ liệu, trong khi trưởng bộ phận chỉ được quyền xem.

Thứ ba, khả năng tích hợp với Google Apps Script giúp tự động hóa quy trình xử lý dữ liệu như tạo báo cáo theo ngày/tuần/tháng, gửi email cảnh báo nếu nhân viên đi muộn, hoặc tính toán số giờ làm việc. Google Apps Script dựa trên nền tảng JavaScript nên dễ dàng học và mở rộng, tạo điều kiện thuận lợi cho việc tích hợp với các hệ thống IoT hoặc API từ bên thứ ba.

Cuối cùng, Google Sheets hoàn toàn miễn phí và không yêu cầu hạ tầng máy chủ riêng, điều này giúp tiết kiệm chi phí đáng kể cho các doanh nghiệp vừa và nhỏ khi triển khai hệ thống điểm danh tự động mà vẫn đảm bảo hiệu quả, độ tin cậy và khả năng mở rộng.

2.1.3. Tổng quan về Google Apps Script

Google Apps Script (GAS) là một nền tảng lập trình dựa trên JavaScript do Google phát triển, cho phép người dùng tự động hóa, mở rộng và tích hợp các ứng dụng trong hệ

sinh thái Google Workspace như Google Sheets, Google Docs, Gmail, Calendar và các dịch vụ khác một cách dễ dàng. Điểm mạnh của GAS là khả năng thực thi hoàn toàn trên nền tảng đám mây, không yêu cầu cài đặt máy chủ, giúp tiết kiệm thời gian triển khai và bảo trì hệ thống [14].

Được thiết kế đặc biệt để hoạt động với các API nội bộ của Google, Google Apps Script cho phép người phát triển tương tác trực tiếp với dữ liệu và sự kiện trong các ứng dụng như Sheets, thông qua các hàm được xây dựng sẵn như SpreadsheetApp, FormApp, hay MailApp. Nhờ đó, trong hệ thống điểm danh tự động, GAS có thể được sử dụng để tự động ghi dữ liệu khi thiết bị IoT gửi thông tin, tính toán giờ làm, lọc dữ liệu nhân viên và thậm chí xuất báo cáo tổng hợp theo chu kỳ định sẵn mà không cần thao tác thủ công [15].

Bên cạnh khả năng xử lý theo lịch trình (trigger theo thời gian) hoặc sự kiện (trigger khi dữ liệu thay đổi), GAS còn hỗ trợ tạo giao diện người dùng tùy chỉnh qua HTML/CSS hoặc sử dụng Google Forms để thu thập thêm thông tin từ nhân sự [14]. Tất cả những đặc điểm trên khiến Google Apps Script trở thành một công cụ lý tưởng trong các hệ thống quản lý nhân sự quy mô vừa và nhỏ, đặc biệt là khi kết hợp với Google Sheets trong mô hình chấm công tự động.

2.1.4. Ưu điểm điểm danh của Google Apps Script trong hệ thống điểm danh

Trong bối cảnh các hệ thống điểm danh hiện đại yêu cầu khả năng xử lý dữ liệu nhanh chóng, linh hoạt và dễ dàng tích hợp với các nền tảng khác, Google Apps Script (GAS) nổi bật nhờ vào tính năng tự động hóa, triển khai nhanh chóng và khả năng kết nối sâu với hệ sinh thái Google. Một trong những ưu điểm lớn nhất của GAS là khả năng tương tác trực tiếp với Google Sheets, đóng vai trò như một cơ sở dữ liệu nền, nơi lưu trữ toàn bộ dữ liệu điểm danh theo thời gian thực [14].

Thứ nhất, Google Apps Script hỗ trợ lập trình hướng sự kiện, nghĩa là hệ thống có thể tự động kích hoạt khi có sự thay đổi trong dữ liệu (ví dụ: khi thiết bị IoT gửi thông tin

điểm danh lên Google Sheets), từ đó giúp xử lý tức thời mà không cần thao tác thủ công từ người quản trị [15]. Đây là một lợi thế rõ rệt trong môi trường yêu cầu phản ứng nhanh, như hệ thống quản lý nhân sự.

Thứ hai, GAS cho phép tạo và gửi báo cáo tự động qua email, giúp tiết kiệm thời gian tổng hợp dữ liệu và hỗ trợ nhà quản lý ra quyết định dựa trên các chỉ số được cập nhật liên tục. Ngoài ra, nó còn cho phép thiết lập các tác vụ theo lịch trình như cập nhật dữ liệu hàng ngày, sao lưu dữ liệu hoặc gửi cảnh báo khi có lỗi hoặc dữ liệu bất thường [14].

Thứ ba, Google Apps Script miễn phí hoàn toàn và không yêu cầu hạ tầng máy chủ riêng biệt, người dùng chỉ cần tài khoản Google để triển khai và quản lý hệ thống. Điều này rất phù hợp với các doanh nghiệp nhỏ và vừa có ngân sách giới hạn, nhưng vẫn muốn áp dụng các giải pháp công nghệ hiện đại.

Cuối cùng, GAS có thể mở rộng và tích hợp với API bên ngoài nếu cần thiết, cho phép hệ thống điểm danh giao tiếp với các nền tảng quản lý nhân sự khác hoặc đồng bộ dữ liệu lên hệ thống quản trị tập trung, giúp mở rộng tiềm năng ứng dụng trong các môi trường doanh nghiệp có yêu cầu cao hơn [15].

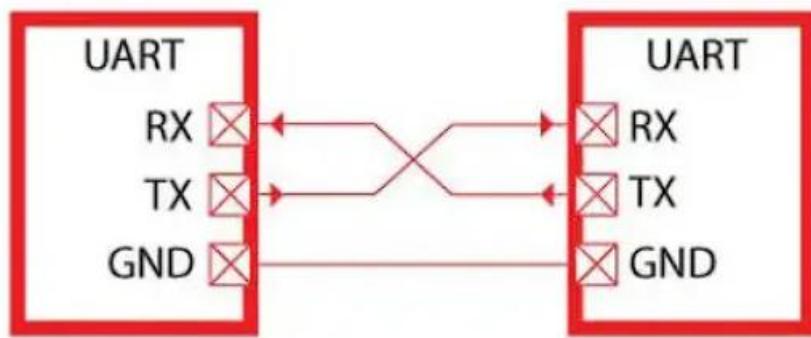
2.2. TỔNG QUAN VỀ CÁC GIAO THỨC TRUYỀN THÔNG

Trong các hệ thống nhúng hiện đại, đặc biệt là hệ thống có nhiều thiết bị ngoại vi như cảm biến, bộ nhớ, module không dây, màn hình,..., việc trao đổi dữ liệu giữa các khối chức năng là điều không thể thiếu. Giao tiếp giữa các thiết bị này thường được thực hiện thông qua các giao thức truyền thông nối tiếp – nơi dữ liệu được gửi từng bit qua một đường dây thay vì gửi song song nhiều bit cùng lúc.

Một số giao thức phổ biến được sử dụng rộng rãi gồm UART, SPI, I2C (giao tiếp nối tiếp nội bộ), và Wi-Fi (truyền thông không dây). Mỗi giao thức có những ưu, nhược điểm riêng và phù hợp với từng loại ứng dụng cụ thể.

2.2.1. Giao thức UART

Trong hầu hết các hệ thống máy tính, từ đơn giản đến phức tạp, giao thức truyền thông nối tiếp UART (Universal Asynchronous Receiver/Transmitter) luôn đóng vai trò then chốt trong việc kết nối và trao đổi dữ liệu giữa các thiết bị phần cứng. UART được xem là một trong những phương thức giao tiếp nền tảng, cho phép các vi điều khiển, module ngoại vi và linh kiện số tương tác với nhau một cách ổn định và hiệu quả thông qua cơ chế truyền dữ liệu nối tiếp không đồng bộ. [16]

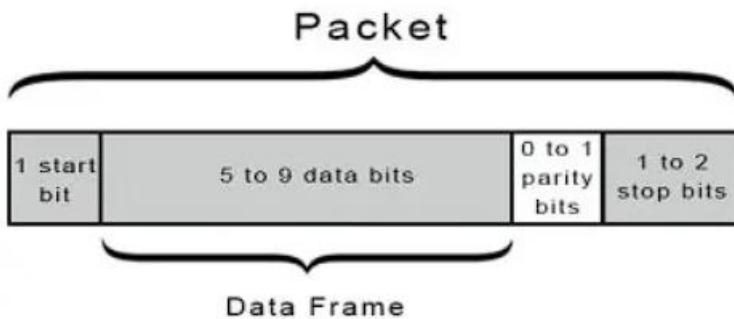


Hình 2.1. Kiến trúc UART [16]

UART được xem là một giao thức truyền thông nối tiếp không đồng bộ, bởi quá trình truyền dữ liệu không đi kèm tín hiệu xung nhịp. Thay vào đó, việc truyền và nhận dữ liệu phụ thuộc vào sự đồng bộ về tốc độ truyền giữa hai thiết bị. Để hiểu rõ hơn về cách thức hoạt động của UART, cần xem xét một số đặc điểm kỹ thuật cơ bản sau:

- Cấu trúc: gồm 2 dây TX (Transmit) – Truyền và RX (Receive) – Nhận
- Truyền song công – hỗ trợ cả truyền và nhận cùng lúc
- Có cấu trúc đơn giản
- Khoảng cách truyền gần
- Chỉ giao tiếp được chỉ hai thiết bị (điểm – điểm)

Dữ liệu trong UART không được truyền theo từng byte đơn lẻ mà được đóng gói thành từng khung truyền (data frame) để đảm bảo thiết bị nhận có thể hiểu chính xác điểm bắt đầu và kết thúc của mỗi đơn vị dữ liệu.



Hình 2.2. Khung dữ liệu của UART [16]

Dựa vào hình trên ta thấy được, một khung dữ liệu UART bao gồm các thành phần sau:

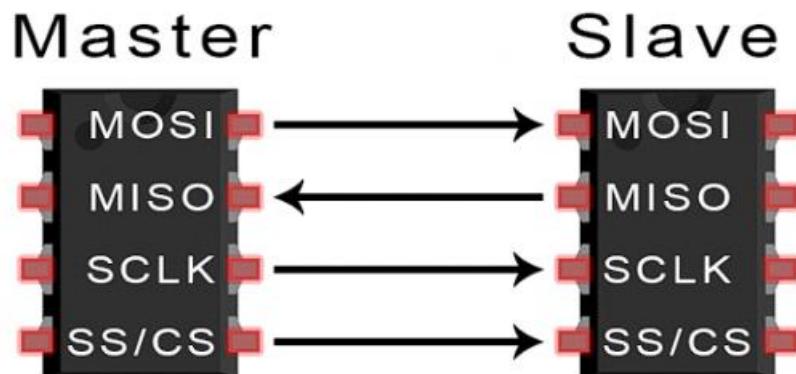
- Start bit (1 bit): bắt đầu mỗi khung truyền là một bit có giá trị logic thấp (0), đánh dấu cho thiết bị biết rằng đây là tín hiệu bắt đầu.
- Data bits (5 đến 9 bit): là phần dữ liệu chính cần truyền, thường là 8 bit (1 byte), nhưng có lúc có dung lượng khác nằm trong giới hạn từ 5 đến 9 bit tùy vào hệ thống.
- Parity bit (có thể tùy chọn là giá trị 0 hoặc 1): dùng để kiểm tra lỗi bằng cách đảm bảo bit 0 hoặc 1 trong khung là chẵn hoặc lẻ. Có thể tắt.
- Stop bit (1 đến 2 bit): là bit logic cao (1) dùng để báo cho thiết bị biết là khung dữ liệu đã kết thúc.

Tóm lại, Tổng độ dài một khung có thể dao động từ 7 đến 13 bit, tùy thuộc vào cấu hình số bit dữ liệu, bit kiểm tra chẵn/lẻ và số bit stop. Thiết lập đúng cấu hình này trên cả thiết bị gửi và nhận là yếu tố quan trọng để đảm bảo dữ liệu được truyền chính xác.

2.2.2. Giao thức SPI

Bên cạnh UART, SPI (Serial Peripheral Interface) là một trong những giao thức truyền thông nối tiếp đồng bộ phổ biến nhất trong các hệ thống nhúng. Được phát triển bởi Motorola, SPI được thiết kế để giao tiếp tốc độ cao giữa vi điều khiển và các thiết bị ngoại vi như cảm biến, bộ nhớ flash, màn hình, hoặc bộ thu phát RF.

SPI hoạt động theo mô hình master–slave, trong đó một thiết bị master (thường là vi điều khiển) điều khiển một hoặc nhiều thiết bị slave thông qua đường xung nhịp chung và các đường dữ liệu riêng biệt.



Hình 2.3. Kiến trúc SPI [17]

Cấu trúc đường truyền của SPI bao gồm 4 dây chính:

- MOSI (Master Out – Slave In): Dữ liệu từ master gửi đến slave.
- MISO (Master In – Slave Out): Dữ liệu từ slave gửi về master.
- SCK (Serial Clock): Xung đồng hồ do master tạo ra để đồng bộ truyền dữ liệu.
- SS/CS (Slave Select / Chip Select): Tín hiệu chọn thiết bị slave cụ thể (logic 0 để kích hoạt).

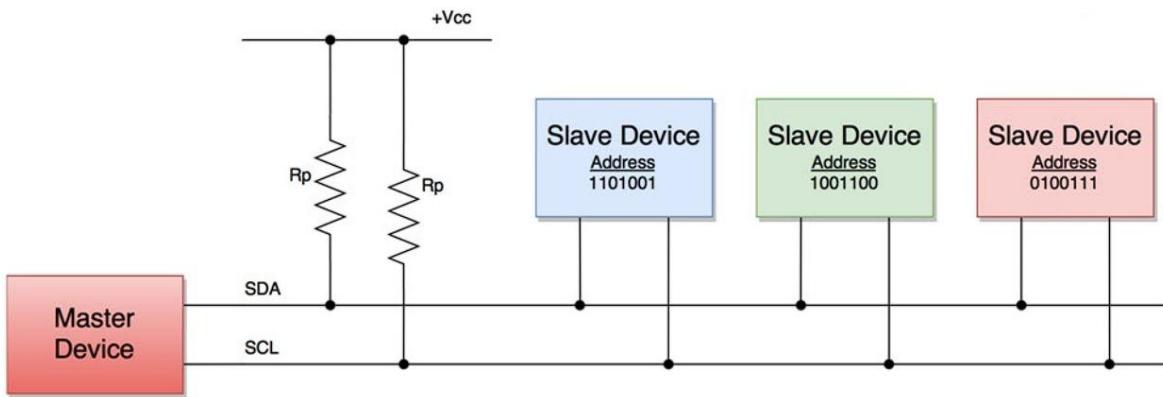
Đặc điểm kỹ thuật chính của SPI:

- Truyền song công: dữ liệu có thể được gửi và nhận đồng thời giữa bên truyền và bên nhận.
- Tốc độ truyền cao: SPI hỗ trợ tốc độ từ vài Mbps đến hàng chục Mbps tùy theo vi điều khiển và thiết bị ngoại vi.
- Đồng bộ hóa bằng xung clock: Dữ liệu được truyền và nhận dựa vào các cạnh lên hoặc xuống của tín hiệu SCK.
- Hỗ trợ nhiều slave: Thông qua các chân SS riêng biệt hoặc giải mã địa chỉ, master có thể giao tiếp với nhiều slave.

Không giống UART, SPI không có khái niệm “khung dữ liệu” cố định với start/stop/parity bit. Thay vào đó, SPI truyền dữ liệu liên tục theo từng byte hoặc word, đồng bộ với xung clock. Mỗi chu kỳ truyền thường gồm 8, 16 hoặc 32 bit, tùy thuộc vào thiết bị được cấu hình.

2.2.3. Giao thức I2C

I2C (đọc là “I-squared-C” hoặc “I-two-C”) là một giao thức truyền thông nối tiếp đồng bộ hai dây, được phát triển bởi hãng Philips Semiconductor vào năm 1982. Không giống như SPI (giao tiếp song công) hay UART (điểm-điểm), I2C được thiết kế để kết nối nhiều thiết bị trên cùng một bus, lý tưởng cho các hệ thống nhúng cần mở rộng với nhiều cảm biến hoặc IC ngoại vi.



Hình 2.4. Kiến trúc I2C [18]

Giao thức I2C chỉ sử dụng hai dây tín hiệu chính:

- SCL (Serial Clock Line): Đường truyền tín hiệu xung nhịp (clock), được tạo ra bởi thiết bị master.
- SDA (Serial Data Line): Đường truyền dữ liệu hai chiều, dùng chung cho tất cả các thiết bị.

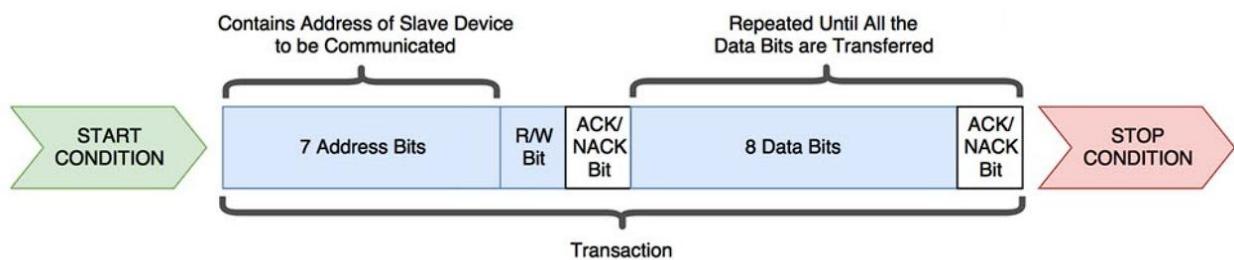
Cả hai đường này đều là dây mở (open-drain), yêu cầu gắn thêm điện trở kéo lên (pull-up resistor) để đảm bảo mức logic ‘1’.

I2C hoạt động theo mô hình master–slave, trong đó một thiết bị master điều khiển bus, khởi tạo giao tiếp và phát xung đồng hồ. Mỗi thiết bị slave được định danh bằng một địa chỉ duy nhất (7 hoặc 10 bit).

Chu trình truyền dữ liệu bao gồm các pha sau:

- Pha 1, Start condition (bắt đầu giao tiếp): master kéo SDA xuống mức thấp trong khi SCL ở mức cao.
- Pha 2, Truyền địa chỉ và bit đọc/ghi (R/W): master gửi địa chỉ slave, kèm theo bit chỉ định đọc hoặc ghi.

- Pha 3, ACK/NACK: slave phản hồi bằng cách kéo SDA xuống (ACK) hoặc giữ ở mức cao (NACK).
- Pha 4, Truyền dữ liệu: dữ liệu được gửi hoặc nhận từng byte (8 bit), sau mỗi byte có bit ACK.
- Pha 5, Stop condition: master kết thúc giao tiếp bằng cách đưa SDA lên mức cao khi SCL đang ở mức cao.



Hình 2.5. Cấu trúc khung dữ liệu I2C [18]

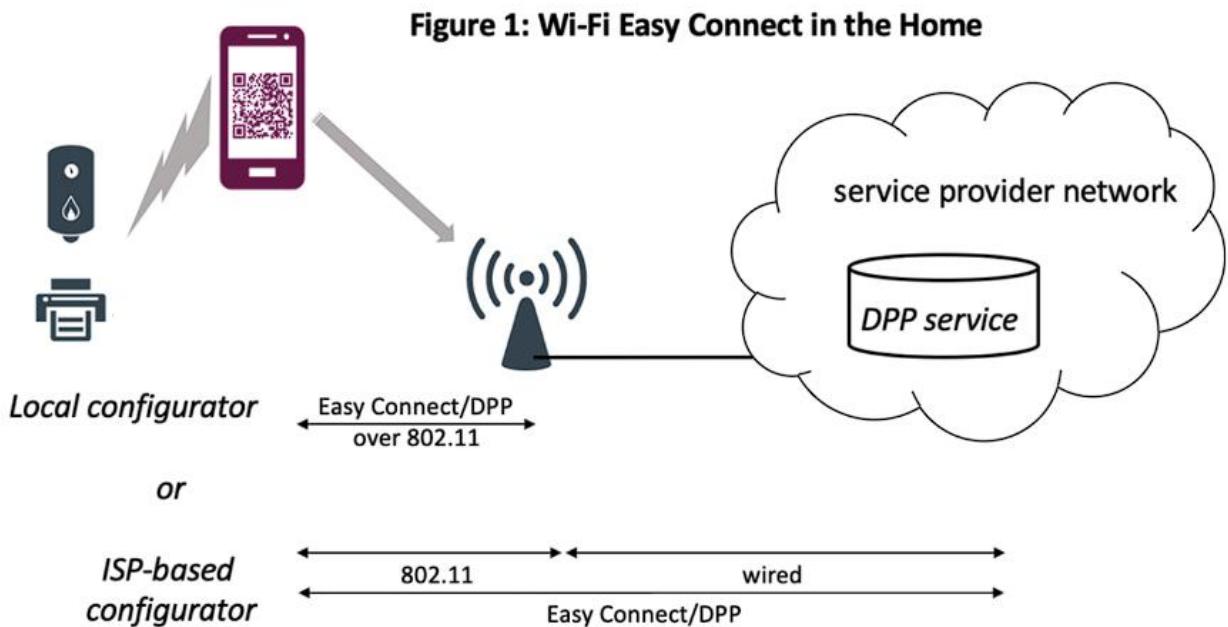
Đặc điểm kỹ thuật của giao thức I2C:

- Tốc độ truyền chuẩn:
 - Standard mode: 100 kbps
 - Fast mode: 400 kbps
 - Fast mode plus: 1 Mbps
 - High-speed mode: lên đến 3.4 Mbps
- Số lượng thiết bị:
 - Lý thuyết hỗ trợ lên đến 127 thiết bị slave (7-bit address)
- Khoảng cách truyền:

- Thích hợp cho kết nối trong bo mạch (< 1 mét), không phù hợp truyền xa.

2.2.4. Giao thức Wi-Fi (Wireless Fidelity)

Wi-Fi là một giao thức truyền thông không dây, hoạt động dựa trên các tiêu chuẩn thuộc họ IEEE 802.11, cho phép các thiết bị trao đổi dữ liệu qua sóng vô tuyến trong các mạng nội bộ (LAN) hoặc kết nối Internet. Khác với các giao thức truyền thông nối tiếp có dây như UART, SPI hay I2C – vốn bị giới hạn về khoảng cách và số lượng thiết bị – Wi-Fi hỗ trợ truyền dữ liệu ở tốc độ cao, phạm vi rộng và quy mô lớn hơn, rất phù hợp với các ứng dụng IoT, điều khiển từ xa, thu thập và đồng bộ dữ liệu lên đám mây.



Hình 2.6. Mô hình Wifi [19]

Wi-Fi sử dụng công nghệ truyền sóng vô tuyến (RF) ở các dải tần 2.4 GHz và 5 GHz, dựa trên các chuẩn phổ biến như:

- 802.11b/g/n: sử dụng dải tần 2.4 GHz (phổ biến trong các thiết bị IoT như ESP32)
- 802.11ac/ax: hoạt động ở dải tần 5 GHz (hiệu suất cao hơn)

Mỗi thiết bị Wi-Fi được định danh bởi một địa chỉ MAC duy nhất, và có thể hoạt động theo các chế độ:

- Station (STA): thiết bị kết nối vào mạng Wi-Fi như một client
- Access Point (AP): thiết bị tạo mạng Wi-Fi cho các thiết bị khác kết nối
- Soft-AP + STA: hỗn hợp giữa AP và client (ESP32 hỗ trợ chế độ này)

Đặc điểm kỹ thuật:

- Chuẩn hỗ trợ: IEEE 802.11 b/g/n (phổ biến trên vi điều khiển như ESP32)
- Tần số hoạt động: 2.4 GHz và/hoặc 5 GHz
- Tốc độ truyền: từ vài Mbps đến hơn 100 Mbps tùy chuẩn
- Phạm vi truyền hiệu quả: từ 10 đến 50 mét (tùy điều kiện môi trường)
- Giao thức truyền: TCP/IP, UDP, HTTP, MQTT,...
- Cơ chế kết nối: chế độ Station (STA), Access Point (AP) hoặc kết hợp Soft-AP + STA
- Bảo mật hỗ trợ: WPA, WPA2, WPA3

Ưu điểm của Wi-Fi so với các giao thức khác:

- Không cần dây kết nối vật lý → dễ lắp đặt và triển khai linh hoạt
- Hỗ trợ kết nối Internet và truyền dữ liệu từ xa qua mạng LAN/WAN
- Tốc độ truyền cao hơn UART, I2C và SPI
- Có thể giao tiếp với nhiều thiết bị thông qua mạng Wi-Fi hoặc nền tảng IoT
- Hỗ trợ các giao thức mạng tiêu chuẩn (HTTP, MQTT, WebSocket...)

Nhược điểm:

- Tiêu thụ điện năng cao hơn các giao tiếp có dây
- Độ trễ có thể lớn, không phù hợp cho truyền dữ liệu thời gian thực ngắn

- Dễ nhanh trong môi trường có nhiều thiết bị không dây
- Cần cấu hình mạng (SSID, mật khẩu, địa chỉ IP)
- Phụ thuộc vào tín hiệu Wi-Fi và chất lượng router

2.3. TỔNG QUAN VỀ VI ĐIỀU KHIỂN STM32F103C8T6

2.3.1. Giới thiệu về dòng STM32 và vi điều khiển STM32F103C8T6

STM32 là dòng vi điều khiển hiệu suất cao do hãng STMicroelectronics phát triển, sử dụng kiến trúc ARM Cortex-M, được thiết kế nhằm đáp ứng nhu cầu của các hệ thống nhúng trong nhiều lĩnh vực như công nghiệp, thiết bị y tế, IoT và tự động hóa. STM32 được phân loại thành nhiều dòng khác nhau như STM32F0 (giá rẻ, tiết kiệm năng lượng), STM32F1 (cân bằng giữa hiệu suất và chi phí), STM32F4 (hiệu suất cao), và các dòng cao cấp hơn như STM32H7 v.v[20]. Để thấy được sự khác biệt trên ta tiến hành so sánh:

Bảng 2.1. Bảng so sánh một số dòng STM32

Tiêu chí	STM32F0	STM32F1	STM32F4	STM32H7
Lõi xử lý	Cortex-M0	Cortex-M3	Cortex-M4	Cortex-M7
Xung nhịp tối đa	48 MHz	72 MHz	180 MHz	480 MHz
Flash	16 – 128 KB	16 – 128 KB	512 KB – 2 MB	1 – 2 MB
RAM	4 – 32 KB	4 – 20 KB	96 – 256 KB	192 KB – 1 MB
Giá thành	Rất thấp	Thấp	Trung bình	Cao

Qua bảng so sánh trên, ta thấy dòng STM32F1, đặc biệt là model STM32F103C8T6, được xem là lựa chọn phổ biến nhờ tính ổn định, giá thành hợp lý và hỗ trợ phần mềm mạnh mẽ.

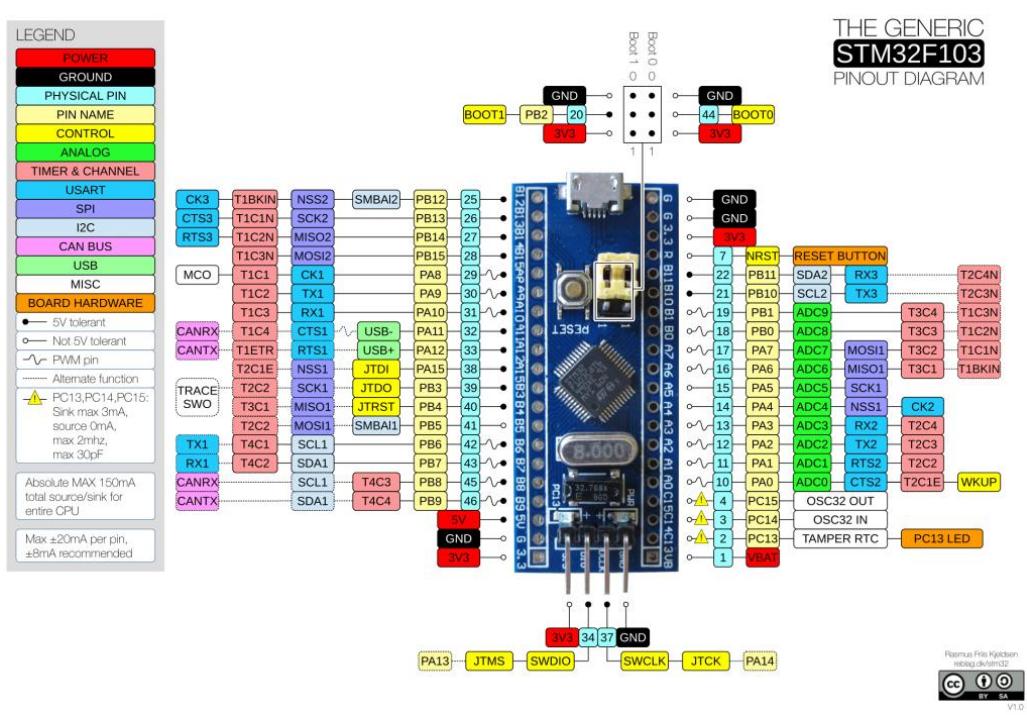
STM32F103C8T6 thuộc nhóm STM32F103x8 trong họ STM32F1, sử dụng nhân ARM Cortex-M3, tốc độ hoạt động tối đa 72 MHz, tích hợp bộ nhớ Flash 64 KB và RAM 20 KB. Vi điều khiển này cung cấp nhiều chuẩn giao tiếp ngoại vi như USART, SPI, I2C,

ADC, PWM, rất phù hợp cho các ứng dụng nhúng yêu cầu kết nối đa thiết bị như cảm biến, màn hình, module không dây hay các hệ thống thu thập và xử lý dữ liệu. Chính vì vậy, STM32F103C8T6 được đánh giá là giải pháp lý tưởng cho các đề tài nghiên cứu và phát triển hệ thống tự động, đặc biệt trong các dự án IoT và chấm công tự động [21].

Không chỉ nổi bật bởi cấu hình phần cứng hợp lý, STM32F103C8T6 còn nhận được sự hỗ trợ mạnh mẽ từ cộng đồng phát triển thông qua các công cụ như STM32CubeIDE, STM32CubeMX, và thư viện lập trình HAL/LL, giúp đơn giản hóa quá trình cấu hình và lập trình. Việc kết hợp các công cụ phần mềm này với phần cứng hiệu quả tạo nên một hệ sinh thái phát triển lý tưởng, hỗ trợ sinh viên, kỹ sư và nhà phát triển triển khai giải pháp nhanh chóng và chính xác [22].

2.3.2. Cấu tạo của Module STM32 Blue pill

Module STM32 Blue pill là một bo mạch phát triển giá rẻ, nhỏ gọn, rất phù hợp để từng bước tiếp cận với STM32, module sử dụng vi điều khiển STM32F103C8T6 làm bộ xử lý trung tâm. Blue pill được thiết kế với đầy đủ các thành phần ngoại vi cơ bản, giúp người dùng dễ dàng tiếp cận và triển khai các ứng dụng hệ thống nhúng trong thực tế.



Hình 2.7. Sơ đồ chân của STM32 Blue pill [23]

Về mặt cấu tạo, module gồm các thành phần chính sau:

- Vi điều khiển STM32F103C8T6: Là chip ARM Cortex-M3 32-bit, tốc độ lên đến 72 MHz, tích hợp 64 KB Flash và 20 KB SRAM. Đây là nhân xử lý chính của module, thực hiện mọi thao tác điều khiển, xử lý tín hiệu và giao tiếp ngoại vi.
- Bộ chân I/O (GPIO): Cung cấp khoảng 37 chân I/O có thể cấu hình, hỗ trợ các chức năng như digital input/output, PWM, ADC, giao tiếp UART/SPI/I2C, giúp module có khả năng kết nối linh hoạt với nhiều loại cảm biến và thiết bị ngoại vi.
- Cổng nạp và debug: Hỗ trợ nạp chương trình qua cổng UART (qua USB-to-Serial như CP2102 hoặc CH340), đồng thời tích hợp header SWD (Serial Wire Debug) giúp debug chương trình với các công cụ như ST-Link.

- Bộ dao động (Oscillator): Module tích hợp hai thạch anh – một thạch anh 8 MHz dùng cho hệ thống chính và một thạch anh 32.768 kHz dành cho bộ RTC (Real-Time Clock), đảm bảo độ chính xác về thời gian và ổn định cho hoạt động của hệ thống.
- Nguồn cấp và ổn áp: Có thể cấp nguồn từ 5V (qua USB hoặc chân VCC) hoặc 3.3V trực tiếp. Trên mạch có tích hợp IC ổn áp AMS1117-3.3 để chuyển đổi từ 5V xuống 3.3V – mức điện áp hoạt động tiêu chuẩn của STM32F103C8T6.
- LED và nút nhấn: Gồm một LED trạng thái (thường nối với chân PC13) và hai nút nhấn – một nút RESET để khởi động lại hệ thống, và một nút BOOT dùng để chọn chế độ khởi động (từ Flash hoặc từ UART để nạp chương trình).

Bảng 2.2. Các tính năng nổi bật của STM32 Blue Pill

Tính năng	Chi tiết
Bộ xử lý	ARM Cortex-M3
Tần số hoạt động	Tối đa 72 MHz
Bộ nhớ	64 KB Flash và 20 KB RAM
Các giao tiếp	UART, SPI, I2C, USB, CAN
Số chân GPIO	37 chân
Số chân ADC	10 chân

2.3.3. Quản lý công suất tiêu thụ trên STM32 Blue pill

Trong các hệ thống nhúng hiện đại, đặc biệt là những ứng dụng sử dụng nguồn pin hoặc nguồn điện hạn chế, việc quản lý và tối ưu công suất tiêu thụ của vi điều khiển đóng vai trò cực kỳ quan trọng. Nếu không kiểm soát tốt mức tiêu thụ điện năng, hệ thống có thể gặp phải các vấn đề như tuổi thọ pin giảm nhanh, phát sinh nhiệt độ cao làm ảnh hưởng đến độ bền của linh kiện, cũng như gây tốn kém chi phí vận hành trong dài hạn. Do đó, tiết kiệm điện không chỉ giúp kéo dài thời gian hoạt động của thiết bị mà còn nâng cao hiệu quả tổng thể của hệ thống. Với các vi điều khiển như STM32 Blue Pill, việc áp dụng các chế độ quản

lý năng lượng hợp lý và sử dụng các tính năng tiết kiệm điện một cách thông minh trở thành yếu tố then chốt giúp cân bằng giữa hiệu năng xử lý và mức tiêu thụ điện, đáp ứng yêu cầu khắc khe của các ứng dụng nhúng hiện đại.

STM32 Blue Pill, dựa trên vi điều khiển STM32F103C8T6, cung cấp nhiều chế độ hoạt động giúp tối ưu hóa công suất tiêu thụ, từ đó kéo dài tuổi thọ pin và giảm thiểu nhiệt lượng phát sinh trong các ứng dụng nhúng. Quản lý công suất tiêu thụ là một yếu tố quan trọng nhằm đảm bảo hiệu suất hệ thống trong khi vẫn duy trì mức tiêu thụ năng lượng ở mức thấp nhất có thể.

Vi điều khiển STM32F103C8T6 tích hợp các chế độ năng lượng thấp (Low Power Modes) như Sleep mode, Stop mode và Standby mode, cho phép thiết bị giảm tần số hoạt động hoặc tắt hoàn toàn các module không cần thiết khi không sử dụng.

Bảng 2.3. Các chế độ năng lượng trên STM32F103C8T6

Chế độ	CPU	Ngoại vi	Dòng tiêu thụ	Đánh thức
Run Mode	Chạy	Hoạt động	20 – 30 mA	Không cần
Sleep Mode	Ngừng chạy	Hoạt động	Khoảng 2-3mA	Ngắt ngoài
Stop Mode	Ngừng chạy	RTC và một số ngoại vi hoạt động	Khoảng 5-10µA	Ngắt ngoài, RTC
Standby Mode	Tắt	Hầu hết bị tắt	Khoảng 2-3µA	Reset hoặc ngắt ngoài đặc biệt

Ngoài ra, STM32 Blue Pill còn hỗ trợ điều chỉnh điện áp hoạt động và tần số xung nhịp thông qua các cơ chế điều khiển nội tại, cho phép cân bằng giữa hiệu năng xử lý và mức tiêu thụ điện năng theo nhu cầu ứng dụng. Việc sử dụng các bộ nguồn cấp hợp lý, kết

hợp với khả năng quản lý năng lượng thông minh giúp Blue Pill phù hợp với các hệ thống nhúng chạy pin hoặc cần tiết kiệm năng lượng tối ưu.

Tổng hợp các tính năng quản lý công suất này tạo nên một nền tảng hiệu quả cho các ứng dụng IoT và hệ thống nhúng, đặc biệt là các dự án yêu cầu hoạt động liên tục với nguồn năng lượng hạn chế, chẳng hạn như hệ thống chấm công tự động và các thiết bị cảm biến thông minh.

2.4. TỔNG QUAN VỀ VI ĐIỀU KHIỂN ESP32

ESP32 là dòng vi điều khiển tích hợp hiệu năng cao do hãng Espressif Systems (Trung Quốc) phát triển, được thiết kế dành riêng cho các ứng dụng nhúng và Internet vạn vật (IoT). Điểm nổi bật của ESP32 nằm ở việc tích hợp hai lõi xử lý Xtensa 32-bit LX6 có thể hoạt động độc lập với xung nhịp lên đến 240 MHz, đi kèm bộ nhớ RAM trong lên đến 520 KB và hỗ trợ Flash ngoài dung lượng lớn. Bên cạnh khả năng xử lý mạnh mẽ, ESP32 còn tích hợp sẵn kết nối không dây Wi-Fi chuẩn 802.11 b/g/n và Bluetooth v4.2 (bao gồm cả BR/EDR và BLE), giúp đơn giản hóa thiết kế hệ thống, giảm chi phí và tối ưu hóa hiệu suất truyền thông trong các mạng thiết bị IoT.

Trong hệ thống chấm công tự động được xây dựng trong đề tài này, nhóm đã khai thác sức mạnh và tính linh hoạt của hai dòng module dựa trên ESP32 là ESP32-WROOM-32 và ESP32-CAM, mỗi module đảm nhận một vai trò riêng biệt trong tổng thể kiến trúc hệ thống.

- ESP32-WROOM-32 đóng vai trò như một trung tâm điều phối và truyền thông, có nhiệm vụ tiếp nhận dữ liệu từ các vi điều khiển STM32 (nơi xử lý tín hiệu từ cảm biến vân tay và RFID), sau đó trao đổi và đồng bộ thông tin với ESP32-CAM thông qua giao tiếp UART.
- Trong khi đó, ESP32-CAM là một biến thể đặc biệt của ESP32, được tích hợp sẵn camera OV2640 và khe cắm thẻ nhớ microSD, phù hợp với các ứng dụng thị giác

máy tính đơn giản. Trong hệ thống, module này đảm nhận nhiệm vụ chụp ảnh khuôn mặt nhân sự tại thời điểm chấm công, lưu trữ ảnh cục bộ và đồng thời đẩy dữ liệu hình ảnh lên nền tảng đám mây, phục vụ cho việc quản lý minh bạch và đồng bộ thông tin nhân sự.

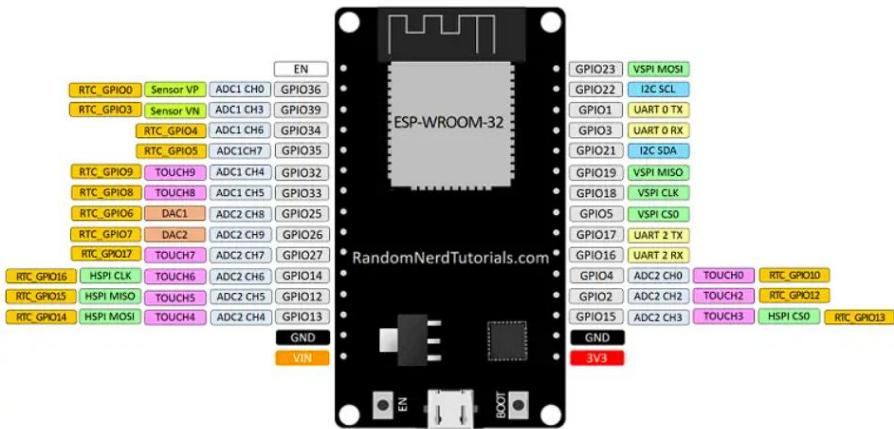
Việc sử dụng kết hợp hai module ESP32 không chỉ giúp tách biệt rõ ràng chức năng giữa thu thập và xử lý ảnh, mà còn tận dụng tối ưu khả năng xử lý song song và kết nối không dây ổn định của nền tảng ESP32, qua đó đảm bảo hệ thống hoạt động hiệu quả, linh hoạt và dễ mở rộng trong tương lai.

2.4.1. Cấu tạo của Module ESP32 Devkit

Module ESP32-WROOM-32 DevKit là một bo mạch phát triển được tích hợp sẵn module vi điều khiển ESP32-WROOM-32, được thiết kế với mục tiêu hỗ trợ nhanh chóng việc lập trình, kiểm thử và triển khai các ứng dụng IoT. Nhờ tính linh hoạt, hiệu năng cao và chi phí thấp, module này được sử dụng rộng rãi trong cả nghiên cứu lẫn sản phẩm thương mại.

ESP32 DEVKIT V1 – DOIT

version with 30 GPIOs



Hình 2.8. Sơ đồ chân của Module ESP32-WROOM32 Devkit [24]

Cấu tạo của bo mạch ESP32-WROOM-32 DevKit bao gồm các thành phần phần cứng cơ bản như sau:

- Module ESP32-WROOM-32: Đây là thành phần cốt lõi, đóng vai trò như bộ xử lý trung tâm của hệ thống. Bên trong tích hợp vi điều khiển Xtensa LX6 dual-core 32-bit có thể hoạt động ở xung nhịp lên đến 240 MHz. Ngoài ra, module còn tích hợp bộ thu phát Wi-Fi (chuẩn 802.11 b/g/n) và Bluetooth (v4.2 BR/EDR và BLE), RAM 520 KB và bộ nhớ Flash SPI 4 MB (hoặc lớn hơn tùy phiên bản).
- Chip chuyển đổi USB-to-Serial (CP2102 hoặc CH340): Thành phần này cho phép giao tiếp giữa ESP32 và máy tính thông qua cổng USB, hỗ trợ quá trình nạp chương trình và truyền dữ liệu khi phát triển phần mềm.
- Cổng USB (Micro-B hoặc USB-C): Được sử dụng để cấp nguồn 5V cho bo mạch và cũng là giao diện kết nối để nạp chương trình. Tùy phiên bản DevKit, có thể sử dụng cổng Micro USB hoặc USB Type-C.
- Mạch nguồn: Thường sử dụng IC AMS1117 hoặc tương đương để hạ áp từ 5V xuống 3.3V, đáp ứng điện áp làm việc chuẩn của ESP32.

- Hai nút nhấn BOOT và RESET (EN):
 - Nút BOOT được sử dụng để đưa thiết bị vào chế độ nạp chương trình thủ công.
 - Nút RESET (hoặc EN) dùng để khởi động lại hệ thống, giúp kiểm thử hoặc reset thiết bị khi đang hoạt động.
- Hệ thống điện trở kéo (pull-up/pull-down): Được bố trí trên các chân đặc biệt như IO0, EN để đảm bảo khởi động đúng chế độ (flash hay run) mỗi khi reset.
- Hàng chân GPIO chuẩn 2.54mm: Các chân GPIO được đưa ra ngoài qua hai hàng chân header giúp kết nối dễ dàng với các thiết bị ngoại vi như cảm biến, relay, màn hình hiển thị,...
- Các tụ lọc và linh kiện chống nhiễu: Được bố trí xung quanh khu vực cấp nguồn và các chân tín hiệu nhằm đảm bảo tính ổn định cho toàn mạch, đặc biệt trong môi trường nhiễu điện từ.

Bảng 2.4. Các tính năng nổi bật của ESP32 Devkit

Tính năng	Chi tiết
Bộ xử lý	Xtensa LX6 dual-core 32-bit
Tần số hoạt động	Lên đến 240 MHz
Bộ nhớ	520 KB SRAM và 4 MB Flash SPI
Các giao tiếp	UART, SPI, I2C, Wi-Fi, Bluetooth
Số chân GPIO	30
Số chân ADC	15

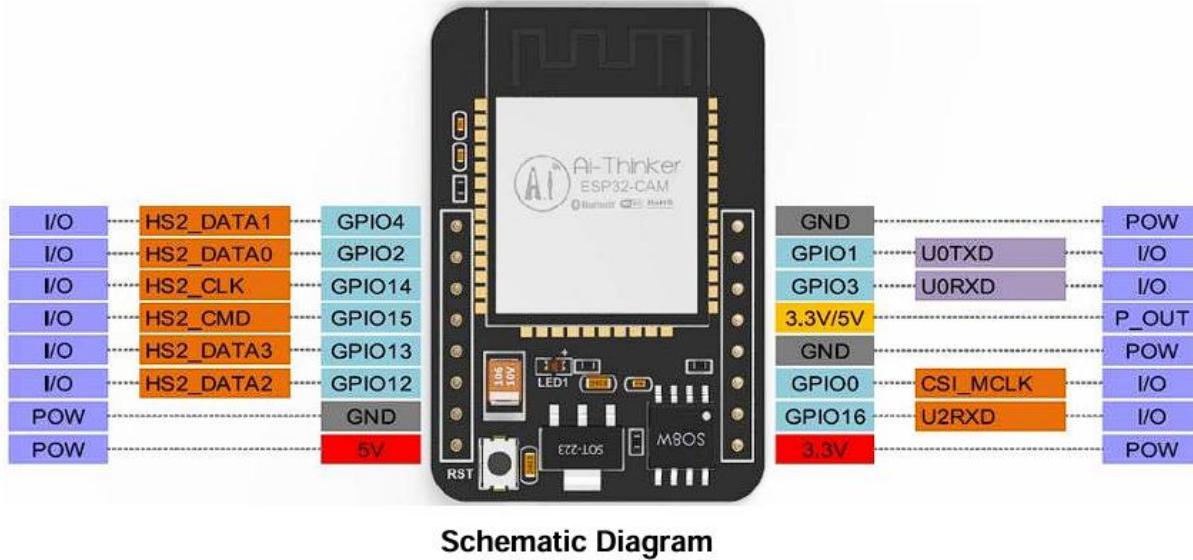
ESP32 được thiết kế để hoạt động hiệu quả cả trong môi trường tiêu tốn năng lượng thấp. Tùy thuộc vào nhu cầu ứng dụng, ESP32 có thể chuyển đổi linh hoạt giữa các chế độ tiết kiệm năng lượng như sau:

Bảng 2.5. Các chế độ năng lượng trên ESP32-WROOM-32

Chế độ	CPU	Ngoại vi	Dòng tiêu thụ	Đánh thức
Active Mode	Hoạt động	Hoạt động	~160–260 mA	Không
Modem Mode	Hoạt động	Wi-fi tắt	~3–20 mA	Tự động khi truyền dữ liệu
Light Mode	Dừng tạm thời	Chỉ RTC hoạt động	~0.8–2 mA	Timer, GPIO, UART, Touch
Deep Sleep	Dừng hoàn toàn	Chỉ RTC hoạt động	~10–150 µA	Timer, GPIO, Touch, ULP
Hibernation	Tắt hoàn toàn	Ngoại vi tắt	~2.5 µA	GPIO RTC hoặc Reset

2.4.2. Cấu tạo của Module ESP32-CAM

ESP32-CAM là một module tích hợp vi điều khiển ESP32 với khả năng xử lý mạnh mẽ, hỗ trợ kết nối Wi-Fi, cùng với camera và khe cắm thẻ nhớ SD. Nhờ sự kết hợp giữa tính năng điều khiển, thu nhận hình ảnh và truyền dữ liệu, ESP32-CAM được ứng dụng rộng rãi trong các hệ thống nhận diện khuôn mặt, giám sát hình ảnh và các ứng dụng IoT cần camera chi phí thấp.



Schematic Diagram

Hình 2.9. Sơ đồ chân của Module ESP32-CAM [25]

Các thành phần chính có trong module:

- Vi điều khiển chính (ESP32-S): Là biến thể của dòng ESP32, tích hợp lõi xử lý XtensaLX6 dual-core, hỗ trợ Wi-Fi và Bluetooth, đóng vai trò trung tâm điều khiển toàn bộ module.
- Camera (OV2640): Cảm biến CMOS 2MP, hỗ trợ độ phân giải lên đến 1600×1200 , có thể điều chỉnh khung hình và chất lượng ảnh. Được kết nối với ESP32 qua giao tiếp SCCB.
- Khe cắm thẻ nhớ microSD: Cho phép lưu trữ ảnh hoặc dữ liệu trực tiếp lên thẻ nhớ thông qua giao tiếp SPI.
- Ăng-ten PCB / Ăng-ten ngoài: Hỗ trợ kết nối không dây, có thể chuyển đổi giữa anten tích hợp và anten ngoài thông qua jumper.
- Bộ ổn áp (AMS1117 – 3.3V): Hạ áp từ nguồn 5V xuống 3.3V để cung cấp cho ESP32 và các linh kiện khác.

- Mạch nguồn và lọc nhiễu: Đảm bảo module hoạt động ổn định với dòng tiêu thụ thấp.
- Chân I/O và UART: Cung cấp các chân giao tiếp cơ bản, gồm UART, SPI, I2C, PWM, GPIO (tuy nhiên số chân bị giới hạn do chia sẻ với camera và thẻ SD).

Bảng 2.6. Các tính năng nổi bật của ESP32-CAM

Tính năng	Chi tiết
Bộ xử lý	ESP32-S, dual-core Xtensa LX6
Tần số hoạt động	Lên đến 240 MHz
Bộ nhớ	520 KB SRAM, 4 MB Flash
Các giao tiếp	UART, SPI, I2C, SCCB
Số chân GPIO	~9 chân khả dụng (nhiều chân chia sẻ với camera)
Số chân ADC	Rất hạn chế (do xung đột chân với camera/SD), thường chỉ 1–2 chân dùng được

ESP32-CAM hỗ trợ nhiều chế độ tiêu thụ điện năng tương tự các dòng ESP32 khác. Tuy nhiên, do tích hợp camera và thẻ nhớ microSD, mức tiêu thụ trong chế độ hoạt động có thể cao hơn đáng kể. Việc tối ưu năng lượng là thật sự cần thiết, vì vậy nắm rõ các chế độ năng lượng của module sẽ góp phần giúp hệ thống hoạt động ổn định và tránh những lỗi không mong muốn.

Bảng 2.7. Các chế độ năng lượng của ESP32-CAM

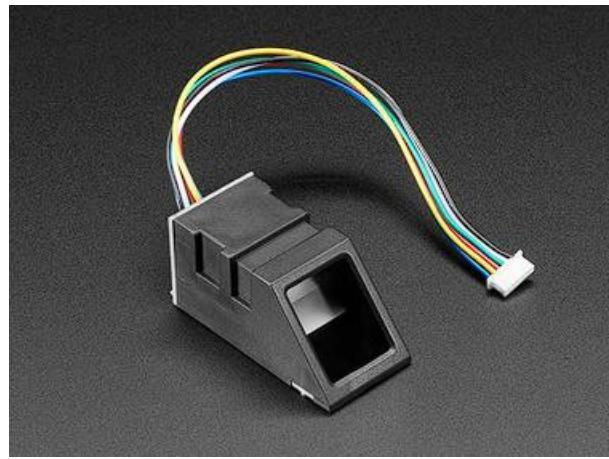
Chế độ	CPU	Ngoại vi	Dòng tiêu thụ	Đánh thức
Active	Hoạt động	Camera + Wi-Fi + thẻ SD hoạt động	~160–310 mA	Không cần
Deep Sleep	Dừng hoàn toàn	RTC giữ lại, camera + SD + Wi-Fi tắt	~6–10 µA	Timer, GPIO, ULP, Touch
Hibernation	Tắt	Tất cả ngoại vi tắt hoàn toàn	~2–3 µA	GPIO RTC hoặc Reset

2.5. TỔNG QUAN VỀ CÁC THIẾT BỊ DÙNG TRONG HỆ THỐNG

2.5.1. Tổng quan về Module cảm biến vân tay AS608

AS608 là một module cảm biến vân tay quang học tích hợp, được thiết kế để xử lý nhận dạng dấu vân tay với độ chính xác cao và khả năng xử lý độc lập. Module này bao gồm một cảm biến ảnh, bộ xử lý DSP nội bộ và bộ nhớ lưu trữ vân tay, cho phép ghi lại, tìm kiếm, so khớp và xóa dấu vân tay mà không cần bộ vi xử lý bên ngoài xử lý trực tiếp ảnh thô.

AS608 được ứng dụng rộng rãi trong các hệ thống điều khiển truy cập, khóa cửa thông minh và các hệ thống xác thực sinh trắc học nhờ sự cân bằng tốt giữa hiệu suất, chi phí và khả năng tích hợp dễ dàng với các vi điều khiển phổ biến hiện nay.

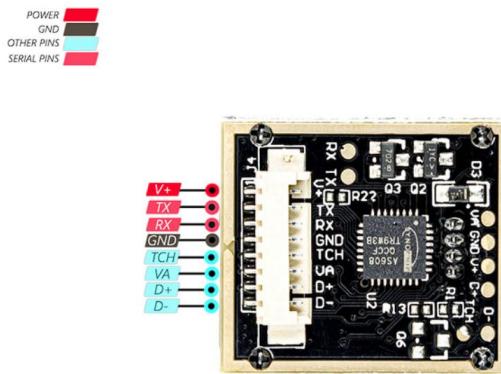


Hình 2.10. Module vân tay AS608 [26]

Module AS608 được tích hợp từ ba khối chính:

- Cảm biến quang học: Dùng để thu ảnh vân tay khi người dùng đặt ngón tay lên bề mặt.
- Bộ xử lý nội bộ (DSP): Thực hiện các thuật toán xử lý ảnh, so khớp vân tay, lưu trữ và tìm kiếm dữ liệu.
- Bộ nhớ Flash tích hợp: Cho phép lưu trữ từ 100 đến 162 mẫu vân tay tùy phiên bản.

Ngoài ra, module còn có lớp vỏ bảo vệ, mạch điều chỉnh điện áp và cổng giao tiếp UART.



Hình 2.11. Sơ đồ chân của Module AS608 [27]

Module cảm biến vân tay AS608 được thiết kế với giao tiếp UART là giao thức truyền thông chính. Đây là dạng truyền nối tiếp tốc độ thấp, đơn giản, phổ biến trên hầu hết các dòng vi điều khiển như STM32, Arduino, hay ESP32. Trên thực tế, AS608 không hỗ trợ các giao tiếp như SPI hay I2C, bởi toàn bộ giao tiếp được tối ưu cho UART với giao thức độc quyền của nhà sản xuất.

- Giao tiếp sử dụng:
 - UART (TX, RX): được sử dụng trong hệ thống vì dễ tích hợp, ổn định, không yêu cầu phần cứng ngoại vi phức tạp. Giao tiếp UART cũng hỗ trợ truyền lệnh và dữ liệu ảnh vân tay với tốc độ lên đến 57600 bps mặc định.
- Giao tiếp không sử dụng:
 - Các chân như D+, D- (USB), TCH, VA có mặt trên module nhưng không dùng trong giao tiếp UART. Chúng phục vụ các chức năng đặc thù (chẳng hạn cho chế độ chạm hay cấp nguồn tạm thời) và thường không cần kết nối khi module được tích hợp qua UART truyền thống.

Hiện nay ngoài AS608 trên thị trường còn tồn tại nhiều loại module khác điển hình gồm:

- R305: Cảm biến vân tay quang học, phổ biến trước đây nhưng kích thước lớn và tiêu thụ điện năng cao hơn AS608
- GT521F32 / GT511C3: Module vân tay tích hợp công nghệ cảm biến điện dung hoặc quang học, thường dùng trong các ứng dụng cao cấp hơn, giá thành cao.

Các thông số kỹ thuật nổi bật của module AS608:

Bảng 2.8. Thông số kỹ thuật của module AS608

Thông số	Chi tiết
Công nghệ cảm biến	Cảm biến vân tay quang học
Độ phân giải ảnh	500 DPI
Số mẫu lưu trữ	Giới hạn 162 mẫu
Giao tiếp	UART
Nguồn cung cấp	3.3 – 6V
Dòng tiêu thụ khi nhận vân tay	~60 mA

Giải thích ý nghĩa các thông số:

DPI: là viết tắt của Dots Per Inch – nghĩa là số điểm ảnh trên mỗi inch. Đây là đơn vị dùng để đo độ phân giải của các thiết bị xử lý hình ảnh như máy quét vân tay, máy in, màn hình, cảm biến camera,...

AS608 được lựa chọn trong hệ thống vì:

- Kích thước nhỏ gọn, dễ tích hợp trên bo mạch.
- Giao tiếp UART đơn giản, dễ triển khai trên vi điều khiển.
- Có thư viện hỗ trợ rộng rãi trong môi trường Arduino, STM32.
- Giá thành hợp lý, hiệu năng ổn định, đáp ứng tốt cho hệ thống nhúng.

Trong các hệ thống nhúng và thiết bị IoT, việc quản lý năng lượng là yếu tố then chốt, đặc biệt khi hệ thống hoạt động bằng pin hoặc yêu cầu hoạt động liên tục với mức tiêu thụ điện năng thấp. Do đó, hiểu rõ các mức tiêu thụ dòng điện trong từng trạng thái hoạt động của module AS608 là cần thiết để thiết kế các giải pháp cấp nguồn, điều khiển

nguồn hoặc lập lịch làm việc hiệu quả cho hệ thống. Bảng dưới đây sẽ trình bày về các mức tiêu thụ năng lượng của AS608:

Bảng 2.9. Tổng hợp mức tiêu thụ năng lượng phổ biến của module AS608

Trạng thái	Dòng tiêu thụ
Chế độ chờ	~ 20 mA
Khi nhận vân tay	~ 60 mA
Chế độ tiết kiệm năng lượng	~ 5 mA

2.5.2. Tổng quan về Module RFID RC522

Module RFID RC522 là một giải pháp phổ biến để nhận diện thẻ từ trong các hệ thống điều khiển truy cập, chấm công, và xác thực cá nhân. RC522 hoạt động ở tần số 13.56 MHz, hỗ trợ chuẩn ISO/IEC 14443A, cho phép đọc/ghi dữ liệu trên các thẻ RFID loại MIFARE Classic. Với chi phí thấp, kích thước nhỏ gọn và dễ tích hợp qua giao tiếp SPI, RC522 là lựa chọn lý tưởng cho các ứng dụng nhúng và IoT.

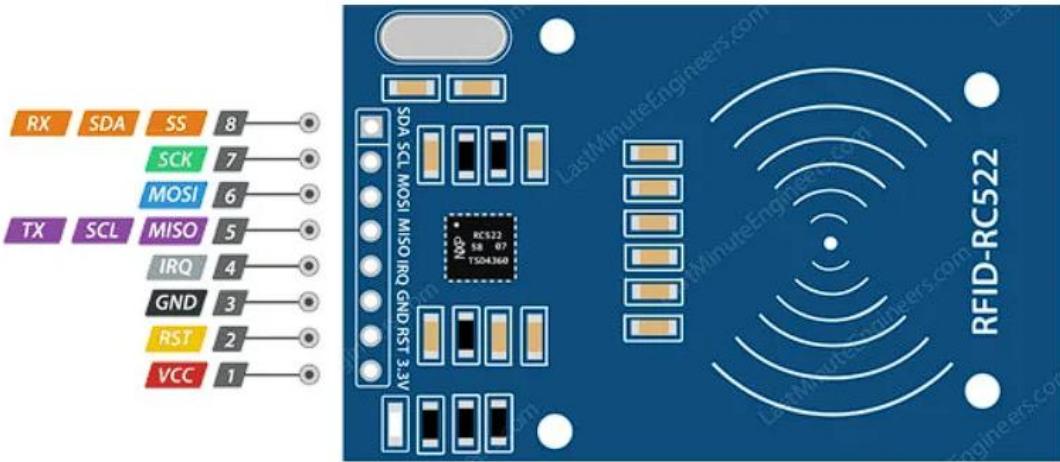
Vì mạch chính MFRC522 trên module do hãng NXP Semiconductors phát triển, tích hợp bộ thu phát RF, bộ mã hóa/giải mã dữ liệu, hỗ trợ chống va chạm thẻ và truyền thông tin ổn định với khoảng cách từ 2–5 cm.



Hình 2.12. Module RFID RC522 [28]

Để hiểu rõ khả năng vận hành của RC522, trước tiên cần nắm được các thành phần phần cứng cấu thành nên module này. RC522 được tích hợp sẵn vi mạch điều khiển RFID MFRC522, cùng các linh kiện hỗ trợ như anten RF, ổn áp và mạch giao tiếp. Module bao gồm các thành phần chính sau:

- Vi mạch MFRC522: Trung tâm xử lý dữ liệu RFID, thực hiện giao tiếp RF với thẻ và xử lý logic mã hóa, truyền nhận.
- Ăng-ten: Dạng cuộn dây tích hợp trên PCB, phát sóng và nhận tín hiệu từ thẻ RFID.
- Mạch ổn áp: Đảm bảo cung cấp nguồn 3.3V ổn định cho vi mạch chính.
- Khối giao tiếp SPI/I2C/UART: Giúp module giao tiếp với các vi điều khiển bên ngoài.
- Điện trở kéo và tụ lọc: Ổn định tín hiệu và chống nhiễu trong quá trình truyền dữ liệu.



Hình 2.13. Sơ đồ chân của Module RFID RC522

Để tích hợp phần cứng chính xác, người thiết kế hệ thống cần nắm rõ chức năng của từng chân tín hiệu. Bảng dưới đây mô tả cụ thể vai trò và trạng thái sử dụng của từng chân trong hệ thống.

Bảng 2.10. Bảng mô tả chức năng các chân trên module RFID RC522

Chân	Tên chân	Chức năng
1	Vcc	Cấp nguồn – 3.3V
2	RST	Reset
3	GND	Chân nối đất
4	IRQ	Ngắt ngoài
5	MISO/SCL/TX	<ul style="list-style-type: none"> - Dữ liệu từ module đến vi điều khiển – giao tiếp SPI (MISO) - Clock cho giao tiếp I2C (SCL) - Truyền dữ liệu trong UART (TX)

6	MOSI	Dữ liệu từ vi điều khiển đến module
7	SCK	Clock cho SPI
8	SS/SDA/RX	<ul style="list-style-type: none"> - Chọn thiết bị trong giao tiếp SPI (SS) - Chân dữ liệu trong giao tiếp I2C (SDA) - Chân nhận dữ liệu trong giao tiếp UART (RX)

Khi thiết kế ta cần quan tâm đến các thông số kỹ thuật nổi bật của module:

- Tốc độ truyền cao, phù hợp với thời gian xử lý điểm danh ngắn.
- Giao tiếp SPI đơn giản, dễ tích hợp với STM32.
- Giá thành thấp, tiết kiệm chi phí toàn hệ thống.
- Được hỗ trợ rộng rãi, có thư viện mã nguồn mở.
- Hoạt động ổn định, nhận thẻ nhanh, chính xác.

Tương tự các thiết bị ngoại vi khác, việc đánh giá mức tiêu thụ năng lượng của RC522 giúp thiết kế hệ thống cấp nguồn hiệu quả và an toàn. Bảng sau trình bày mức dòng tiêu thụ ước tính của module trong các trạng thái hoạt động phổ biến.

Bảng 2.11. Tổng hợp mức tiêu thụ năng lượng của module RFID RC522

Trạng thái	Dòng tiêu thụ
Chế độ chờ	~ 10 – 15 mA
Khi đọc thẻ	~ 20 – 30 mA
Khi reset/tạm dừng	< 5 mA

2.5.3. Tổng quan về màn hình OLED SH1106

Màn hình OLED SH1106 là một loại màn hình hiển thị đồ họa đơn sắc, sử dụng công nghệ OLED (Organic Light-Emitting Diode), cho phép hiển thị văn bản, ký tự hoặc hình ảnh với độ tương phản cao, tiết kiệm điện và góc nhìn rộng. Màn hình này thường có độ phân giải 128×64 pixel, kích thước phổ biến là 0.96 inch hoặc 1.3 inch, sử dụng driver điều khiển SH1106 do hãng Sino Wealth sản xuất.

Khác với màn hình LCD truyền thống, OLED không cần đèn nền, mỗi điểm ảnh tự phát sáng giúp tiết kiệm năng lượng và hiển thị rõ nét ngay cả trong môi trường thiếu sáng. Với kích thước nhỏ, giá thành rẻ và khả năng điều khiển đơn giản qua giao tiếp I2C hoặc SPI, SH1106 rất phù hợp trong các hệ thống nhúng, thiết bị IoT, hiển thị trạng thái, thông báo, hoặc dữ liệu cảm biến.



Hình 2.14. Màn hình OLED SH1106 [29]

Màn hình SH1106 bao gồm các thành phần chính:

- **Tấm nền OLED:** gồm ma trận điểm ảnh (pixels) tự phát sáng, độ phân giải thường là 128×64 .

- Driver điều khiển SH1106: nhận lệnh từ vi điều khiển và điều khiển từng điểm ảnh theo nội dung hiển thị.
- Mạch chuyển đổi logic: hỗ trợ giao tiếp với vi điều khiển ở mức điện áp 3.3V hoặc 5V.
- Chân giao tiếp (I2C hoặc SPI): dùng để truyền lệnh và dữ liệu từ vi điều khiển.
- Điện trở kéo, tụ lọc: ổn định tín hiệu và chống nhiễu.

Module SH1106 hỗ trợ cả hai chuẩn giao tiếp:

- I2C (hai dây: SCL, SDA)
- SPI (bốn dây: MOSI, SCK, CS, DC)

Trong phần lớn hệ thống nhúng hiện đại, I2C thường được sử dụng, do:

- Cần ít dây - đơn giản hóa thiết kế mạch
- Dễ mở rộng, chia sẻ bus với nhiều thiết bị khác
- Tốc độ truyền vừa đủ cho nội dung hiển thị

SPI tuy nhanh hơn nhưng ít được dùng với SH1106 do yêu cầu nhiều chân hơn và phức tạp hơn trong cấu hình nếu không cần tốc độ cao.

Thông số kỹ thuật nổi bật của OLED SH1106:

Bảng 2.12. Thông số kỹ thuật của OLED SH1106

Thông số	Chi tiết
Driver	SH1106
Kích thước màn hình	1.3 inch
Độ phân giải	128x64 pixels
Điện áp hoạt động	3.3 – 5V

Trong ứng dụng này oled được dùng sử dụng giao thức truyền thông I2C với sơ đồ chân chi tiết như sau:

- Vcc: chân cấp nguồn từ 3.3 – 5V
- GND: chân nối đất
- SCK: chân cấp xung clock cho giao tiếp I2C
- SDA: chân truyền dữ liệu trong giao tiếp I2C

Vì để đảm bảo nguồn cấp chính xác và ổn định cho hệ thống nên việc chú ý đến hiệu suất của từng thiết bị là rất cần thiết và màn hình oled cũng vậy tuy tiêu thụ dòng nhỏ nhưng con số này rất đáng quan tâm trong hệ thống nhưng có nguồn điện hạn chế, bảng bên dưới sẽ trình bày về các chế độ năng lượng của oled:

Bảng 2.13. Các chế độ năng lượng trên oled SH1106

Trạng thái	Dòng tiêu thụ
Khi hiển thị toàn bộ pixel	~ 20 – 30 mA
Khi hiển thị một phần nhỏ	~ 5 – 15 mA
Chế độ chờ	< 1 mA

CHƯƠNG 3. THIẾT KẾ HỆ THỐNG

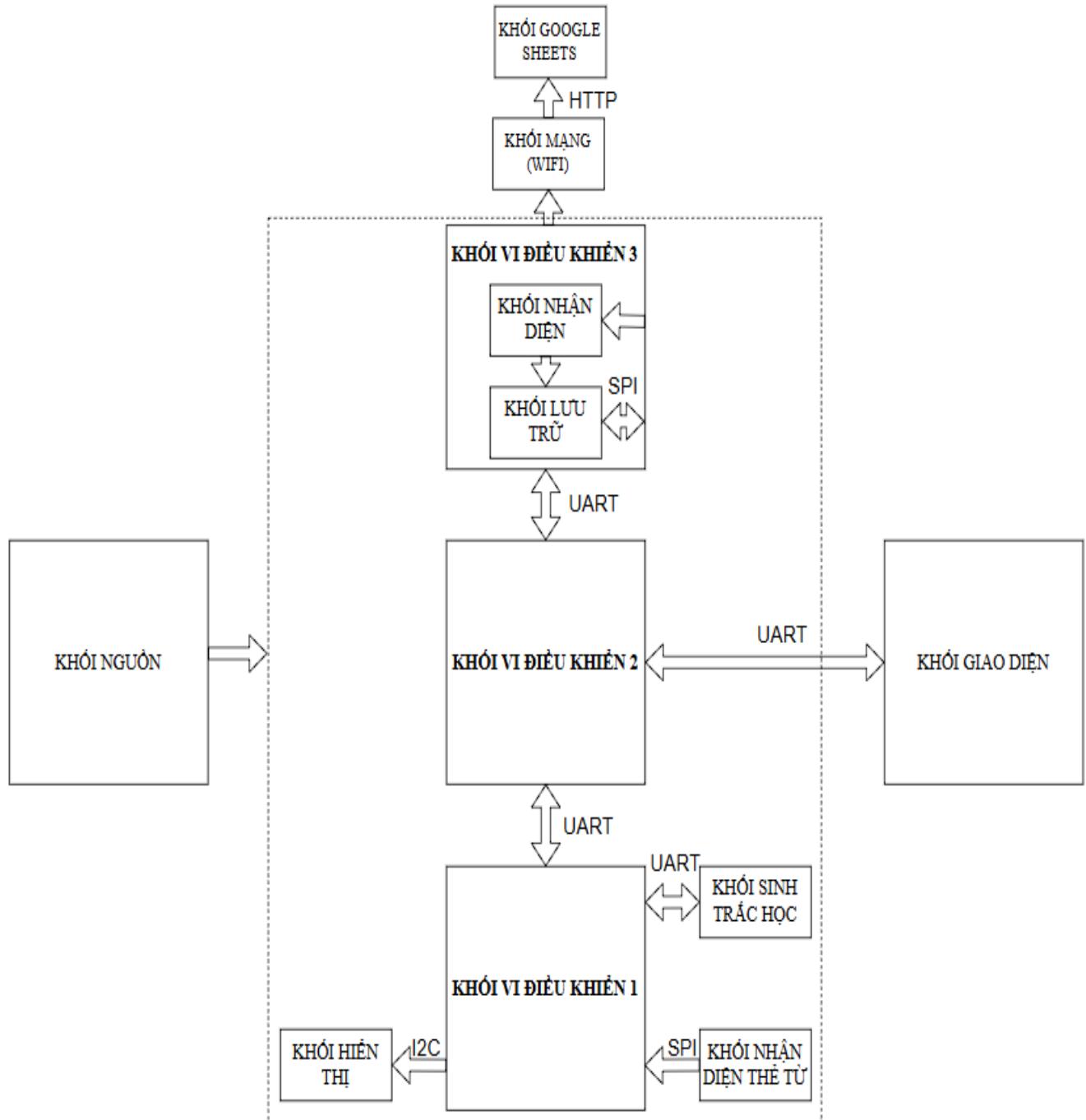
3.1. YÊU CẦU HỆ THỐNG

Từ những mục tiêu đặt ra ban đầu của đề tài “XÂY DỰNG HỆ THỐNG CHẨM CÔNG TỰ ĐỘNG ỨNG DỤNG TRONG QUẢN LÝ NHÂN SỰ”, nhóm tiến hành thiết kế và xây dựng một hệ thống có các tính năng dưới đây:

- Điểm danh bằng xác thực hai bước:
 - Điểm danh bằng thẻ RFID trước
 - Tiếp đó là điểm danh bằng vân tay
- Đăng ký người dùng mới:
 - Đăng ký thông tin thô của nhân viên bao gồm: ID, tên nhân viên và 4 byte mã hex của thẻ RFID
 - Tiếp đến là đăng ký vân tay dựa trên ID của nhân viên
- Lưu trữ thông tin nhân viên:
 - Dữ liệu của nhân viên được lưu trữ trong thẻ nhớ
 - Dữ liệu được lưu trên Google Sheets
- Tương tác giao diện quản trị:
 - Dùng các nút hoặc lệnh để đăng ký nhân viên trên giao diện ứng dụng
 - Giao diện xây dựng thân thiện dễ dàng tiếp cận
- Hệ thống hoạt động ổn định:
 - Hệ thống điểm danh tự động phải hoạt động ổn định
 - Hệ thống phải có thời gian đáp ứng nhanh chóng phù hợp với nhu cầu thực tế

3.2. SƠ ĐỒ KHỐI HỆ THỐNG

3.2.1. Sơ đồ khói



Hình 3.1. Sơ đồ khói hệ thống

3.2.2. Chức năng từng khối

- **Khối hiển thị:** Có nhiệm vụ hiển thị các thông báo, trạng thái và hướng dẫn thao tác cho người dùng trong quá trình điểm danh và đăng ký.
- **Khối nhận diện thẻ từ:** Đảm nhận vai trò nhận diện mã UID từ thẻ MIFARE Classic của nhân viên. Đây là bước xác thực đầu tiên trong quy trình điểm danh. Khối này giao tiếp với Khối vi điều khiển 1 bằng giao thức SPI
- **Khối sinh trắc học:** Thực hiện nhận dạng vân tay của nhân viên sau khi đã quét thẻ thành công. Đây là bước xác thực thứ hai, giúp tăng cường độ tin cậy. Khối này giao tiếp với Khối vi điều khiển 1 bằng giao thức UART
- **Khối vi điều khiển 1:** Chịu trách nhiệm thu thập và xử lý dữ liệu từ các cảm biến đầu vào (RFID, vân tay), đồng thời giao tiếp UART với Khối vi điều khiển 2. Sau khi xử lý, khối này điều khiển khói hiển thị để phản hồi tương tác với người dùng.
- **Khối vi điều khiển 2:** Là cầu nối trung gian giữa khói vi điều khiển 1, khói vi điều khiển 3 (UART) và người quản trị hệ thống (UART). Đảm bảo dữ liệu được truyền tải liền mạch và nhất quán.
- **Khối vi điều khiển 3:** Thực hiện các chức năng chuyên biệt bao gồm điều khiển khói nhận diện (camera) và khói lưu trữ (thẻ nhớ SD). Đồng thời, khói này chịu trách nhiệm gửi thông tin điểm danh thành công lên Google Sheets thông qua kết nối Wi-Fi.
- **Khối lưu trữ:** Dùng để lưu trữ toàn bộ thông tin nhân viên bao gồm: ID, tên, mã UID (4 byte), và ảnh chụp tại thời điểm điểm danh. Khối này được tích hợp sẵn trên vi điều khiển 3, sử dụng giao tiếp SPI để trao đổi dữ liệu.
- **Khối nhận diện:** Tích hợp trực tiếp trên vi điều khiển 3, bao gồm camera điều khiển thông qua giao thức SCCB. Có chức năng chụp ảnh nhân viên sau khi xác thực thành công.
- **Khối mạng:** Kết nối hệ thống với mạng Internet thông qua Wi-Fi, phục vụ việc gửi dữ liệu lên hệ thống lưu trữ trực tuyến như Google Sheets.

- **Khối Google Sheets:** Là nền tảng lưu trữ đám mây nơi tập hợp dữ liệu các nhân viên đã điểm danh thành công. Đồng thời cung cấp giao diện trực quan, dễ truy cập cho người quản trị.
- **Khối giao diện:** Cho phép người quản trị tương tác với hệ thống, theo dõi hoạt động điểm danh, cũng như thực hiện các thao tác như thêm, xóa nhân viên hoặc gửi lệnh điều khiển hệ thống.
- **Khối nguồn:** Cung cấp điện năng ổn định cho toàn bộ hệ thống hoạt động liên tục và hiệu quả.

3.2.3. Hoạt động của hệ thống

Hệ thống chấm công tự động được thiết kế vận hành theo chu trình xác thực hai bước, kết hợp với lưu trữ nội bộ và đồng bộ dữ liệu lên nền tảng đám mây, nhằm đảm bảo độ chính xác, an toàn và thuận tiện trong công tác quản lý nhân sự.

Khi hệ thống được cấp nguồn, các vi điều khiển và khói chức năng sẽ được khởi tạo đồng bộ. Vi điều khiển 1 sẽ tiến hành kiểm tra từng khói cảm biến xem có hoạt động ổn định không sau đó báo cho khói hiển thị, khói hiển thị sẽ hiển thị lỗi nếu có lỗi phát sinh còn nếu không có thì bắt đầu vào hiển thị các bước để hướng dẫn nhân viên điểm danh. Nhân viên bắt đầu quy trình điểm danh bằng cách đưa thẻ MIFARE đến gần khói nhận diện thẻ từ. Tại đây, module RC522 thực hiện đọc mã UID và truyền dữ liệu về vi điều khiển 1 để xác thực bước đầu. Nếu mã UID trùng khớp với dữ liệu đã đăng ký – dữ liệu này được lưu trữ trong khói lưu trữ thuộc vi điều khiển 3, vi điều khiển 1 phải gửi UID nhận được đến vi điều khiển 2 và vi điều khiển 2 yêu cầu vi điều khiển 3 trả dữ liệu (gồm ID, họ và tên người dùng) về nếu có tồn tại, còn không thì báo không tồn tại cho vi điều khiển 1 biết và điều khiển oled hiển thị dữ liệu không tồn tại yêu cầu người dùng quét thẻ lại, tiếp đó hệ thống chuyển sang bước xác thực thứ hai bằng vân tay.

Người dùng tiếp tục đặt ngón tay lên cảm biến vân tay AS608, để hệ thống đối chiếu mẫu vân tay với thông tin tương ứng với ID nhận được khi xác thực ở bước trước đó. Việc

xác thực vân tay được xử lý ngay tại vi điều khiển 1 và kết quả được phản hồi qua khói hiển thị. Nếu quá trình xác thực thành công, hệ thống sẽ thực hiện các thao tác tiếp theo.

Ngay sau khi điểm danh được xác nhận, vi điều khiển 3 sẽ kích hoạt khói nhận diện – một camera giao tiếp thông qua giao thức SCCB – để tiến hành chụp ảnh chân dung của nhân viên. Hình ảnh thu được cùng với thông tin nhân viên (UID, họ tên, thời gian điểm danh) sẽ được ghi lại và lưu trữ trên thẻ nhớ SD tích hợp trong khói lưu trữ, thông qua giao tiếp SPI.

Đồng thời, hệ thống thực hiện đồng bộ dữ liệu điểm danh vừa được ghi lại lên Google Sheets thông qua kết nối Wi-Fi của vi điều khiển 3, thông qua khói mạng. Việc đồng bộ thời gian thực này cho phép quản trị viên dễ dàng theo dõi tình trạng điểm danh của toàn bộ nhân sự từ xa.

Toàn bộ quá trình hoạt động của hệ thống được hỗ trợ và giám sát thông qua khói giao diện, nơi người quản trị có thể theo dõi trực tiếp trạng thái hệ thống, thực hiện các thao tác như thêm hoặc xóa nhân viên, xem lại lịch sử điểm danh hoặc cấu hình lại hệ thống khi cần thiết.

Nhờ sự kết hợp đồng bộ giữa các khói chức năng – từ nhận dạng, xử lý, hiển thị, chụp ảnh đến lưu trữ và đồng bộ dữ liệu – hệ thống đảm bảo tính bảo mật cao, tốc độ phản hồi nhanh, khả năng mở rộng tốt và phù hợp cho các ứng dụng quản lý nhân sự hiện đại.

3.3. THIẾT KẾ TÙNG KHỐI

3.3.1. Thiết kế khói nhận diện thẻ từ

Trong hệ thống chấm công tự động, khói nhận diện thẻ từ đóng vai trò là bộ phận nhận dạng đầu tiên, tiếp nhận thông tin định danh từ thẻ nhân viên thông qua mã UID. Để đảm bảo hiệu quả hoạt động và tính ổn định của toàn hệ thống, việc thiết kế phần cứng cho khói RFID cần hướng tới các mục tiêu sau:

- Thứ nhất, đảm bảo khả năng nhận diện chính xác mã UID từ thẻ MIFARE Classic trong khoảng cách 2–5 cm. Module cần hoạt động ổn định trong môi trường thực tế có thể có nhiễu điện từ và ánh sáng phức tạp.
- Thứ hai, tối ưu hóa kết nối phần cứng giữa module RC522 và vi điều khiển STM32 thông qua giao tiếp SPI. Việc thiết kế phải đảm bảo đúng mức điện áp logic (3.3V), tốc độ truyền dữ liệu phù hợp, tránh xung nhiễu và giảm thiểu lỗi truyền nhận.
- Thứ ba, bố trí vị trí module RFID trên hệ thống một cách hợp lý, thuận tiện cho thao tác người dùng và tránh bị che chắn anten, đồng thời duy trì khoảng cách tối ưu giữa RC522 và các thành phần linh kiện khác để hạn chế nhiễu xuyên kẽm.
- Thứ tư, đảm bảo tiêu thụ điện năng thấp và tương thích tốt với nguồn cấp hệ thống, giúp hệ thống hoạt động ổn định ngay cả trong điều kiện sử dụng nguồn hạn chế.
- Thứ năm, hỗ trợ khả năng báo hiệu trạng thái quét thẻ thông qua LED hoặc màn hình hiển thị, tạo sự tương tác trực quan giữa hệ thống và người dùng.

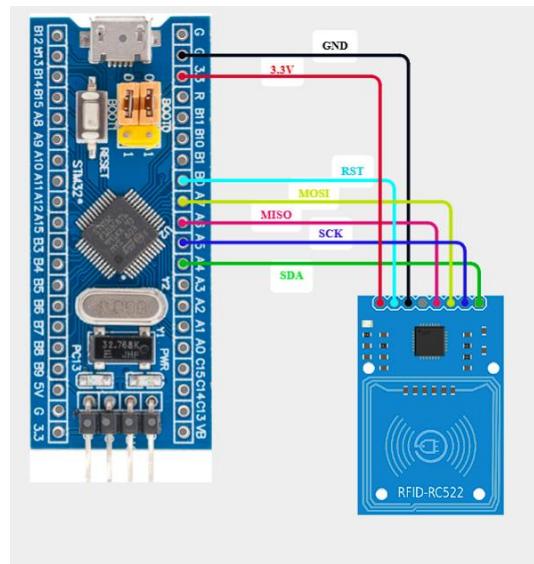
Những mục tiêu trên là cơ sở để triển khai thiết kế phần cứng hiệu quả, đảm bảo RC522 phát huy tối đa khả năng nhận diện, đồng thời tích hợp mượt mà với hệ thống điều khiển trung tâm.

Để đảm bảo khối RFID hoạt động ổn định và đáp ứng các mục tiêu thiết kế đã đặt ra, phần cứng của khối này được cấu hình xoay quanh module RC522 – một module RFID phổ biến, hoạt động ở tần số 13.56 MHz.

Các thành phần phần cứng chính bao gồm:

- Vi điều khiển 1 (STM32) kết nối với module RFID qua SPI
- Module RFID RC522 (có đèn báo), được cấp nguồn từ chân 3.3V của vi điều khiển.
- Nguồn adapter (cấp vào chân 5V của vi điều khiển)

Tiếp theo, việc thiết kế sơ đồ nguyên lý và sơ đồ kết nối cho khối RFID cần đảm bảo các tiêu chí: giao tiếp ổn định với vi điều khiển, hạn chế nhiễu tín hiệu, và thuận tiện cho thao tác quét thẻ từ phía người dùng. Module RC522 sử dụng giao tiếp SPI nên yêu cầu kết nối chính xác và logic với các chân điều khiển của vi điều khiển STM32.



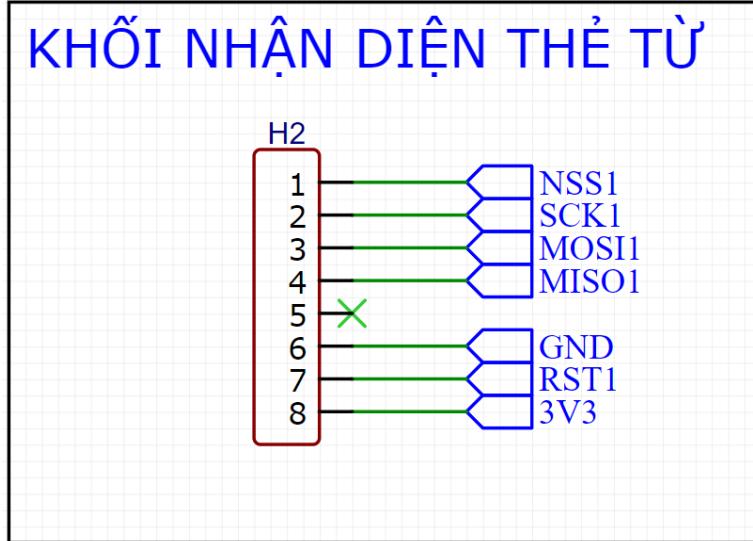
Hình 3.2. Kết nối STM32 Blue pill với module RFID RC522

Trong đó STM32 Blue pill nhận nguồn chung từ adapter nối vào chân 5V và GND vào GND chung.

Bảng 3.1. Sơ đồ kết nối STM32 Blue pill và module RFID RC522

Nguồn Adapter	STM32 Blue pill	RFID RC522
Cáp vào chân 5V của STM32	3.3	3.3V
	G	GND
	B0	RST
	A7	MOSI
	A6	MISO
	A5	SCK
	A4	SDA

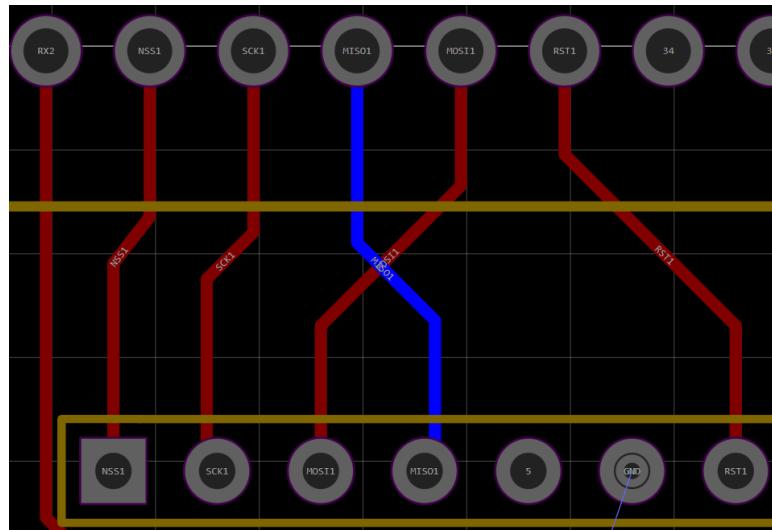
KHỐI NHẬN DIỆN THẺ TỪ



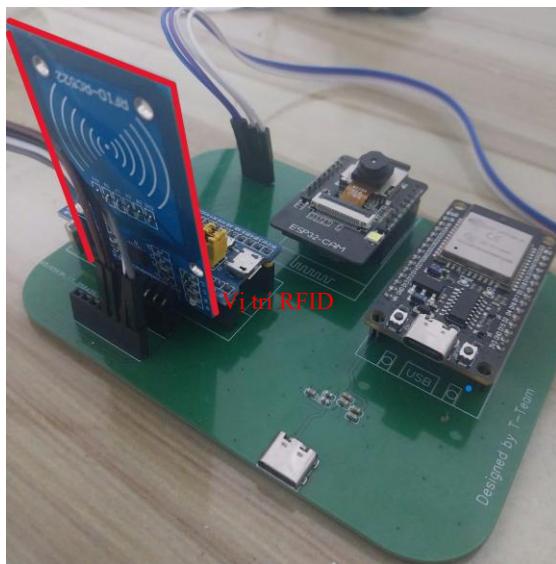
Hình 3.3. Sơ đồ nguyên lý khói nhận diện thẻ từ

Trên board mạch in, việc bố trí RC522 cần được thực hiện theo các nguyên tắc sau:

- Ăng-ten tích hợp trên RC522 nên được đặt gần mép ngoài của thiết bị, không bị che chắn bởi vỏ nhựa hoặc kim loại, để đảm bảo vùng quét tín hiệu RFID không bị hạn chế.
- Đường SPI từ RC522 đến STM32 nên ngắn và thẳng, tránh đi vòng hoặc song song với các đường tín hiệu cao tần khác, để giảm hiện tượng nhiễu chéo.



Hình 3.4. Đường dây trên mạch in



Hình 3.5. Vị trí đặt module RFID RC522 trên mạch in

Một số lưu ý quan trọng khi thiết kế:

- Sử dụng kích thước pad và footprint tiêu chuẩn cho header RC522 để dễ hàn và bảo trì.
- Nếu thiết kế module tháo rời, nên dùng socket hoặc header chuẩn 2.54 mm để kết nối RC522 với main board.

- Nên sử dụng dây tín hiệu có độ rộng 0.3mm cho giao thức SPI này vì đây là dây tín hiệu tốc độ cao, khoảng cách ngắn, và tiết kiệm diện tích mạch.
- Dây nguồn 3.3V thì nên dùng dây có độ rộng khoảng 0.6mm vì để đảm bảo đủ dòng điện cấp (thường dùng với dòng điện dưới 500mA), tránh tỏa nhiệt và sụt áp trên đường truyền (theo chuẩn IPC-2221)

Việc thiết kế sơ đồ nguyên lý và bố trí PCB hợp lý sẽ đảm bảo RC522 hoạt động ổn định trong thời gian dài, đồng thời mang lại trải nghiệm người dùng tốt hơn khi thao tác với hệ thống chấm công.

3.3.2. Thiết kế khối sinh trắc học

Khối sinh trắc học có vai trò then chốt trong hệ thống, đảm nhiệm bước xác thực thứ hai – xác minh đặc điểm sinh trắc học của người dùng (vân tay) sau khi đã quét thẻ RFID. Thiết kế phần cứng cho khối này cần đảm bảo các yêu cầu sau:

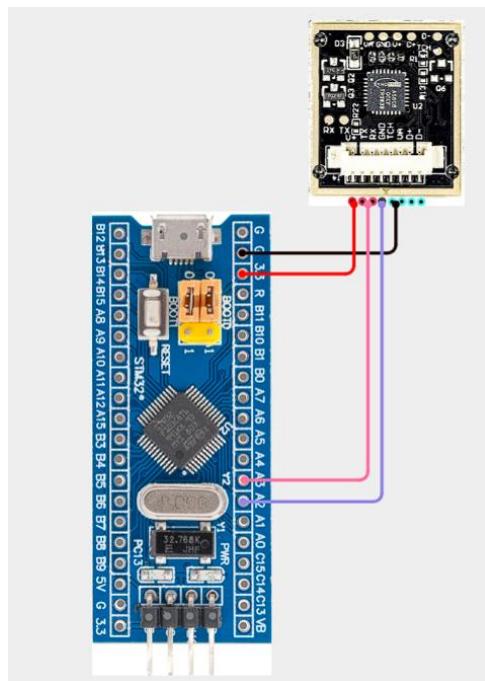
- Thứ nhất, nhận diện vân tay nhanh chóng và chính xác, đảm bảo độ tin cậy cao trong xác thực người dùng.
- Thứ hai, giao tiếp ổn định với vi điều khiển STM32 qua UART, tốc độ truyền phù hợp với dung lượng dữ liệu vân tay (57600 bps).
- Thứ ba, đảm bảo điện áp hoạt động đúng mức (3.3V) và bố trí hợp lý để người dùng dễ dàng đặt ngón tay vào cảm biến.
- Thứ tư, đảm bảo mức tiêu thụ điện hợp lý, không ảnh hưởng đến tổng tải của hệ thống và duy trì hoạt động ổn định khi chạy liên tục.

Khối này sử dụng module cảm biến vân tay AS608, một giải pháp sinh trắc học phổ biến và hiệu quả trong các hệ thống nhúng. Thành phần phần cứng gồm:

- Module cảm biến vân tay AS608 giao tiếp UART với vi điều khiển
- Vi điều khiển STM32 Blue pill

- Nguồn 3.3V cấp từ vi điều khiển
- Nguồn adapter (cấp vào chân 5V của vi điều khiển)

Để đảm bảo khả năng kết nối và hoạt động ổn định giữa cảm biến vân tay AS608 và vi điều khiển STM32, sơ đồ nguyên lý được xây dựng nhằm thể hiện rõ các chân kết nối, mức điện áp phù hợp và chuẩn giao tiếp UART được sử dụng. Chi tiết kết nối được trình bày như sau:

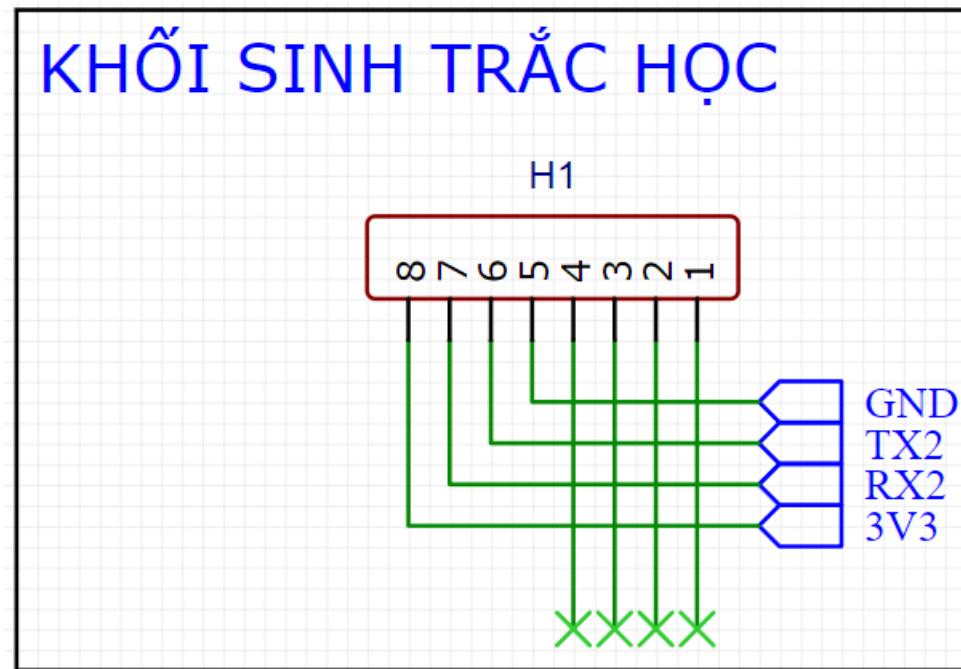


Hình 3.6. Kết nối của STM32 Blue pill và module cảm biến vân tay

Trong đó STM32 được cấp nguồn từ adapter vào chân 5V và GND vào GND chung.

Bảng 3.2. Sơ đồ kết nối của STM32 Blue pill và module cảm biến vân tay

Nguồn Adapter	STM32 Blue pill	Module vân tay AS608
Cáp vào chân 5v của STM32	3.3	V+
	G	GND
	A3	TX
	A2	RX



Hình 3.7. Sơ đồ nguyên lý của module vân tay AS608

Mạch không yêu cầu sử dụng thêm bất kỳ giao tiếp phức tạp nào ngoài UART.

Về bố trí thực tế trên PCB, cảm biến cần được đặt tại vị trí thuận tiện cho người dùng thao tác trực tiếp, với mặt quét hướng ra ngoài và không bị cản trở bởi vỏ thiết bị. Đường truyền UART nên ngắn gọn, tránh đi song song với đường xung clock hoặc nguồn để hạn chế nhiễu chéo.



Hình 3.8. Bố trí header cho module vân tay

Trong mạch, cảm biến vân tay được nối với một hàng rào cái 8 chân 2.54mm, hàng rào này được bố trí ở một vị trí phù hợp và gần với STM32 Blue pill để giao tiếp UART diễn ra tốt hơn.

Khi vẽ mạch nên chú ý:

- Dây tín hiệu nên vẽ có độ rộng 0.3mm giống như giao tiếp SPI ở trên
- Dây nguồn 3.3V có độ rộng 0.6mm
- Header dùng pad và footprint mặc định

3.3.3. Thiết kế khói hiển thị

Khói hiển thị đóng vai trò là giao diện phản hồi trực quan giữa hệ thống và người dùng. Tại đây, người dùng sẽ nhận được hướng dẫn thao tác, kết quả xác thực và các trạng thái hệ thống. Thiết kế phần cứng của khói này cần đảm bảo các mục tiêu sau:

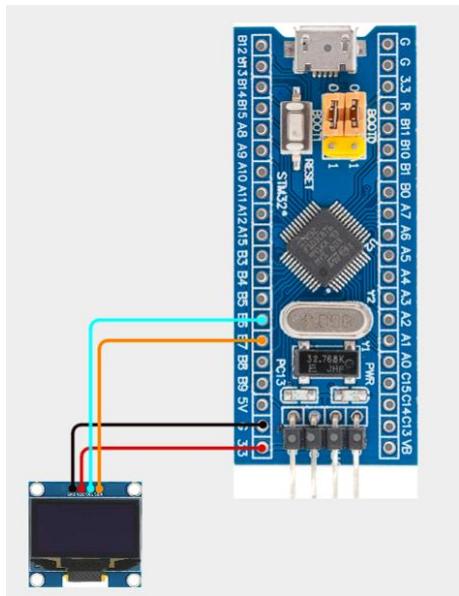
- Thứ nhất, hiển thị rõ ràng thông tin cần thiết (văn bản, biểu tượng), dễ đọc trong nhiều điều kiện ánh sáng.

- Thứ hai, sử dụng màn hình có giao tiếp đơn giản, chiếm ít chân I/O, phù hợp với tài nguyên phần cứng của hệ thống.
- Thứ ba, đảm bảo khả năng tương thích điện áp và tốc độ truyền dữ liệu phù hợp để hiển thị thời gian thực.
- Thứ tư, tiêu thụ điện năng thấp, phù hợp với hệ thống nhúng sử dụng nguồn hạn chế.

Khối hiển thị sử dụng màn hình OLED điều khiển bằng driver SH1106, với độ phân giải 128×64 pixel, hỗ trợ giao tiếp I²C, khối này được điều khiển bởi khói vi điều khiển 1. Các thành phần chính gồm:

- Màn hình OLED SH1106 – giao tiếp I²C (điện trở 4.7k ohm đã được tích hợp vào module)
- Vi điều khiển STM32 Blue pill
- Nguồn cấp 3.3V từ vi điều khiển
- Nguồn adapter cấp vào chân 5V cho vi điều khiển

Kết nối giữa màn hình SH1106 và vi điều khiển STM32 được thực hiện như sau:



Hình 3.9. Kết nối của màn hình OLED và STM32 Blue pill

Sơ đồ kết nối:

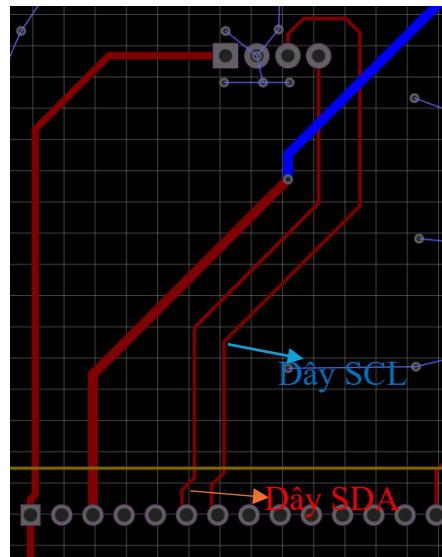
Bảng 3.3. Sơ đồ kết nối của OLED và STM32 Blue pill

Nguồn Adapter	STM32 Blue pill	OLED
Cấp vào chân 5v của STM32	3.3	VCC
	G	GND
	B7	SDA
	B6	SCL

Hai điện trở kéo lên (4.7k ohm) được nối từ SCL và SDA lên nguồn 3.3V để đảm bảo giao tiếp ổn định theo chuẩn I2C (đã tích hợp trên module).

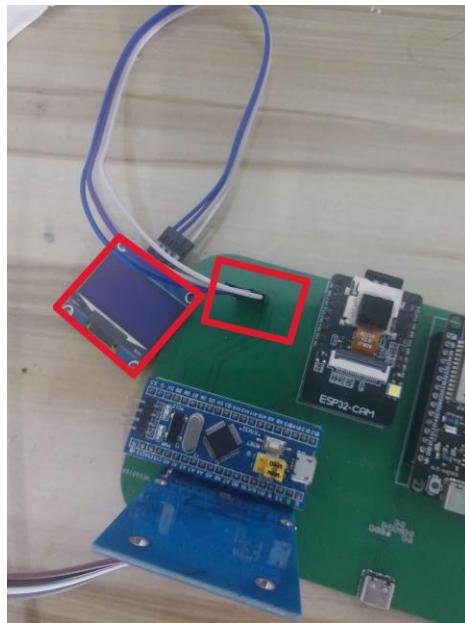
Cách bố trí linh kiện trên mạch in sao cho: hai dây của giao thức I2C đặt xa các dây nguồn để tránh nhiễu điện từ, màn hình oled đặt sao cho người dùng dễ nhìn, có thể cắm dây vào header để có dây dài hơn để sau này có thể lắp lên hộp. Đặt xa các linh kiện khác để tránh bị ảnh hưởng về nhiệt độ.

Dưới đây là cách bố trí linh kiện và đi dây sao cho phù hợp với các vấn đề vừa đặt ra ở trên:



Hình 3.10. Cách đi dây cho I2C trên OLED

Cách bố trí module OLED trên mạch in:



Hình 3.11. Cách bố trí OLED trên mạch in

Một số lưu ý khi vẽ mạch:

- Vẽ dây tín hiệu có độ rộng 0.3 mm
- Vẽ dây nguồn 3.3V có độ rộng 0.6 mm
- Đặt module xa các thiết bị khác để tránh ảnh hưởng về nhiệt
- Bố trí dây sao cho hợp lý và ngắn nhất có thể
- Đặt dây tín hiệu càng xa các dây nguồn càng tốt vì để tránh nhiễu điện từ
- Dùng pad và footprint của header là mặc định

3.3.4. Thiết kế khối vi điều khiển 1

Khối vi điều khiển 1 được xem là trung tâm xử lý dữ liệu cục bộ đầu tiên trong chuỗi vận hành của hệ thống. Đây là nơi tiếp nhận tín hiệu từ các thiết bị đầu vào như module đọc thẻ RFID RC522 và cảm biến vân tay AS608, thực hiện các bước kiểm tra xác thực ban đầu, điều phối hiển thị, đồng thời truyền dữ liệu đã xử lý đến các khối vi điều khiển cấp cao hơn.

Việc thiết kế phần cứng cho khối này đòi hỏi phải đáp ứng đồng thời các yêu cầu về tốc độ xử lý, khả năng giao tiếp đa dạng, độ ổn định cao trong môi trường thực tế, và khả năng mở rộng trong tương lai. Cụ thể:

- Thứ nhất, vi điều khiển phải đảm nhiệm được tác vụ xử lý đa nhiệm, phản hồi nhanh với tín hiệu đầu vào từ cả RFID và cảm biến vân tay mà không gây xung đột hoặc trễ.
- Thứ hai, đảm bảo giao tiếp chính xác và ổn định với các ngoại vi sử dụng nhiều chuẩn khác nhau: SPI (RC522), UART (AS608), I²C (OLED), UART nội bộ (truyền sang STM32 khác).
- Thứ ba, dễ dàng cấu hình, lập trình và cập nhật phần mềm thông qua cổng SWD hoặc UART Bootloader.

- Thứ tự, tiêu thụ điện năng ở mức thấp, hoạt động ổn định trong thời gian dài, đặc biệt trong các ứng dụng sử dụng nguồn pin hoặc năng lượng giới hạn.

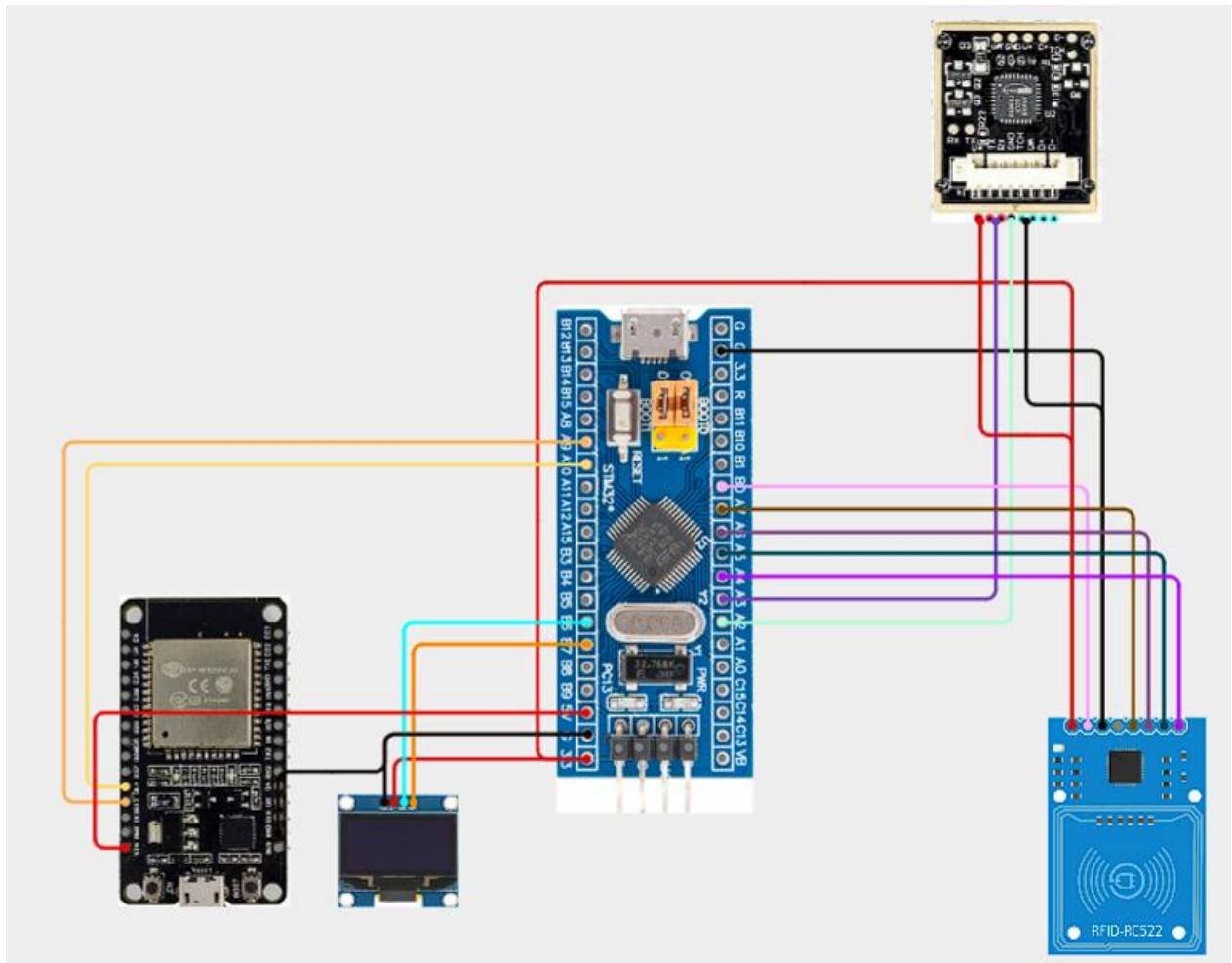
Khối vi điều khiển 1 sử dụng STM32F103C8T6, một vi điều khiển phổ biến thuộc dòng ARM Cortex-M3, đáp ứng tốt các yêu cầu xử lý và giao tiếp trong hệ thống.

Các thành phần phần cứng chính bao gồm:

- Vi điều khiển STM32 Blue pill
- Nguồn cấp từ adapter
- Giao tiếp ngoại vi:
 - SPI để kết nối với module RC522
 - UART1 kết nối cảm biến vân tay AS608
 - UART2 truyền dữ liệu sang STM32 thứ hai
 - I²C điều khiển màn hình OLED SH1106
 - GPIO điều khiển LED, nút nhấn và các tín hiệu điều khiển khác

Thiết kế phần cứng được bố trí khoa học, đảm bảo khả năng giao tiếp ổn định, xử lý nhanh và tương thích tốt với các khái niệm năng còn lại của hệ thống.

Dưới đây là sơ đồ kết nối của khái vi điều khiển 1 với các thành phần quanh nó:



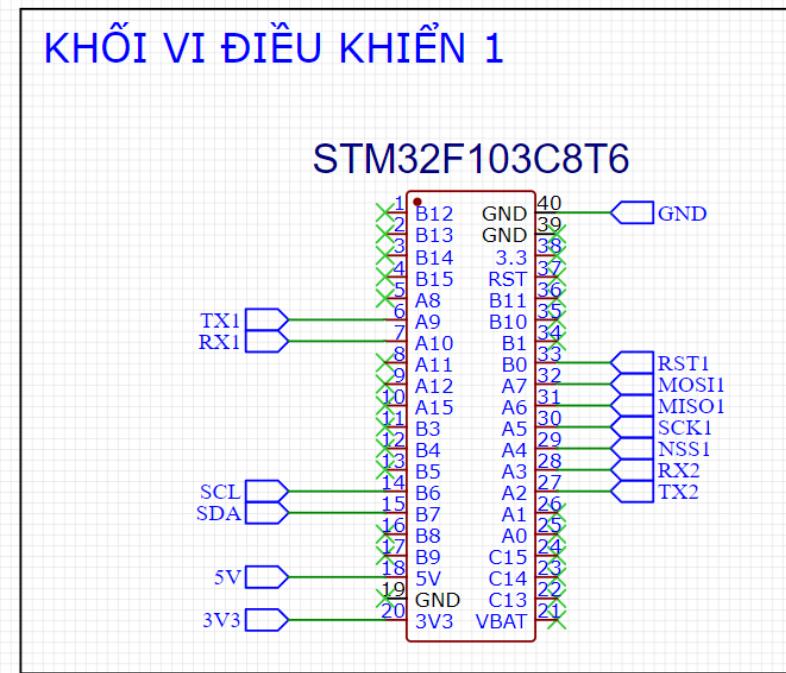
Hình 3.12. Kết nối của khối vi điều khiển 1 với các thành phần quanh nó

Dưới đây là bảng trình bày sơ đồ kết nối của khối vi điều khiển 1 với các thành phần quanh nó:

Bảng 3.4. Sơ đồ kết nối của khối vi điều khiển 1 với các thành phần xung quanh

Nguồn Adapter	STM32 Blue pill	Module AS608	RFID RC522	OLED SH1106	ESP32
Cấp vào chân 5V của STM32	5V	x	x	x	Vin
	3.3	V+	3.3V	Vcc	x
	G	GND	GND	GND	GND
	A2	RX	x	x	x
	A3	TX			
	A4	x	SDA		
	A5		SCK		
	A6		MOSI		
	A7		MISO		
	A9		x		
	A10		RST		D12
	B0		x		D14
	B6		SCL	x	
	B7		SDA		

Sơ đồ nguyên lý của khối vi điều khiển 1:



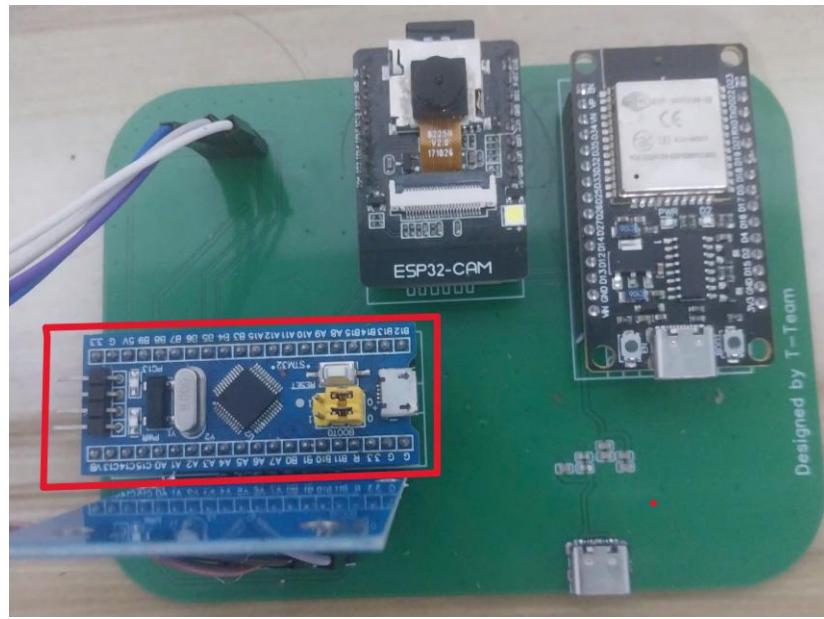
Hình 3.13. Sơ đồ nguyên lý của khối vi điều khiển 1

Sơ đồ nguyên lý tập trung thể hiện rõ các kết nối giữa vi điều khiển STM32 và các ngoại vi. Các điểm nối bao gồm:

- Kết nối SPI đến RC522 với đầy đủ 4 đường tín hiệu
- I2C được chia sẻ với OLED, có điện trở kéo lên 4.7k ohm để ổn định tín hiệu (đã có sẵn trên module)
- Tụ decoupling 100nF đặt sát mỗi chân VCC để triệt nhiễu nguồn (được trình bày trong khối nguồn)

Cách bố trí mạch in:

- STM32 được đặt ở gần các khối nhận diện thẻ từ, sinh trắc học, khối hiển thị để đường dây đi qua các khối này đảm bảo ngắn nhất có thể.
- Đường SPI và UART được đi ngắn nhất có thể, hợp lý và tiết kiệm diện tích mạch in.
- Khu vực nạp code cho STM32 bằng ST-link quay ra ngoài và nằm sát với rìa mạch in.
- Các tụ lọc, điện trở kéo được đặt sát linh kiện liên quan để giảm nhiễu.



Hình 3.14. Vị trí đặt STM32 trên mạch in

3.3.5. Thiết kế khối vi điều khiển 2

Khối vi điều khiển 2 đảm nhiệm vai trò cầu nối trung tâm, có nhiệm vụ tiếp nhận dữ liệu từ khối STM32 (vi điều khiển 1), đồng thời điều phối luồng dữ liệu sang khối nhận diện và lưu trữ (vi điều khiển 3 - ESP32-CAM). Ngoài ra, đây cũng là thành phần có thể thực hiện kiểm tra logic, lọc dữ liệu và sửa lỗi cho hệ thống.

Việc thiết kế phần cứng cho khối này cần bảo đảm các yếu tố:

- Thứ nhất, hỗ trợ giao tiếp UART song song với cả STM32 và ESP32-CAM một cách ổn định, đồng thời duy trì tốc độ truyền không bị nghẽn hoặc mất gói.
- Thứ hai, hoạt động ổn định ở mức điện áp 3.3V, tương thích logic với các vi điều khiển khác trong hệ thống.
- Thứ ba, có khả năng xử lý trung gian, hỗ trợ phân luồng truyền và bổ sung các tính năng bảo mật hoặc mã hóa nếu cần.

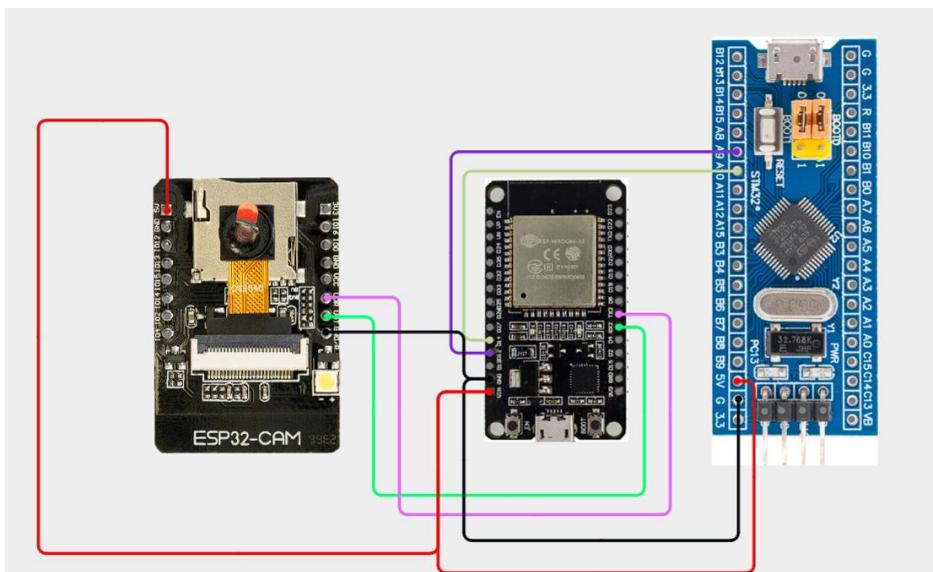
- Thứ tư, hỗ trợ nạp chương trình dễ dàng, có thể mở rộng chức năng xử lý nếu yêu cầu hệ thống phát triển thêm.

Khối này sử dụng ESP32-WROOM-32 DevKit v1, một dòng vi điều khiển tích hợp Wi-Fi và Bluetooth, cấu hình linh hoạt và giàu tài nguyên phần cứng. Các thành phần chính bao gồm:

- Vị trí điều khiển ESP32-WROOM-32
 - Nguồn cấp nhận từ adapter
 - Các UART

Khối ESP32 được thiết kế gọn gàng, kết nối trực tiếp với hai khối vi điều khiển còn lại thông qua UART, không yêu cầu thêm chip chuyển đổi mức logic vì tất cả các thiết bị đều hoạt động ở 3.3V.

Dưới đây là kết nối của khối vi điều khiển 2 với khối vi điều khiển 1 và 3:



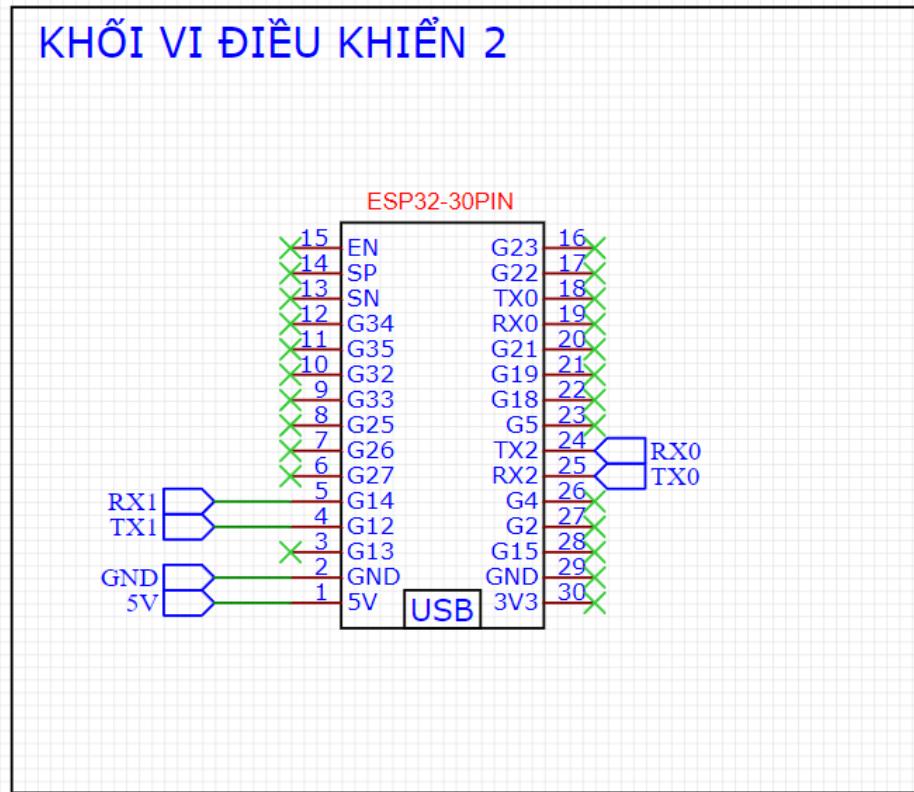
Hình 3.15. Kết nối của khối vi điều khiển 2 với các thành phần xung quanh

Đây là sơ đồ kết nối:

Bảng 3.5. Sơ đồ kết nối của khói vi điều khiển 2 với các thành phần khác

Nguồn Adapter	ESP32 Devkit	STM32 Blue pill	ESP32-CAM
Cấp vào chân 5V của ESP32 Devkit	Vin	5V	5V
	GND	G	GND
	TX2	x	U0R
	RX2		U0T
	D12	A10	x
	D14	A9	

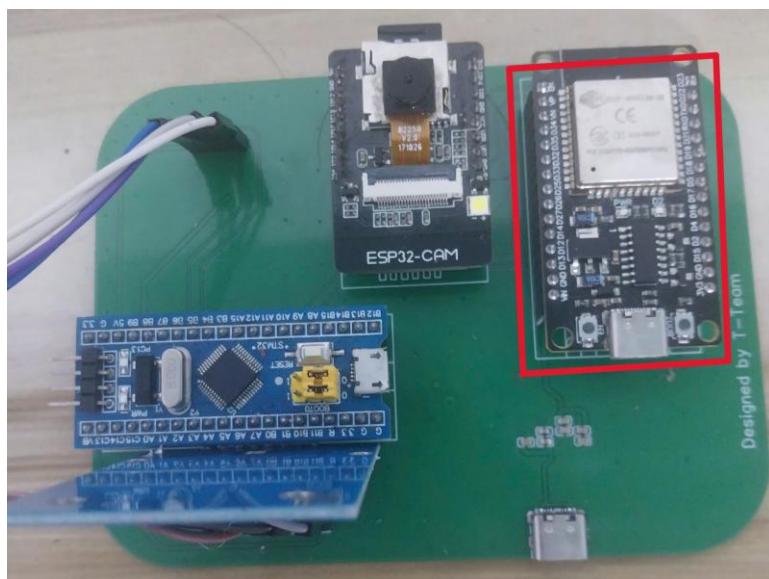
Dưới đây là sơ đồ nguyên lý:



Hình 3.16. Sơ đồ nguyên lý của khói vi điều khiển 2

Cách bố trí trên mạch in:

- Đặt vị trí hợp lý, tắt kiệm điện tích mạch in
- Cổng USB hướng xuống để dễ dàng cắm dây nguồn và sửa lỗi
- Đặt sao cho đường dây tín hiệu đi ngắn nhất có thể



Hình 3.17. Cách bố trí khói vi điều khiển 2 trên mạch in

3.3.6. Thiết kế khói vi điều khiển 3

Khối vi điều khiển 3 đóng vai trò là trung tâm xử lý ảnh, lưu trữ dữ liệu hình ảnh và đồng bộ dữ liệu điểm danh lên nền tảng đám mây. Sau khi nhân viên được xác thực thành công bởi các khói trước, khói này sẽ chụp ảnh khuôn mặt, lưu ảnh kèm thông tin vào thẻ nhớ microSD, đồng thời gửi dữ liệu lên Google Sheets thông qua Wi-Fi.

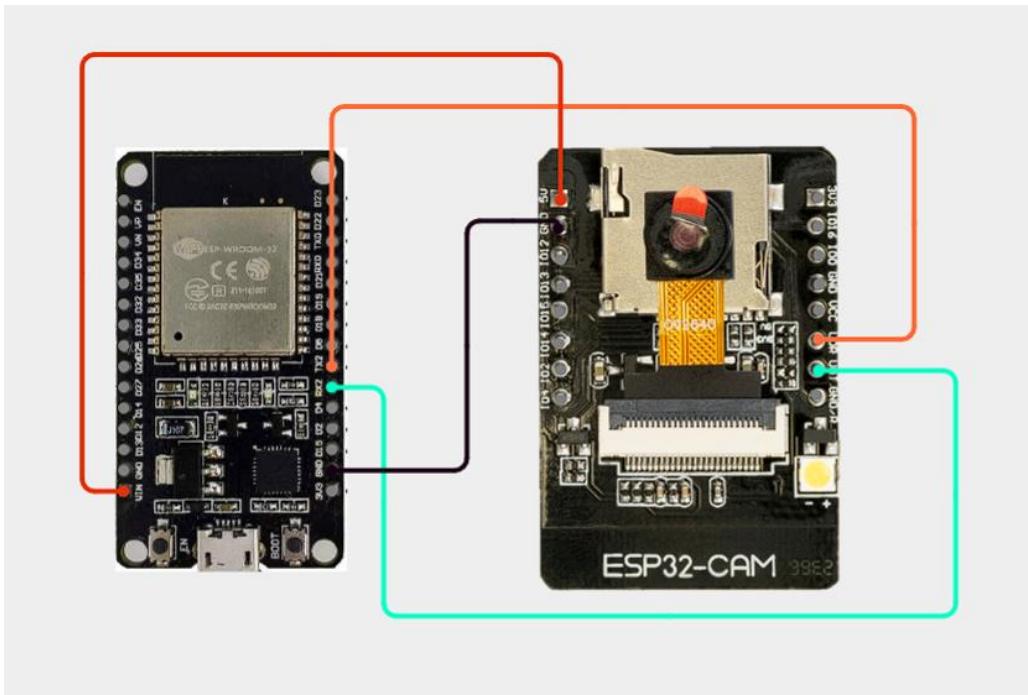
Thiết kế phần cứng cho khói ESP32-CAM cần đáp ứng các yêu cầu:

- Thứ nhất, hỗ trợ xử lý ảnh ở độ phân giải phù hợp, hoạt động ổn định với camera OV2640 tích hợp.
- Thứ hai, có khả năng lưu trữ ảnh vào thẻ nhớ microSD, với tốc độ ghi đủ nhanh, tránh nghẽn dữ liệu.

- Thứ ba, hỗ trợ giao tiếp UART để nhận tín hiệu điều khiển từ vi điều khiển 2 (ESP32), đồng thời kích hoạt chức năng chụp ảnh, lưu file và truyền dữ liệu.
- Thứ tư, tích hợp Wi-Fi để đồng bộ dữ liệu điểm danh lên hệ thống quản lý trên cloud (Google Sheets).

Khối này sử dụng ESP32-CAM, một module vi điều khiển tích hợp camera OV2640, Wi-Fi, khe cắm thẻ nhớ microSD và đủ tài nguyên để xử lý hình ảnh và truyền dữ liệu không dây.

Dưới đây là sơ đồ kết nối của khối vi điều khiển 3 với các thành phần quanh nó:



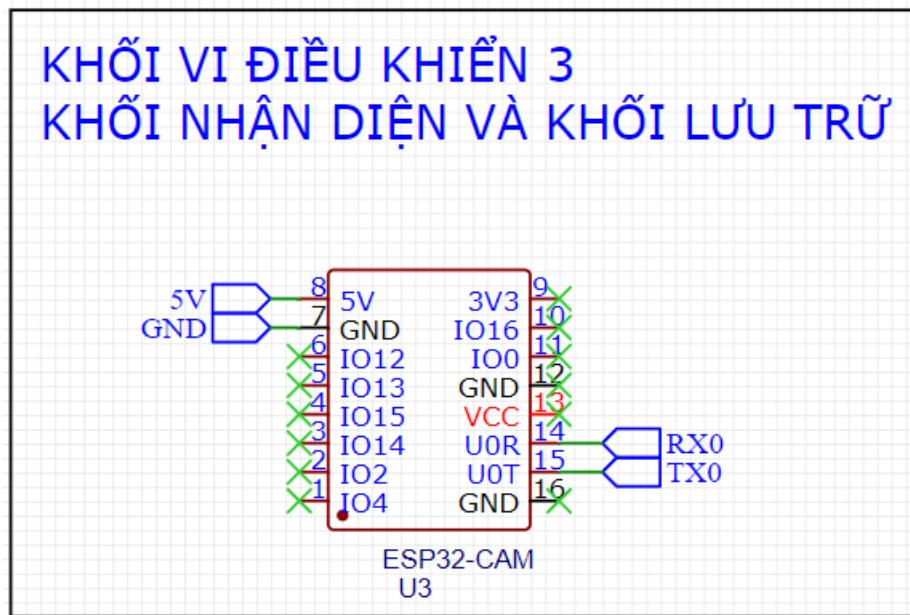
Hình 3.18. Kết nối chân của ESP32-CAM với ESP32 Devkit

Sơ đồ kết nối:

Bảng 3.6. Sơ đồ kết nối của ESP32 Devkit và ESP32-CAM

Nguồn adapter	ESP32 Devkit	ESP32-CAM
Cáp vào chân 5V của khối vi điều khiển 2 hoặc 3 đều được	5V	5V
	GND	GND
	TX2	U0R
	RX2	U0T

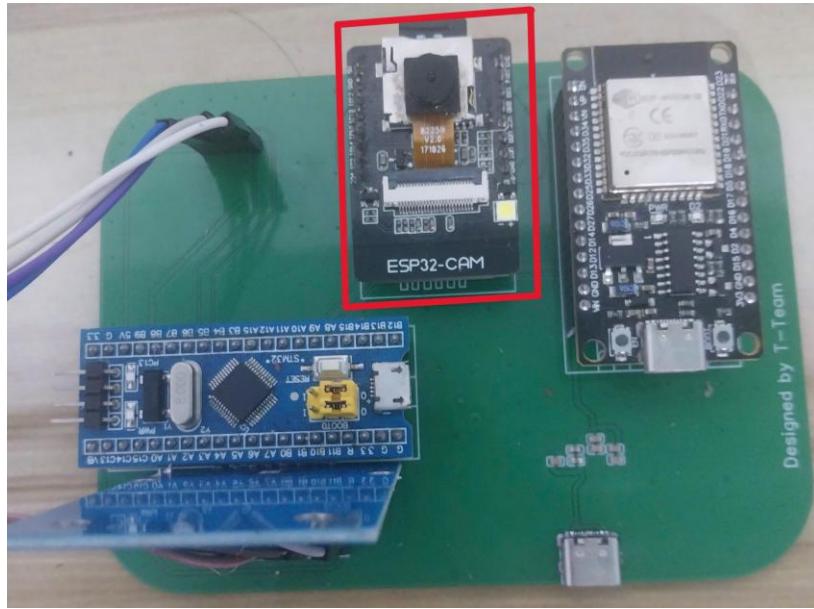
Dưới đây là sơ đồ nguyên lý:



Hình 3.19. Sơ đồ nguyên lý của khối vi điều khiển 3

Cách bố trí trên mạch in:

- Camera cần được đặt lộ ra ngoài, dễ hướng vào mặt người dùng khi điểm danh.
- Module ESP32-CAM nên được tách rời bằng header, thuận tiện tháo lắp và bảo trì.
- Thẻ nhớ SD nên để mặt trước dễ thao tác, dễ lấy dữ liệu thủ công khi cần.
- Tụ lọc đặt gần chân nguồn, đảm bảo ổn định điện áp khi ghi ảnh và bật Wi-Fi (trình bày ở khối nguồn).



Hình 3.20. Cách bố trí ESP32-CAM trên mạch in

3.3.7. Thiết kế khói lưu trữ

Khói lưu trữ là bộ phận chịu trách nhiệm ghi nhận, lưu trữ và truy xuất dữ liệu quan trọng trong hệ thống chấm công tự động. Dữ liệu bao gồm: ID nhân viên, họ tên, mã UID từ thẻ RFID, hình ảnh chụp tại thời điểm điểm danh và thời gian điểm danh tương ứng. Thiết kế phần cứng của khói này cần đảm bảo:

- Thứ nhất, lưu trữ được cả dữ liệu dạng văn bản (file .txt) và dữ liệu nhị phân (ảnh định dạng JPG).
- Thứ hai, đảm bảo tốc độ ghi đủ nhanh để đáp ứng hoạt động liên tục trong các ca điểm danh.
- Thứ ba, dung lượng lưu trữ phải đủ lớn để phục vụ hệ thống trong thời gian dài mà không cần thay đổi thường xuyên.
- Thứ tư, dễ dàng tháo lắp, truy xuất hoặc sao lưu dữ liệu thủ công khi cần thiết, bên cạnh việc đồng bộ lên cloud.

Khối lưu trữ sử dụng thẻ nhớ microSD, giao tiếp với vi điều khiển 3 (ESP32-CAM) thông qua giao thức SPI, được tích hợp sẵn trên module ESP32-CAM.

Vì ESP32-CAM tích hợp sẵn khe cắm thẻ nhớ microSD, các kết nối phần cứng nội bộ đã được định sẵn theo thiết kế của nhà sản xuất. Khi thiết kế chỉ cần đảm bảo:

- Cấp nguồn ổn định 3.3V/5V từ mạch nguồn chính (qua ổn áp nội bộ)
- Cấu hình chính xác các chân SPI trong phần mềm tương ứng với sơ đồ chân phần cứng

Cách bố trí mạch: Thẻ nhớ microSD nên được bố trí ở vị trí dễ tháo lắp, thuận tiện cho việc kiểm tra hoặc sao lưu thủ công.

3.3.8. Thiết kế khói nhận diện

Khối nhận diện là thành phần đảm nhiệm chức năng chụp ảnh người dùng ngay sau khi xác thực điểm danh thành công. Hình ảnh được ghi lại sẽ được lưu trữ cục bộ trên thẻ nhớ và đồng thời có thể được gửi lên hệ thống lưu trữ đám mây nhằm phục vụ việc quản lý, giám sát và đối chiếu khi cần thiết.

Khối này sử dụng camera OV2640 được tích hợp sẵn trong module ESP32-CAM. Đây là dòng cảm biến ảnh CMOS có độ phân giải 2.0 MP, hỗ trợ chụp ảnh định dạng JPEG, cho phép ghi lại hình ảnh rõ nét với dung lượng nhỏ gọn. Camera giao tiếp trực tiếp với vi điều khiển 3 (ESP32-CAM) thông qua bus nội bộ chuyên dụng, sử dụng giao thức cấu hình SCCB và truyền dữ liệu ảnh qua bus DVP.

Quá trình hoạt động của khói nhận diện diễn ra sau khi người dùng đã được xác thực qua thẻ từ và vân tay. Lúc này, ESP32-CAM nhận tín hiệu điều khiển từ vi điều khiển 2, tiến hành kích hoạt camera, chụp ảnh và lưu trữ vào thẻ nhớ microSD. Ảnh sẽ được đặt tên theo mã UID và thời gian điểm danh để dễ truy xuất. Tùy theo cấu hình, ảnh này cũng có thể được đồng bộ lên Google Sheets hoặc các nền tảng cloud khác thông qua Wi-Fi.

Khả năng xử lý ảnh của camera OV2640 đáp ứng tốt yêu cầu thời gian thực của hệ thống, với độ trễ trung bình khi chụp và lưu ảnh chỉ từ 500–800 ms. Độ rõ nét ở mức SVGA (800×600) đủ để quan sát khuôn mặt người dùng. Việc sử dụng module tích hợp như ESP32-CAM giúp tối giản thiết kế phần cứng, giảm số lượng linh kiện và rút ngắn thời gian triển khai hệ thống.

Tóm lại, khói nhận diện giúp tăng cường tính minh bạch và trực quan trong quá trình chấm công, đồng thời mở ra khả năng mở rộng cho các hệ thống sử dụng AI nhận diện khuôn mặt về sau.

3.3.9. Thiết kế khối mạng

Khối mạng có nhiệm vụ kết nối hệ thống với môi trường Internet để truyền dữ liệu điểm danh lên nền tảng lưu trữ từ xa. Trong hệ thống này, khói mạng được tích hợp trực tiếp trên module ESP32-CAM, thông qua kết nối Wi-Fi nội bộ do chính vi điều khiển ESP32 cung cấp. Nhờ đó, hệ thống có thể hoạt động độc lập, không cần đến gateway hoặc thiết bị phát trung gian.

Việc thiết kế phần cứng cho khói mạng gần như không yêu cầu thêm linh kiện ngoài, bởi ESP32-CAM đã tích hợp sẵn bộ thu phát Wi-Fi chuẩn 802.11 b/g/n. Điều quan trọng trong thiết kế là đảm bảo môi trường truyền thông ổn định, cấp nguồn đủ dòng và có cấu hình phần mềm phù hợp để kết nối đúng với mạng nội bộ được thiết lập.

Khi hệ thống hoạt động, ESP32-CAM sẽ tự động kết nối vào Wi-Fi, sau đó thực hiện gửi dữ liệu điểm danh gồm UID thẻ, mã nhân viên, thời gian, trạng thái xác thực và đường dẫn ảnh (nếu có) lên Google Sheets hoặc server đám mây thông qua giao thức HTTP hoặc dịch vụ trung gian như Google Apps Script.

Để đảm bảo kết nối ổn định, cần lưu ý:

- Cấp nguồn đủ dòng (≥ 300 mA) cho ESP32-CAM khi Wi-Fi hoạt động liên tục
- Đảm bảo cường độ tín hiệu Wi-Fi mạnh và không bị nhiễu trong khu vực lắp đặt

- Kiểm soát lỗi truyền (timeout, retry) để không làm gián đoạn quá trình điểm danh

Tổng thể, khôi mạng đóng vai trò là kênh liên kết giữa hệ thống nhúng và người quản trị, giúp dữ liệu luôn được cập nhật theo thời gian thực, đồng thời tạo điều kiện cho việc truy xuất, theo dõi và thống kê lịch sử điểm danh từ xa một cách linh hoạt.

3.3.10. Thiết kế khôi Google Sheets

Khôi Google Sheets là nơi lưu trữ dữ liệu điểm danh trên nền tảng đám mây, đóng vai trò như một bảng điều khiển dữ liệu - data dashboard để người quản trị theo dõi lịch sử điểm danh từ xa, theo thời gian thực. Dữ liệu được ghi nhận gồm: ngày/tháng/năm, thời gian điểm danh, ID, họ và tên nhân viên.

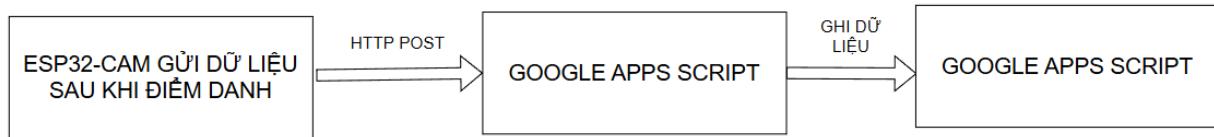
Thiết kế khôi này cần đảm bảo:

- Thứ nhất, lưu trữ dữ liệu từ hệ thống nhúng (ESP32-CAM) một cách tự động, chính xác, không cần thao tác thủ công.
- Thứ hai, dữ liệu được tổ chức khoa học, dễ tra cứu, lọc, thống kê.
- Thứ ba, hoạt động ổn định và bảo mật khi truyền thông tin từ thiết bị vật lý lên nền tảng cloud.
- Thứ tư, cho phép người quản trị có thể truy cập, giám sát, hoặc tải về dữ liệu bất kỳ lúc nào, từ mọi thiết bị có kết nối Internet.

Dữ liệu từ hệ thống không được gửi trực tiếp đến Google Sheets mà thông qua một kiến trúc trung gian sử dụng:

- Thứ nhất, Google Apps Script – một đoạn mã chạy trong nền Google, dùng để tiếp nhận dữ liệu từ thiết bị và ghi vào bảng tính.
- Tiếp theo, HTTP Request từ ESP32-CAM – gửi dữ liệu dưới dạng JSON hoặc form-data qua phương thức POST đến endpoint của Apps Script.

- Cuối cùng, Google Sheets – là bảng tính được liên kết sẵn với Apps Script, tự động nhận dữ liệu và lưu từng dòng vào các cột tương ứng.



Hình 3.21. Luồng xử lý dữ liệu truyền từ ESP32-CAM đến Google Sheets

Để khôi Google Sheets hoạt động tron tru, cần thực hiện các bước sau:

- Bước 1, Tạo bảng tính trên Google Sheets với các cột sau: Date, Time, ID, Fullname

Hình 3.22. Các cột được tạo trong Google Sheets

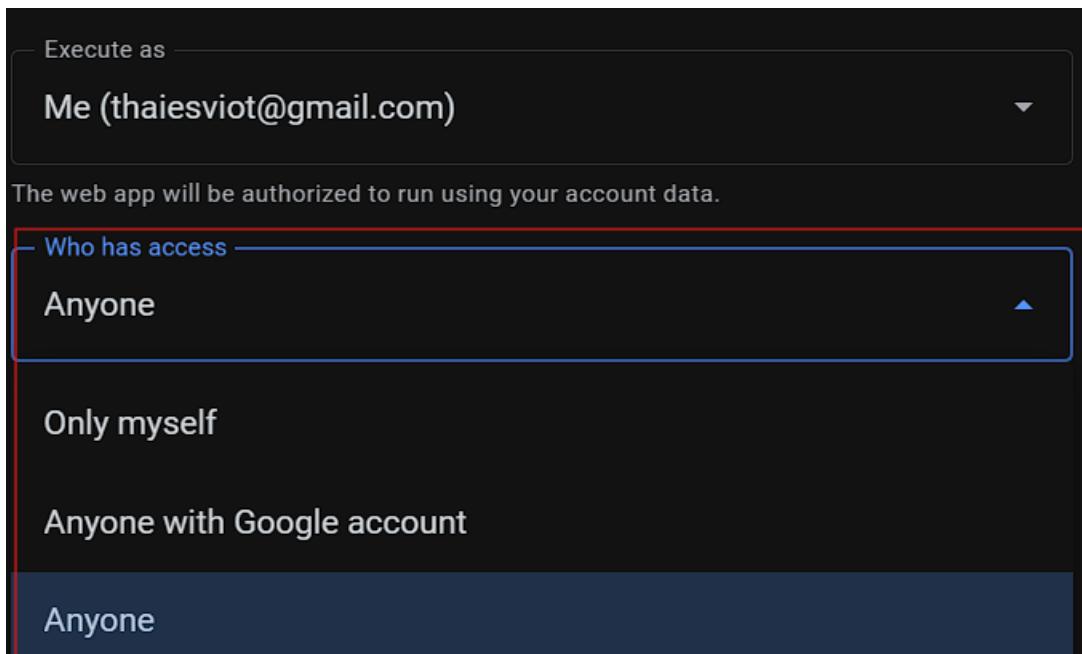
- Bước 2, trên thanh công cụ của Google Sheets, chọn vào Extensions và chọn Apps Script. Và viết đoạn code JavaScript như bên dưới để xử lý POST từ ESP32-CAM.

```

1  function doPost(e) {
2      var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
3      var data = JSON.parse(e.postData.contents);
4
5      sheet.appendRow([
6          data.date,
7          data.time,
8          data.id,
9          data.name
10     ]);
11
12     return ContentService.createTextOutput("Success");
13 }
```

Hình 3.23. Đoạn code JavaScript xử lý POST từ ESP32-CAM

- Bước 3, trong Google Apps Script chọn Deploy, chọn tiếp Manage deployments, chọn vào biểu tượng cây viết và lướt xuống thấy dòng như trong ảnh thì đổi thành Anyone:



Hình 3.24. Cách Deploy để ESP32-CAM có thể POST dữ liệu

Tóm lại, khói Google Sheets là điểm kết thúc của chuỗi xử lý trong hệ thống chấm công, nơi dữ liệu từ thiết bị vật lý được lưu trữ trên nền tảng số hóa, giúp quản trị viên dễ dàng theo dõi, giám sát và phân tích lịch sử điểm danh một cách linh hoạt và hiệu quả.

3.3.11. Thiết kế khói giao diện

Khói giao diện là thành phần giúp người quản trị tương tác trực tiếp với hệ thống, cung cấp một môi trường trực quan để thực hiện các thao tác quản lý nhân sự, điều khiển các chức năng đăng ký/xóa vân tay, kiểm tra kết nối và theo dõi dữ liệu điểm danh theo thời gian thực.

Mục tiêu thiết kế khói này bao gồm:

- Thứ nhất, cung cấp giao diện đồ họa dễ sử dụng cho người quản trị, không yêu cầu hiểu biết chuyên sâu về hệ thống nhúng.
- Thứ hai, cho phép giao tiếp trực tiếp với vi điều khiển 1 (STM32) thông qua cổng COM – UART, để thực hiện các lệnh điều khiển.
- Thứ ba, hỗ trợ các chức năng quản trị dữ liệu như thêm, sửa, xóa nhân viên, xem danh sách hiện có.
- Thứ tư, hỗ trợ truy cập nhanh đến bảng Google Sheets lưu trữ dữ liệu điểm danh.

Khói giao diện được xây dựng bằng ngôn ngữ Python, sử dụng thư viện PyQt5 để thiết kế UI và PySerial để giao tiếp với cổng COM. Giao diện hoạt động trên máy tính Windows, kết nối với hệ thống thông qua cổng USB-UART.

Các thành phần chính trong khói này gồm:

- PyQt5
 - Tạo giao diện đồ họa (GUI), gồm các thành phần như combobox chọn COM, nút kết nối, màn hình log dữ liệu, ô nhập lệnh.

- Thiết kế hiện đại, dễ nhìn và trực quan.
- PySerial
 - Giao tiếp UART qua cổng COM để truyền lệnh từ người quản trị đến STM32 (vi điều khiển 1).
 - Nhận phản hồi và log dữ liệu thời gian thực từ hệ thống.
- Module xử lý chức năng (backend)
 - Đảm nhiệm xử lý sự kiện khi người dùng nhấn nút (kết nối, gửi lệnh, mở Google Sheet...).
 - Đóng vai trò cầu nối giữa GUI và các vi điều khiển qua UART.



Hình 3.25. Giao diện quản trị viên

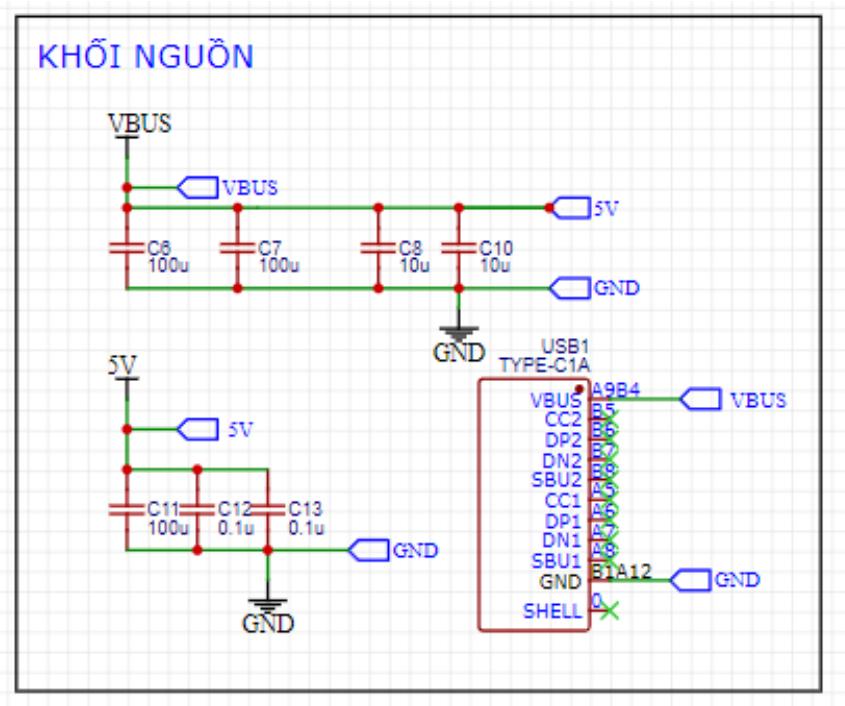
Giao diện UI được thiết kế thân thiện và đáp ứng đầy đủ các yêu cầu vận hành của người quản trị. Một số tính năng nổi bật bao gồm:

- Chọn và kết nối cổng COM
 - Tự động dò tìm các cổng COM đang kết nối với máy tính.
 - Hiển thị log tình trạng kết nối hoặc thông báo lỗi.
- Giao tiếp 2 chiều với vi điều khiển
 - Gửi lệnh UART đến STM32 để thực hiện các chức năng như:
 - Đăng ký/xóa dấu vân tay
 - Gửi dữ liệu nhân viên xuống vi điều khiển
 - Nhận dữ liệu phản hồi và log lại trong khung hiển thị.
- Quản lý nhân viên
 - Cho phép thêm, sửa, xóa thông tin nhân viên từ máy tính.
 - Cập nhật vào hệ thống thông qua serial hoặc lưu vào file.
- Truy cập Google Sheets
 - Nút bấm mở nhanh bảng tính Google Sheets chứa dữ liệu điểm danh để xem, in hoặc xuất báo cáo.
- Hiển thị thông báo thời gian thực
 - Các phản hồi từ hệ thống (như trạng thái xác thực, lỗi, hướng dẫn) được hiển thị trực tiếp lên giao diện để người quản trị nắm bắt.

3.3.12. Thiết kế khối nguồn

Khối nguồn là một trong những thành phần quan trọng nhất trong bất kỳ hệ thống nhúng nào. Trong hệ thống chấm công này, nhiệm vụ của khối nguồn là:

- Thứ nhất, cung cấp đầy đủ và ổn định các mức điện áp cần thiết (5V) cho toàn bộ vi điều khiển.
- Thứ hai, đảm bảo công suất đầu ra phù hợp với mức tiêu thụ cao nhất của hệ thống, đặc biệt khi Wi-Fi, camera và thẻ nhớ hoạt động đồng thời.
- Thứ ba, tổ chức lọc nguồn tốt để tránh nhiễu gây ảnh hưởng đến tín hiệu UART, SPI, hay ảnh đầu ra từ camera.
- Thứ tư, đảm bảo an toàn, dễ thay thế, dễ kiểm soát trong quá trình vận hành và bảo trì.



Hình 3.26. Sơ đồ nguyên lý của khối nguồn

Hệ thống sử dụng đầu vào là nguồn 5V từ cổng USB Type-C, thông qua đường VBUS. Từ đây, nguồn được lọc và phân phối đến các phần khác.

Các tụ lọc (C7, C8, C10...) được bố trí gần chân VBUS và các nhánh nguồn nhằm:
Ôn định dòng khi có dao động từ tải (ESP32-CAM, Wi-Fi...)

Sau khi lọc, nhánh 5V sẽ được dùng cấp cho các chân 5V trên 3 vi điều khiển, từ đó vi điều khiển sẽ lấy nguồn từ đây hạ áp xuống 3.3V và đưa hệ thống hoạt động.

Tụ phân cực 100 μ F kết hợp tụ gồm 0.1 μ F được đặt song song nhằm lọc tốt cả nhiễu tần số thấp và cao, đảm bảo ổn định điện áp ở cả tải nặng và nhẹ.

Để chọn được nguồn có công suất phù hợp để cấp ổn định cho toàn bộ hệ thống, ta sẽ đi tính toán công suất từ các tải tiêu hao.

Bảng 3.7. Công suất tiêu thụ của toàn hệ thống

Thiết bị	Điện áp hoạt động	Dòng tiêu thụ (ước tính)	Công suất (ước tính) (mW)
STM32 Blue pill	5V	~ 35 mA	175
ESP32 Devkit	5V	~ 160 mA	800
ESP32-CAM	5V	~ 310 mA	1550
OLED SH1106	3.3V	~ 25 mA	82.5
RFID RC522	3.3V	~ 30 mA	99
Module AS608	3.3V	~ 100 mA	330
Tổng			3036.5

Công suất tiêu thụ của toàn hệ thống khoảng 3036,5mW = 3.0365W

Nguồn chính sử dụng cho hệ thống là adapter 5V–3A, tương đương công suất 15W, hoàn toàn đáp ứng ổn định cho toàn bộ thiết bị kể cả khi tải hoạt động đồng thời ở mức cao nhất. Trong quá trình phát triển và kiểm thử, hệ thống cũng có thể được cấp nguồn tạm thời thông qua cổng USB của máy tính, cụ thể là qua ESP32 DevKit, với mức cấp điện 5V–1.5A (tương đương 7.5W). Mức công suất này vẫn đủ đáp ứng cho quá trình debug và chạy thử các chức năng chính mà không gây sụt áp hay mất ổn định hệ thống.

Dựa theo định luật Joule-Lenz và tiêu chuẩn IPC-2221, dây dẫn nguồn phải được chọn đúng tiêu diện để đảm bảo không bị sụt áp và quá nhiệt.

- Đường 5V cấp cho ESP32-CAM (dòng cao) → rộng từ 0.8 – 1.2 mm, vì dòng lên tới 300–400 mA.
- Đường 3.3V cho ngoại vi → nên từ 0.6 mm trở lên, đảm bảo cấp đủ dòng cho nhiều module chạy song song.
- Các đường GND được làm plane GND phủ hai mặt để chống nhiễu cho hệ thống.

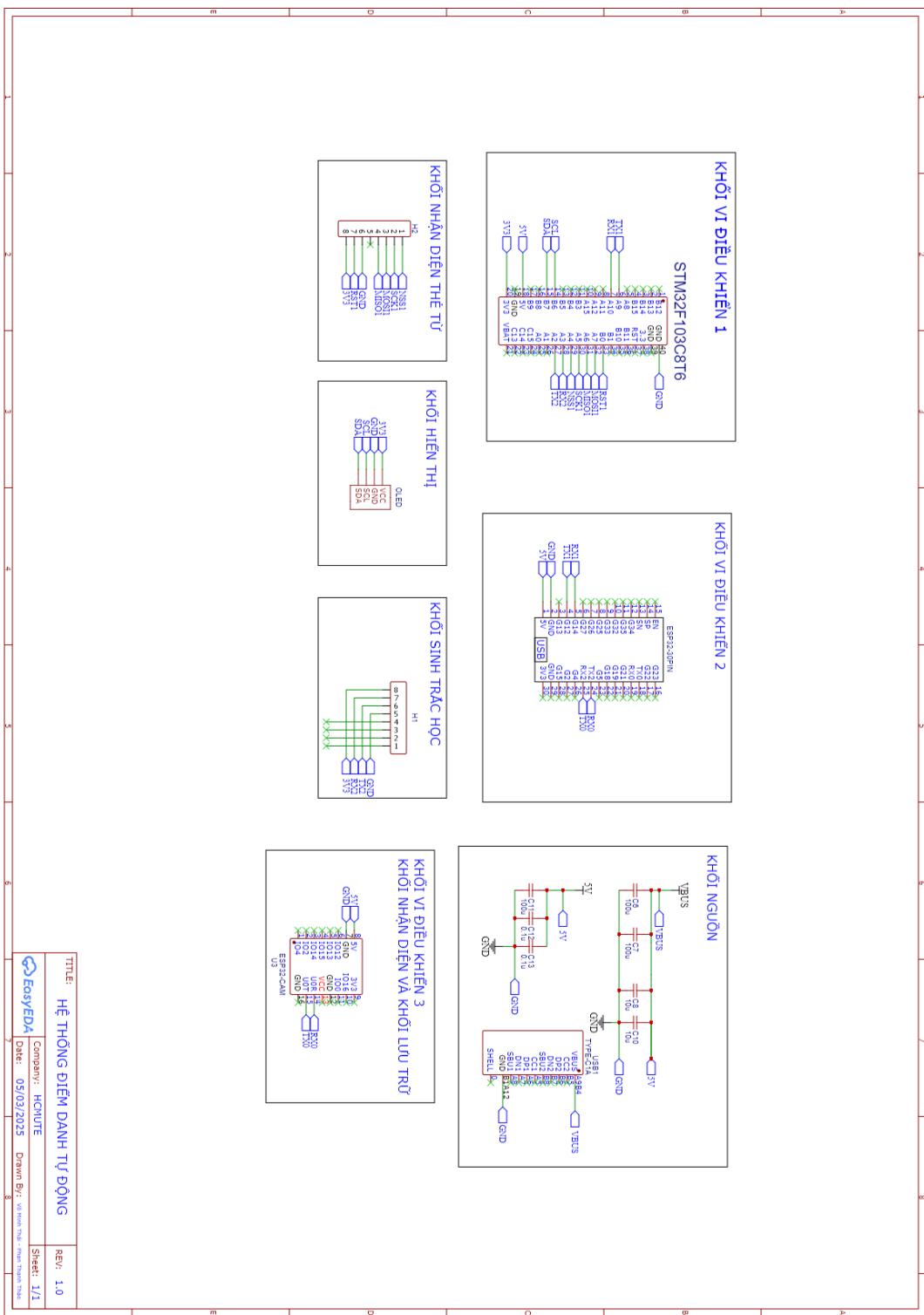
Lưu ý khi đi dây:

- Luôn đặt tụ lọc gần IC tiêu thụ hoặc gần chân cấp nguồn.
- Tránh để đường nguồn chạy song song hoặc quá gần đường xung (clock, SPI).

Khối nguồn là yếu tố nền tảng giúp toàn hệ thống hoạt động ổn định, bền vững. Việc thiết kế đúng kỹ thuật từ cách chọn nguồn, bố trí tụ lọc, đến đi dây PCB sẽ giúp giảm lỗi không mong muốn, tăng tuổi thọ phần cứng và tối ưu trải nghiệm sử dụng hệ thống trong thời gian dài.

3.3.13. Sơ đồ nguyên lý toàn hệ thống

Sau khi hoàn tất quá trình thiết kế từng khối chức năng riêng biệt như vi điều khiển, cảm biến, giao diện, khói lưu trữ, khói mạng và đặc biệt là khói nguồn – thành phần đảm bảo cấp điện ổn định cho toàn bộ hệ thống – bước tiếp theo là tổng hợp lại toàn bộ các thành phần vào một sơ đồ nguyên lý hoàn chỉnh. Sơ đồ này thể hiện mối liên kết giữa các khối chức năng, các đường giao tiếp chính (UART, SPI, I2C), cũng như cấu trúc cấp nguồn và logic điều khiển toàn diện của hệ thống chấm công tự động.



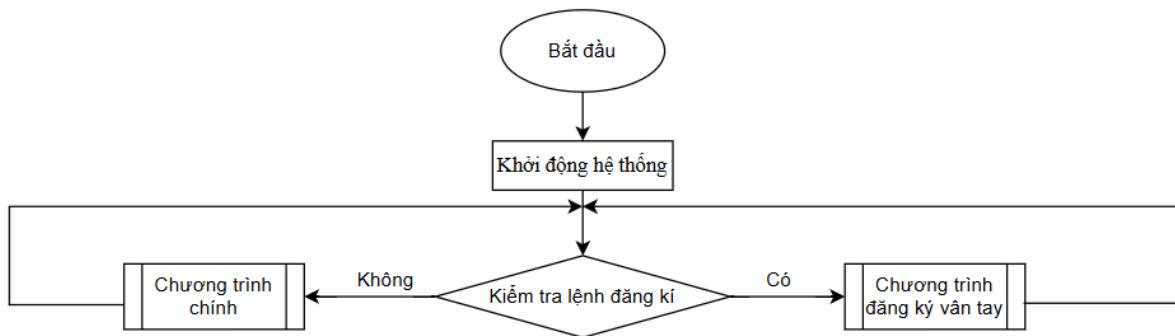
Hình 3.27. Sơ đồ nguyên lý toàn mạch

3.4. THIẾT KẾ CHƯƠNG TRÌNH CỦA HỆ THỐNG

3.4.1. Chương trình chính của hệ thống

Chương trình chính của hệ thống được thiết kế để vận hành theo cấu trúc vòng lặp liên tục, đảm bảo khả năng theo dõi và phản hồi tức thời với các sự kiện xảy ra trong quá trình tương tác với người dùng. Từ thời điểm khởi động, hệ thống thực hiện tuần tự các bước kiểm tra, nhận diện và xác thực thông tin dựa trên dữ liệu từ thẻ RFID và cảm biến vân tay, đồng thời cập nhật trạng thái lên màn hình hiển thị. Mọi tình huống xử lý đều được kiểm soát chặt chẽ thông qua luồng dữ liệu một chiều hoặc hai chiều giữa các thành phần chức năng.

Để minh họa rõ hơn cho logic điều khiển này, lưu đồ thuật toán dưới đây trình bày tổng quan quá trình xử lý của hệ thống trong suốt vòng đời hoạt động:



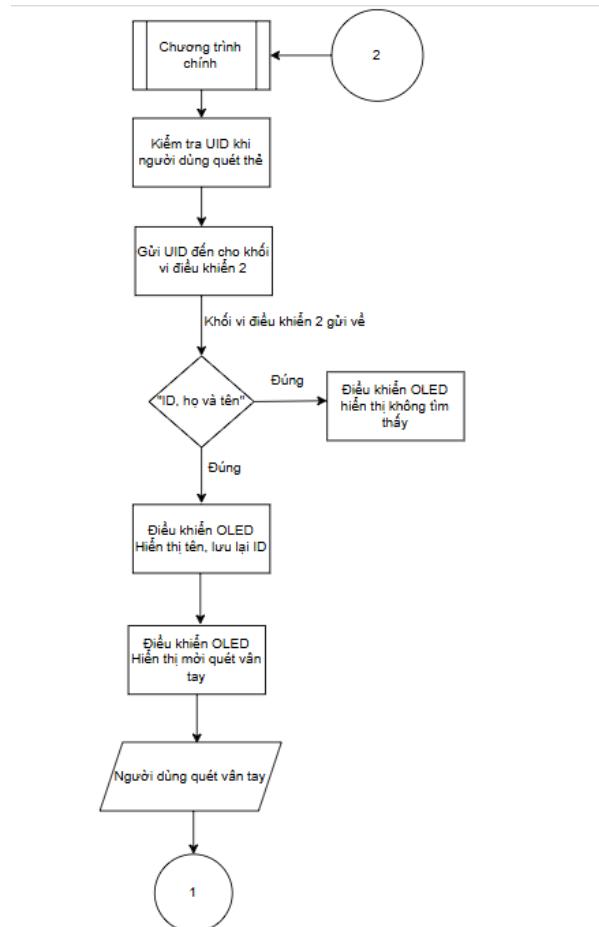
Hình 3.28. Lưu đồ thuật toán tổng quát của hệ thống

Giải thích ý nghĩa của lưu đồ:

- **Bắt đầu:** Chương trình được khởi động khi hệ thống cấp nguồn hoặc được reset.
- **Khởi động hệ thống:** Vi điều khiển thực hiện các tác vụ khởi tạo phần cứng như cấu hình UART, SPI, I2C, khởi tạo các module như OLED, RFID, cảm biến vân tay.

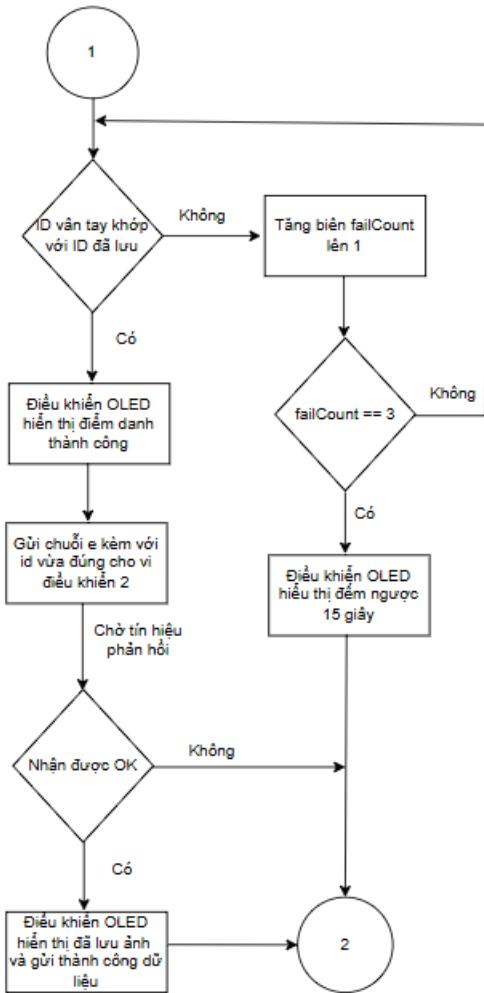
- Kiểm tra lệnh đăng ký: Ở mỗi chu kỳ vòng lặp chính, vi điều khiển kiểm tra xem có lệnh yêu cầu đăng ký vân tay được gửi từ giao diện quản trị qua UART không.
 - Nếu có, chương trình chuyển sang chế độ đăng ký vân tay, cho phép người dùng thêm vân tay mới vào bộ nhớ của AS608.
 - Nếu không, hệ thống quay lại và tiếp tục thực thi chương trình chính, bao gồm nhận diện thẻ RFID, kiểm tra vân tay, hiển thị kết quả, và truyền thông tin cho các khối còn lại.

Lưu đồ thuật toán dành cho chương trình chính:



Hình 3.29. Lưu đồ chương trình chính

Lưu đồ chương trình chính tiếp theo:



Hình 3.30. Lưu đồ chương trình chính (tiếp theo)

Giải thích chi tiết lưu đồ thuật toán

Khi hệ thống bắt đầu hoạt động, màn hình OLED sẽ hiển thị thông báo hướng dẫn người dùng đưa thẻ RFID lại gần để quét. Người dùng tiến hành đưa thẻ lại gần đầu đọc RFID RC522, tại đây UID của thẻ sẽ được đọc và truyền về vi điều khiển STM32. STM32 sau đó gửi UID này thông qua giao tiếp UART đến ESP32 Devkit. Tiếp theo, ESP32 Devkit truyền UID này đến ESP32-CAM.

Tại ESP32-CAM, hệ thống truy cập vào thẻ nhớ và mở file users.txt để thực hiện đối chiếu UID. Việc so sánh được thực hiện từng dòng trong file. Nếu tìm thấy dòng chứa UID trùng khớp, ESP32-CAM sẽ trả về một chuỗi chứa ID và tên nhân viên tương ứng cho ESP32 Devkit, và ESP32 Devkit sẽ chuyển tiếp chuỗi này về STM32. STM32 sẽ xử lý chuỗi bằng cách tách thông tin ID và tên: phần tên được hiển thị lên màn hình OLED, còn ID sẽ được lưu lại để sử dụng cho bước xác thực vân tay tiếp theo.

Trong trường hợp UID không trùng khớp với bất kỳ dòng nào trong users.txt, ESP32-CAM sẽ trả về chuỗi "ERROR" cho ESP32 Devkit, và chuỗi này tiếp tục được gửi về STM32. Khi nhận được "ERROR", STM32 sẽ hiển thị lên OLED thông báo rằng không tìm thấy UID, đồng thời quay lại bước đầu tiên và yêu cầu người dùng quét lại thẻ RFID.

Sau khi đã nhận được tên nhân viên hợp lệ từ bước quét thẻ, hệ thống chuyển sang bước quét vân tay. Màn hình OLED lúc này hiển thị hướng dẫn người dùng đặt ngón tay lên cảm biến. Người dùng đưa ngón tay vào cảm biến vân tay, và hệ thống kiểm tra xem vân tay này đã được đăng ký và lưu trữ trong bộ nhớ của module hay chưa. Nếu vân tay trùng khớp, hệ thống xác nhận điểm danh thành công và OLED hiển thị thông báo tương ứng. Đồng thời, STM32 sẽ gửi một chuỗi dữ liệu có định dạng "e<ID>", trong đó <ID> là ID của vân tay vừa được xác nhận, gửi lên ESP32 Devkit để xử lý.

Dữ liệu tiếp tục được xử lý tương tự như luồng quét thẻ: truyền lên ESP32-CAM, truy xuất thông tin tương ứng, rồi phản hồi ngược lại STM32. Nếu STM32 nhận lại được chuỗi "OK" thì hệ thống sẽ hiển thị lên OLED thông báo đã lưu ảnh và gửi dữ liệu thành công, sau đó quay trở lại bước ban đầu chờ người dùng quét thẻ RFID mới.

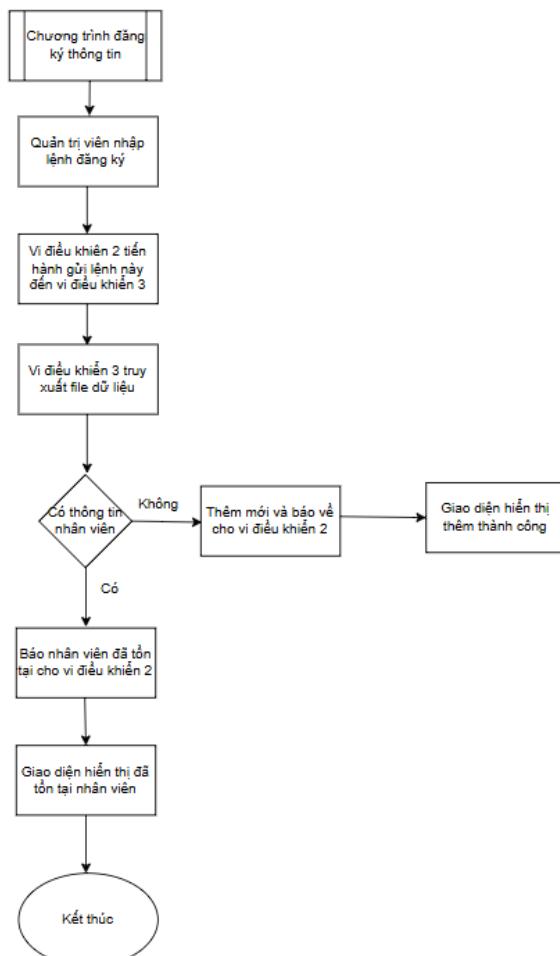
Ngược lại, nếu người dùng quét sai vân tay, hệ thống sẽ cho phép quét lại tối đa ba lần. Nếu đến lần quét thứ tư vẫn không hợp lệ, OLED sẽ hiển thị thông báo đếm ngược 15 giây và sau khi kết thúc thời gian chờ, hệ thống sẽ quay lại bước đầu tiên, yêu cầu quét lại thẻ RFID.

3.4.2. Chương trình đăng ký thông tin nhân viên mới

Chương trình đăng ký thông tin nhân viên mới được thiết kế nhằm mở rộng cơ sở dữ liệu người dùng của hệ thống, thông qua việc liên kết giữa ID định danh và dữ liệu sinh trắc học vân tay. Quá trình này đảm bảo rằng mỗi cá nhân chỉ được gán một mã ID duy nhất đi kèm với mẫu vân tay đã xác thực, phục vụ cho việc kiểm soát truy cập hoặc điểm danh tự động về sau.

Toàn bộ thao tác đăng ký được thực hiện một cách tương tác, thông qua giao diện quản trị viên. Quy trình bắt đầu bằng việc nhập lệnh: ax, họ và tên nhân viên mới, 4 byte mã UID, sau đó tiến hành xác nhận vân tay theo hai bước nhằm đảm bảo tính chính xác.

Để minh họa rõ ràng logic vận hành của chương trình, hình dưới đây trình bày lưu đồ giải thuật tổng quát của quá trình đăng ký nhân viên mới. Lưu đồ này thể hiện đầy đủ các bước kiểm tra, xử lý, phản hồi và điều kiện rẽ nhánh được thực hiện trong suốt chu trình hoạt động.



Hình 3.31. Lưu đồ thuật toán đăng ký thông tin nhân viên mới

Giải thích chi tiết ý nghĩa lưu đồ:

Lưu đồ trên mô tả quy trình xử lý thông tin đăng ký nhân viên mới trong hệ thống, thể hiện luồng dữ liệu và logic phối hợp giữa các vi điều khiển để đảm bảo việc thêm mới thông tin người dùng diễn ra chính xác và có kiểm soát.

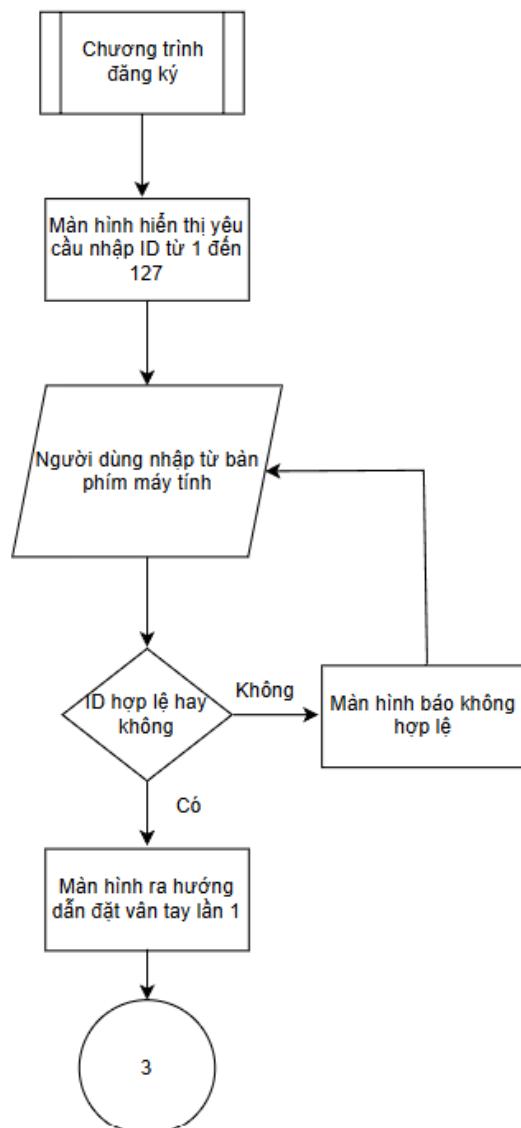
Quy trình bắt đầu khi quản trị viên nhập lệnh đăng ký thông qua giao diện điều khiển. Lệnh này được vi điều khiển 2 tiếp nhận và truyền đến vi điều khiển 3, nơi thực hiện thao tác truy xuất dữ liệu từ tệp tin chứa danh sách nhân viên hiện tại.

Sau khi dữ liệu được đọc, hệ thống kiểm tra xem thông tin nhân viên cần đăng ký đã tồn tại hay chưa. Nếu chưa có, hệ thống sẽ tiến hành thêm mới thông tin nhân viên, gửi phản hồi xác nhận lại cho vi điều khiển 2, đồng thời giao diện hiển thị thông báo thêm thành công. Ngược lại, nếu đã tồn tại, vi điều khiển 2 sẽ báo lại rằng nhân viên đã có trong hệ thống và cập nhật giao diện hiển thị tương ứng.

Quy trình kết thúc ngay sau khi hệ thống đưa ra phản hồi phù hợp với tình huống, đảm bảo rằng thông tin nhân viên được quản lý nhất quán và tránh tình trạng trùng lặp dữ liệu.

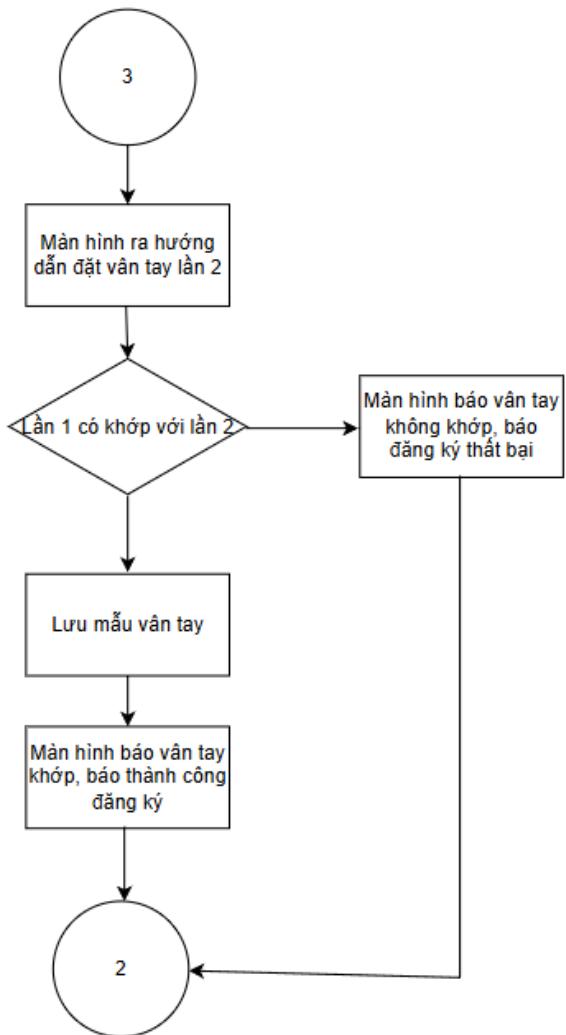
3.4.3. Chương trình đăng ký vân tay

Chương trình đăng ký vân tay là một thành phần quan trọng trong hệ thống nhận diện và kiểm soát truy cập, nhằm đảm bảo mỗi người dùng được gán một mã định danh sinh trắc học duy nhất. Mục tiêu của chương trình là thu thập dữ liệu vân tay thông qua cảm biến, xử lý và lưu trữ mẫu vân tay vào bộ nhớ nội bộ của module, đồng thời gán kèm với một ID định danh để phục vụ cho việc xác thực sau này. Quá trình đăng ký yêu cầu sự tương tác trực tiếp từ người dùng và tuân thủ các bước kiểm tra nhằm đảm bảo chất lượng dữ liệu đầu vào, tránh các lỗi nhận dạng trong tương lai. Phần mềm được xây dựng theo cấu trúc tuần tự, có khả năng phát hiện trạng thái của cảm biến, phản hồi các thao tác người dùng và hiển thị hướng dẫn trực quan qua màn hình để đảm bảo quá trình đăng ký diễn ra thuận lợi và chính xác. Dưới đây là lưu đồ thuật toán của phần mềm:



Hình 3.32. Lưu đồ chương trình đăng ký vân tay

Lưu đồ thuật toán tiếp theo:



Hình 3.33. Lưu đồ chương trình đăng ký (tiếp theo)

Dưới đây là giải thích chi tiết về lưu đồ thuật toán này:

Khi kết nối hệ thống với máy tính thông qua cổng USB Type-C trên ESP32 Devkit, người vận hành có thể thực hiện thao tác đăng ký vân tay cho nhân viên mới. Việc này được kích hoạt bằng cách nhập ký tự r hoặc R vào khung giao tiếp serial và gửi lệnh. Ngay sau

đó, ESP32 Devkit sẽ truyền lệnh này xuống cho vi điều khiển STM32 để khởi động quy trình đăng ký vân tay.

Khi bước vào chương trình đăng ký, màn hình OLED sẽ hiển thị hướng dẫn yêu cầu người dùng nhập ID nhân viên trong khoảng từ 1 đến 127. Nếu ID được nhập hợp lệ, hệ thống sẽ chuyển sang bước xác thực vân tay hai lần liên tiếp theo hướng dẫn hiển thị. Người dùng đặt ngón tay lên cảm biến hai lần; nếu hai lần quét cho kết quả trùng khớp, ảnh vân tay sẽ được lưu vào bộ nhớ và hệ thống thông báo đăng ký thành công trên màn hình OLED.

Ngược lại, nếu hai lần quét không trùng nhau, hệ thống sẽ báo lỗi đăng ký thất bại và tự động quay về chương trình chính đã trình bày trước đó. Trong trường hợp người dùng nhập sai ID (ngoài phạm vi cho phép), hệ thống sẽ hiển thị thông báo lỗi ID không hợp lệ” và yêu cầu nhập lại. Sau đó, quá trình đăng ký được tiếp tục theo đúng trình tự như trên.

CHƯƠNG 4. KẾT QUẢ VÀ ĐÁNH GIÁ

Sau khi hoàn tất quá trình thiết kế phần cứng và lập trình phần mềm theo định hướng đã đề ra trong chương 3, hệ thống chấm công tự động được tiến hành lắp ráp, triển khai và kiểm thử thực tế nhằm đánh giá toàn diện tính hiệu quả và mức độ đáp ứng các yêu cầu kỹ thuật, chức năng đã đặt ra trong đề tài. Việc đưa hệ thống vào vận hành thử không chỉ giúp kiểm chứng tính đúng đắn trong thiết kế mạch, kết nối giữa các vi điều khiển và cảm biến, mà còn đóng vai trò quan trọng trong việc kiểm tra độ ổn định của phần mềm điều khiển, độ trễ truyền thông, khả năng tương thích giữa các khối chức năng và mức tiêu thụ năng lượng tổng thể.

Chương này tập trung trình bày các kết quả thu được thông qua quá trình kiểm tra vận hành toàn hệ thống, bao gồm cả phần kết quả phần cứng (lắp ráp, cấp nguồn, kết nối cảm biến, hoạt động của vi điều khiển) và kết quả phần mềm (hiển thị trạng thái, nhận diện thông tin, phản hồi UART, gửi dữ liệu lên Google Sheets...). Ngoài ra, chương cũng cung cấp các hình ảnh minh họa trực quan cho từng giai đoạn chạy thử hệ thống, từ giao tiếp cảm biến đến chụp ảnh và cập nhật dữ liệu, qua đó giúp đánh giá được khả năng ứng dụng thực tiễn của hệ thống cũng như mức độ hoàn thiện của giải pháp thiết kế.

4.1. KẾT QUẢ THIẾT KẾ PHẦN CỨNG

Hệ thống được triển khai thực tế dưới dạng một mô hình tích hợp, trong đó toàn bộ chức năng được tổ chức thành ba khối xử lý chính, tương ứng với ba vi điều khiển độc lập, liên kết chặt chẽ với nhau thông qua giao tiếp nối tiếp. Cụ thể:

- Vi điều khiển STM32F103C8T6 đóng vai trò như một trung tâm tiếp nhận dữ liệu từ các thiết bị đầu vào. Nó chịu trách nhiệm điều khiển và giao tiếp với các cảm biến sinh trắc học như máy quét vân tay AS608, đầu đọc thẻ RFID RC522 và màn hình OLED SH1106 để hiển thị trạng thái hệ thống. Đây là nơi xử lý bước xác thực đầu

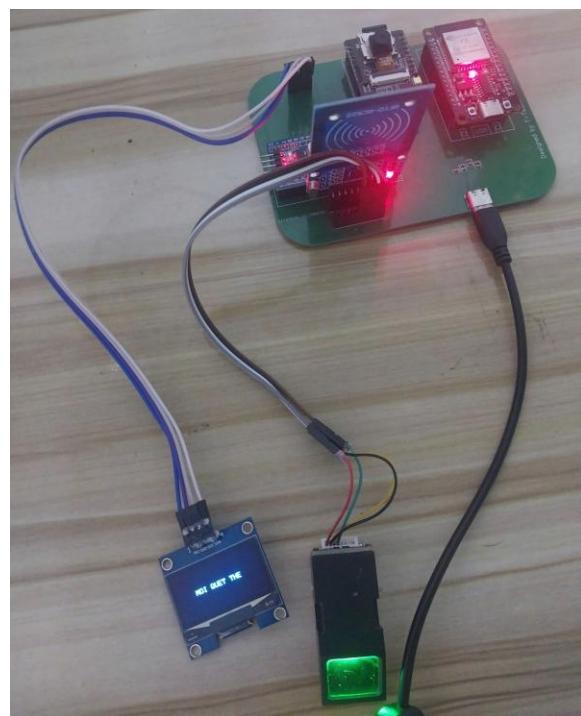
tiên và bước xác thực sinh trắc học của người dùng trước khi gửi dữ liệu đến hệ thống trung tâm.

- ESP32 DevKit hoạt động như một cầu nối giao tiếp trung gian giữa STM32 và ESP32-CAM. Nó tiếp nhận thông tin từ STM32 qua UART, xử lý lệnh điều khiển cấp cao, đồng thời điều phối dữ liệu, chuyển tiếp các tín hiệu cần chụp ảnh đến ESP32-CAM. Ngoài ra, ESP32 DevKit cũng đảm nhiệm vai trò quản lý đồng bộ giữa các tiến trình xác thực và ghi nhận thông tin.
- ESP32-CAM là khối xử lý cuối cùng trong chuỗi luồng dữ liệu. Sau khi nhận lệnh từ ESP32 DevKit, nó thực hiện chụp ảnh người dùng, lưu trữ hình ảnh vào thẻ nhớ SD, và sau đó đẩy thông tin đã xác thực (gồm ảnh, mã UID, thời gian điểm danh) lên Google Sheets thông qua kết nối Wi-Fi. Vì điều khiển này vừa đảm nhận tác vụ xử lý hình ảnh vừa thực hiện các chức năng truyền thông mạng, đòi hỏi cấu hình phần mềm tối ưu và lập trình đa tiến trình hiệu quả.

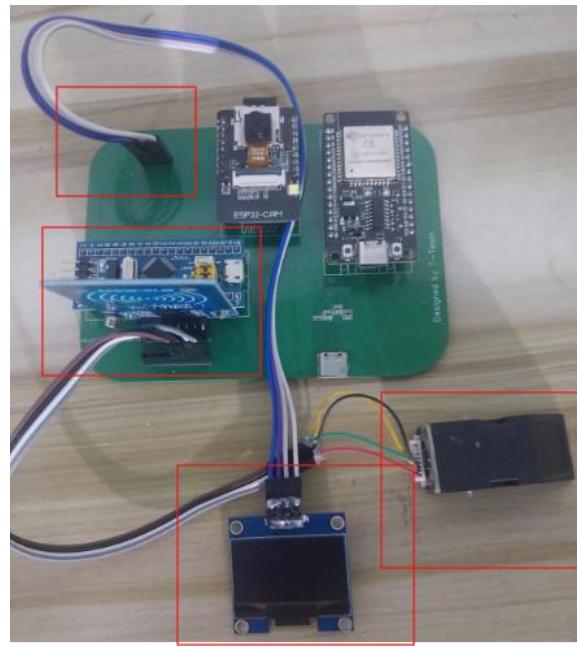
Việc tách riêng ba vi điều khiển theo vai trò chức năng không chỉ giúp tăng khả năng mở rộng, phân tách trách nhiệm, mà còn giảm tải tính toán cho từng thiết bị, đảm bảo hoạt động ổn định và thời gian phản hồi nhanh trong điều kiện thực tế. Cấu trúc này cũng tạo tiền đề để phát triển các chức năng nâng cao như quản lý tập trung, giám sát từ xa, hoặc tích hợp thêm các cơ chế bảo mật trong tương lai.



Hình 4.1. Mô hình tổng thể của hệ thống

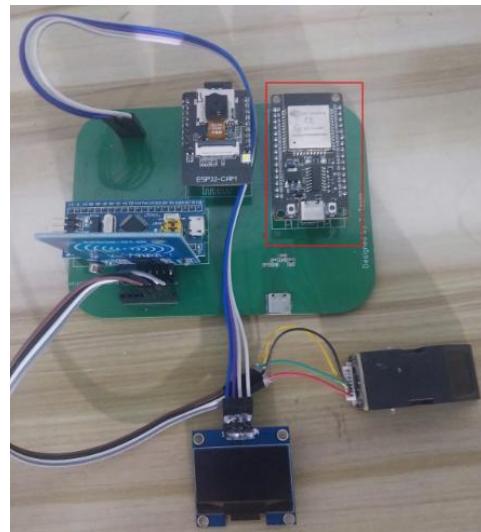


Hình 4.2. Mô hình hệ thống sau khi cấp nguồn



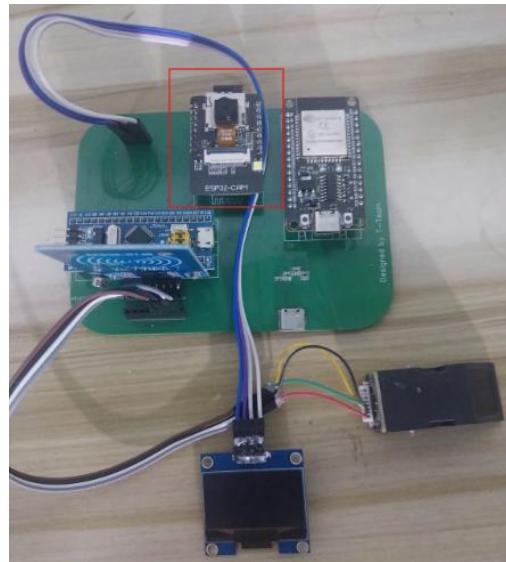
Hình 4.3. Mô hình của khối vi điều khiển 1 và các thiết bị

Sau khi thi công mạch xong, nhìn chung khối vi điều khiển 1 đã đáp ứng được những yêu cầu đã đề ra ở chương 3, các khối thiết bị được đặt gần STM32 Blue pill, tất cả đều dùng header để đảm bảo linh hoạt và dễ bảo trì, có gắp tiết kiệm không gian mạch nhất có thể.



Hình 4.4. Mô hình của khối vi điều khiển 2 với các thành phần xung quanh

Hình trên cho thấy module ESP32 Devkit đã đặt với cổng USB quay xuống đã đáp ứng được nhu cầu dễ cắm và dễ debug, module được đặt gần khói vi điều khiển 3 để tiết kiệm độ dài dây tín hiệu tuy nhiên còn gặp hạn chế do đặt khá xa khói vi điều khiển 1 làm cho dây tín hiệu đi dài ra, dù vậy việc này vẫn phải thực hiện vì mong muốn đặt ESP32-CAM có góc chụp hợp lý nhất có thể.



Hình 4.5. Mô hình của khói vi điều khiển 3 và các phần tử xung quanh

Hình bên trên cho thấy khói vi điều khiển 3 đã đặt vị trí và các yêu cầu được đưa ra ở chương 3, khói này được đặt sao cho camera hướng thẳng về mặt của nhân viên đang thực hiện điểm danh.



Hình 4.6. Mô hình của khối nguồn

Nhìn hình bên ta có thể thấy được cổng USB và các tụ lọc được bố trí khá hợp lý, cổng USB được đặt sao cho có thể dễ dàng dùng dây từ adapter cắm vào và đẹp mắt.

4.2. KẾT QUẢ THIẾT KẾ PHẦN MỀM

Sau khi lắp ráp hoàn thiện phần cứng, chương trình nhúng được nạp vào các vi điều khiển. Hệ thống phần mềm được phân chia thành 3 luồng xử lý chính, tương ứng với 3 vai trò:

- STM32 chạy firmware xử lý nhận dạng vân tay, đọc mã UID và điều khiển OLED
- ESP32 DevKit làm cầu nối truyền dữ liệu giữa STM32 và ESP32-CAM
- ESP32-CAM tiếp nhận dữ liệu, chụp ảnh và gửi thông tin lên Google Sheets

4.2.1. Kết quả giao tiếp UART trên giao diện quản trị và màn hình OLED



Hình 4.7. Kết nối UART với máy tính

Sau khi mở giao diện và thực hiện kết nối dây USB vào cổng USB Type C của ESP32 Devkit thì đã kết nối được với máy tính ở COM4. Tiếp theo nhóm sẽ thực hiện tương tác lệnh từ khung nhập dữ liệu để xem các kết quả:

Đầu tiên, tiến hành đưa thẻ RFID vào đầu đọc (thẻ đúng) và thu được phản hồi từ ESP32-CAM như sau:

```
[INFO] Đã kết nối với COM4 @ 115200 baud.  
>> Đã gửi UID từ STM32 tới ESP32-CAM: 7C,9F,E0,00  
>> ESP32-CAM phản hồi hợp lệ: 1,vo minh thai
```

Hình 4.8. Phản hồi qua UART khi đưa thẻ đúng vào



Hình 4.9. Màn hình OLED hiển thị tên nhân viên

Tiếp theo, tiến hành đưa thẻ chưa được đăng ký vào thì ta thu được kết quả:

- >> Đã gửi UID từ STM32 tới ESP32-CAM: EF,CE,E0,1E
 - >> ESP32-CAM phản hồi hợp lệ: OK
-

Hình 4.10. Phản hồi qua UART khi đưa thẻ chưa được đăng ký vào

Khi thẻ chưa được đăng ký thì nó trả về OK tức là một chuỗi khác với chuỗi mong muốn là id và họ tên nhân viên nên màn hình OLED sẽ phản hồi là không tìm thấy



Hình 4.11. Màn hình OLED hiển thị không tìm thấy

Nhìn chung, giao tiếp UART trên mô hình đã đáp ứng đúng yêu cầu, đồng thời đã gửi và dữ liệu được từ phần cứng lên giao diện quản trị trên máy tính. Tiếp theo ta sẽ đi khảo sát chiều ngược lại:



Hình 4.12. Phản hồi của phần cứng khi nhận được tín hiệu qua UART

Hình trên cho ta thấy được phản hồi của mô hình khi ta dùng khung chat gửi tín hiệu g1 yêu cầu truy xuất thông tin của người dùng có id 1 về cho máy tính hiển thị lên giao diện.

4.2.2. Kết quả ghi nhận ảnh và dữ liệu

Sau khi hoàn tất quá trình điểm danh, ảnh sẽ được lưu vào thẻ nhớ với định dạng của thư mục là: năm – tháng – ngày như ảnh đính kèm dưới đây:

📁	2025-05-23	5/23/2025 1:47 PM	File folder
📁	2025-05-24	5/24/2025 12:07 AM	File folder
📁	2025-05-28	5/28/2025 6:12 PM	File folder
📁	2025-05-29	5/29/2025 12:59 AM	File folder
📁	2025-06-01	6/1/2025 11:15 PM	File folder
📁	2025-06-03	6/3/2025 11:45 PM	File folder
📁	2025-06-04	6/4/2025 1:02 AM	File folder
📁	Screenshots	5/16/2025 12:27 AM	File folder
📄	users.txt	5/29/2025 1:58 AM	Text Document 1 KB

Hình 4.13. Tên các thư mục ảnh nhân viên sẽ được lưu vào

Name	Date modified	Type	Size
📄 ID0_07-36-31.jpg	6/4/2025 7:36 AM	JPG File	45 KB
📄 ID0_07-37-25.jpg	6/4/2025 7:37 AM	JPG File	154 KB
📄 ID1_01-02-18.jpg	6/4/2025 1:02 AM	JPG File	39 KB
📄 ID1_07-53-50.jpg	6/4/2025 7:53 AM	JPG File	149 KB

Hình 4.14. Tên của các file ảnh

Sau khi ảnh được chụp xong thì dữ liệu sẽ được đẩy lên Google Sheets với ngày, giờ, id và họ tên tương ứng của nhân viên vừa điểm danh thành công.

Date	Time	ID	Fullname
03/06/2025	23:45:37	1	vo minh thai
03/06/2025	23:46:05	4	phan thanh thao
04/06/2025	1:02:18	1	vo minh thai
04/06/2025	7:36:31	0	Unknown
04/06/2025	7:53:50	1	vo minh thai

Hình 4.15. Dữ liệu được đẩy lên Google Sheets sau khi điểm danh xong

4.2.3. Giao diện quản trị viên

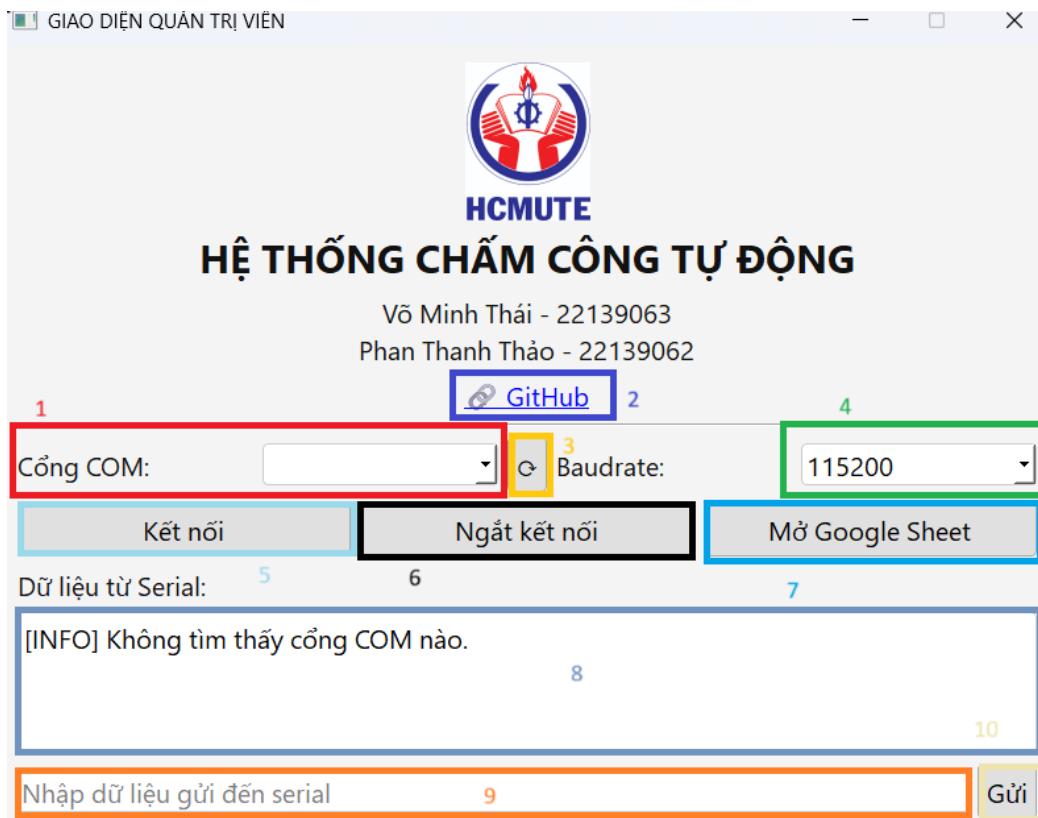
Phần mềm giao diện quản trị được lập trình bằng Python với PyQt5 và PySerial, cho phép người dùng:

- Thêm/xóa/sửa thông tin nhân viên
 - Gửi lệnh đăng ký vân tay đến STM32
 - Mở nhanh Google Sheets để kiểm tra lịch sử

Dưới đây là giao diện và chức năng của từng khối trong giao diện:

- Khối 1: Dùng để chọn cổng COM của máy tính, ban đầu nếu cảm sẵn thì nó mặc định tự chọn COM đầu tiên.
 - Khối 2: là đường link dẫn đến trang github chi tiết dự án của nhóm
 - Khối 3: là nút dùng để load vào các cổng COM mới
 - Khối 4: dùng để chọn baudrate cho giao tiếp
 - Khối 5: là nút dùng để kết nối với COM sau khi chọn COM xong
 - Khối 6: dùng để ngắt kết nối với COM
 - Khối 7: khi người quản trị nhấn vào nó sẽ dẫn đến trang Google Sheets để xem thông tin

- Khối 8: tại đây người quản trị có thể xem và quản lý các thông tin được truyền qua UART giữa các thiết bị trong hệ thống và giữa máy tính với hệ thống
- Khối 9: là khung nhập liệu các lệnh giao tiếp với hệ thống
- Khối 10: sau khi nhập liệu lệnh xong thì nhấn vào đây để gửi dữ liệu đi



Hình 4.16. Giao diện quản trị hoàn chỉnh và sơ đồ chức năng của nó

4.3. ĐÁNH GIÁ

Sau quá trình xây dựng, triển khai và kiểm thử hệ thống chấm công tự động, nhóm thực hiện tiến hành đánh giá toàn diện về tính đúng đắn, hiệu quả hoạt động, mức độ ổn định và khả năng mở rộng của hệ thống. Kết quả đánh giá được tổng hợp dựa trên quá trình vận hành thực tế, khả năng xử lý dữ liệu, tốc độ phản hồi, độ chính xác của các cảm biến và tính tương thích phần mềm giữa các khối vi điều khiển.

4.3.1. Ưu điểm

- Tính chính xác cao: Hệ thống sử dụng kết hợp hai bước xác thực (RFID + vân tay), giúp tăng độ an toàn và hạn chế giả mạo trong điểm danh.
- Phân tách chức năng hợp lý: Việc sử dụng 3 vi điều khiển tương ứng với 3 tầng xử lý (nhận dạng – truyền dữ liệu – xử lý ảnh & đồng bộ cloud) giúp hệ thống ổn định, dễ bảo trì và mở rộng.
- Tốc độ phản hồi nhanh: Tổng thời gian từ lúc quét thẻ đến khi ảnh được chụp và dữ liệu được cập nhật lên Google Sheets chỉ mất khoảng 1–2 giây.
- Giao diện trực quan: Người dùng và quản trị viên dễ dàng thao tác thông qua màn hình OLED, phần mềm giao diện trên máy tính và Google Sheets.
- Tích hợp lưu trữ đám mây: Hệ thống cập nhật dữ liệu thời gian thực lên Google Sheets, phù hợp với nhu cầu truy xuất từ xa, lưu trữ và báo cáo.

4.3.2. Hạn chế

- ESP32-CAM không debug dễ dàng: Do ESP32-CAM không có chip nạp/debug riêng, việc nạp chương trình và kiểm thử mất thời gian hơn nếu không chuẩn bị mạch riêng.
- Chưa có cơ chế xử lý lỗi mạng: Trong trường hợp Wi-Fi mất kết nối, ảnh và dữ liệu sẽ chưa được gửi đi. Hệ thống hiện chưa có cơ chế lưu đệm hoặc gửi lại khi mạng ổn định.
- Giao tiếp UART còn chưa được xử lý triệt để: Trong quá trình vận hành thực tế, hệ thống xuất hiện tình trạng nhiều UART hoặc lệnh bị chòng lấn khi gửi từ phần mềm giao diện xuống STM32. Cụ thể, khi gửi lệnh đăng ký vân tay, màn hình OLED đôi khi vẫn tiếp tục nhận và xử lý các lệnh không cần thiết, gây ra hiện tượng hiển thị sai hoặc tự động chuyển vào chế độ đăng ký mà không có yêu cầu rõ ràng. Ngoài ra,

sau khi hoàn tất một phiên đăng ký vân tay, hệ thống đôi khi không tự động trở về chế độ chờ ban đầu, buộc phải nhấn reset trên STM32 để đưa hệ thống về trạng thái ổn định. Nguyên nhân chủ yếu đến từ thiếu cơ chế khóa trạng thái giao tiếp UART hoặc chưa xử lý triệt để lệnh nền khi giao tiếp giữa các khói diễn ra liên tục.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. KẾT LUẬN

Trong bối cảnh chuyển đổi số và nhu cầu hiện đại hóa quản lý nhân sự tại các cơ quan, doanh nghiệp ngày càng tăng cao, hệ thống chấm công tự động kết hợp các công nghệ định danh như RFID, sinh trắc học vân tay, xử lý ảnh và lưu trữ đám mây đã trở thành một giải pháp tiềm năng, thay thế cho các phương pháp truyền thống còn nhiều bất cập. Đề tài “Xây dựng hệ thống chấm công tự động ứng dụng trong quản lý nhân sự” được thực hiện với mục tiêu hiện thực hóa mô hình này bằng cách kết hợp nhiều vi điều khiển và cảm biến hiện đại trong một kiến trúc nhúng hoàn chỉnh.

Hệ thống được thiết kế theo hướng modular hóa từng khối chức năng, trong đó sử dụng ba vi điều khiển chính:

- STM32F103C8T6 đóng vai trò thu thập dữ liệu từ các thiết bị đầu vào như thẻ RFID và cảm biến vân tay AS608, đồng thời xử lý logic điều khiển và hiển thị.
- ESP32 DevKit hoạt động như một trung tâm trung gian, truyền nhận dữ liệu từ STM32, điều phối tác vụ giữa các khối và giao tiếp với ESP32-CAM qua UART.
- ESP32-CAM thực hiện chụp ảnh khuôn mặt nhân viên, lưu trữ cục bộ trên thẻ nhớ microSD và cập nhật thông tin điểm danh (gồm ảnh, UID, thời gian...) lên Google Sheets thông qua kết nối Wi-Fi.

Phần mềm giao diện quản trị được xây dựng bằng PyQt5 và PySerial, cho phép người dùng trực tiếp thao tác với hệ thống: gửi lệnh điều khiển, đăng ký/xóa nhân viên, quản lý dữ liệu và truy cập nhanh vào bảng điểm danh trực tuyến. Giao tiếp UART giữa các khối được thiết kế theo dạng lệnh điều khiển đơn giản, nhưng hiệu quả, đảm bảo khả năng mở rộng và kiểm thử dễ dàng.

Thông qua quá trình triển khai thực tế, hệ thống đã chứng minh được các đặc điểm sau:

- Hoạt động ổn định, thời gian phản hồi nhanh, thông tin điểm danh được xử lý và cập nhật đầy đủ.
- Tính xác thực cao, nhờ áp dụng xác minh 2 lớp: thẻ từ RFID và vân tay sinh trắc học.
- Tích hợp thành công giữa phần cứng nhúng và nền tảng đám mây, giúp truy xuất dữ liệu mọi lúc mọi nơi.
- Giao diện trực quan, thân thiện với người dùng, giúp quản trị viên dễ dàng thao tác và theo dõi.

Tuy nhiên, hệ thống cũng còn tồn tại một số hạn chế nhất định, điển hình như:

- Giao tiếp UART còn có thể bị nhiễu khi có nhiều lệnh liên tục.
- Phần xử lý lỗi mạng và lưu đệm ảnh chưa hoàn chỉnh khi mất Wi-Fi.
- Hệ thống chưa tối ưu năng lượng, chưa phù hợp với các ứng dụng chạy bằng pin.

Dù vậy, với kiến trúc phần cứng được tổ chức bài bản, thuật toán phần mềm rõ ràng và khả năng tương tác cao, hệ thống đã hoàn toàn đáp ứng được mục tiêu ban đầu của đề tài. Kết quả triển khai đạt được là cơ sở vững chắc để tiếp tục phát triển, mở rộng trong tương lai.

5.2. Hướng phát triển

Để nâng cao hơn nữa tính ứng dụng và hiệu quả của hệ thống trong các môi trường thực tế như doanh nghiệp, nhà máy, trường học..., một số hướng phát triển khả thi bao gồm:

- Tối ưu hóa truyền thông UART giữa các vi điều khiển bằng cách bổ sung cơ chế hàng đợi lệnh (command queue), giao thức kiểm tra checksum hoặc xác thực trạng thái giao tiếp để tránh nhiễu, lỗi treo hệ thống sau khi gửi lệnh.
- Tăng cường tính ổn định mạng thông qua tích hợp giao thức MQTT hoặc HTTPS, cho phép kết nối đến server riêng hoặc nền tảng IoT chuyên dụng thay vì chỉ sử dụng Google Sheets.
- Tích hợp thêm các phương thức xác thực nâng cao như nhận diện khuôn mặt AI (Face Recognition), mã OTP, hoặc quét mã QR, nhằm tạo thêm các lớp bảo mật phù hợp với các môi trường đòi hỏi kiểm soát nghiêm ngặt.
- Thiết kế mạch in PCB chuyên nghiệp và tích hợp vỏ hộp nhằm chuyển từ dạng mô hình thử nghiệm sang sản phẩm thực tế, hướng đến khả năng thương mại hóa.
- Tối ưu tiêu thụ năng lượng bằng cách áp dụng các chế độ deep sleep, cut-off camera và cảm biến khi không cần thiết, từ đó có thể vận hành hệ thống bằng nguồn dự phòng như pin Li-ion hoặc nguồn ngoài qua năng lượng mặt trời.
- Tăng tính tương tác người dùng bằng việc thiết kế ứng dụng điện thoại đi kèm, sử dụng Bluetooth hoặc Wi-Fi để quản lý, xem nhanh lịch sử điểm danh, hoặc điều khiển hệ thống từ xa.

TÀI LIỆU THAM KHẢO

- [1] Klaus Schwab, *The Fourth Industrial Revolution*, World Economic Forum, 2016.
- [2] IBOM, “Quản lý nhân sự chấm công tính lương – Thực trạng và Giải pháp,” *IBOM*, Jun. 14, 2023. [Online]. Available: <https://ibom.vn/tin-giai-phap/quan-ly-nhan-su-cham-cong-tinh-luong-thuc-trang-va-giai-phap.html>. [Truy cập: May-30- 2025].
- [3] M. M. Hassan et al., "A survey on smart attendance monitoring systems using IoT," *Journal of Network and Computer Applications*, vol. 150, pp. 102–120, 2020.
- [4] A. S. N. Habib et al., "IoT based smart attendance system using fingerprint and RFID", *Proceedings of the 2019 International Conference on Smart Systems and Inventive Technology*, pp. 872–877, 2019.
- [5] Deloitte, *Digital HR: Revolution not evolution*, Human Capital Trends, 2017.
- [6] S. Gary, “Automating attendance management in human resources: A design science approach using computer vision and facial recognition,” *ScienceDirect*, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2667096824000429>. [Accessed: 30-May-2025].
- [7] J. Kim, “Automated attendance management systems: Systematic literature review,” *ResearchGate*, 2022. [Online]. Available: https://www.researchgate.net/publication/358178607_Automated_attendance_management_systems_systematic_literature_review. [Accessed: 30-May-2025].
- [8] “Top 12 phần mềm chấm công tốt nhất 2025 dành cho doanh nghiệp,” *Base.vn*, 14-Mar-2025. [Online]. Available: <https://base.vn/blog/phan-mem-cham-cong/>. [Truy cập: 30-May-2025].
- [9] “Phần mềm Quản lý Nhân sự nhiều doanh nghiệp sử dụng nhất,” *Tanca.io*. [Online]. Available: <https://tanca.io/>. [Truy cập: 30-May-2025].

- [10] “Cách sử dụng máy chấm công để tự động hóa quản lý nhân sự tại công ty,” *Dogoo.vn*. [Online]. Available: <https://dogoo.vn/cach-su-dung-may-cham-cong-de-tu-dong-hoa-quan-ly-nhan-su-tai-cong-ty/>. [Truy cập: 30-May-2025].
- [11] Google, “Google Sheets,” [Online]. Available: <https://workspace.google.com/products/sheets/> . [Accessed: May 30, 2025].
- [12] M. Rees, *Google Workspace User Guide*, O'Reilly Media, 2022.
- [13] Spencer Lanoue, “Guide: How to Use Google Sheets for Data Analysis,” *The Bricks*, 20-Feb-2023.[Online]. Available: <https://www.thebricks.com/resources/guide-how-to-use-google-sheets-for-data-analysis>. [Accessed: May 30, 2025].
- [14] Google Developers, “Apps Script Overview,” *Google Developers*, <https://developers.google.com/apps-script/overview>, [Truy cập: ngày 30 tháng 5 năm 2025].
- [15] Google Developers, “Extending Google Sheets,” *Google Developers*, <https://developers.google.com/apps-script/guides/sheets>, [Truy cập: ngày 30 tháng 5 năm 2025].
- [16] N. H. Phuoc, “Kiến thức cơ bản về giao tiếp UART,” *Điện Tử Việt*, Apr. 25, 2021. [Online]. Available: <https://dientuviet.com/kien-thuc-co-ban-ve-giao-tiep-uart/>. [Accessed: May 31, 2025].
- [17] DMDT, “Chuẩn giao tiếp SPI và cách kết nối,” *Điện Tử Cơ Bản*, Jun. 21, 2022. [Online]. Available: <https://dammedientu.vn/chuan-giao-tiep-spi-va-cach-ket-noi>. [Accessed: May 31, 2025].
- [18] DMDT, “Giới thiệu chuẩn giao tiếp I2C,” *Điện Tử Cơ Bản*, Jun. 21, 2022. [Online]. Available: <https://dammedientu.vn/gioi-thieu-chuan-giao-tiep-i2c>. [Accessed: May 31, 2025].

- [19] D. Harkins, “Wi-Fi Easy Connect™: Simple and secure onboarding for IoT,” The Beacon Blog, Wi-Fi Alliance, Feb. 13, 2023. [Online]. Available: <https://www.wi-fi.org/beacon/dan-harkins/wi-fi-easy-connect-simple-and-secure-onboarding-for-iot>. [Accessed: May 31, 2025].
- [20] STMicroelectronics, “STM32 Microcontroller Series,” ST.com. [Online]. Available: <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>. [Accessed: May 30, 2025].
- [21] STMicroelectronics, “STM32F103x8/xB Datasheet,” ST.com. [Online]. Available: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>. [Accessed: May 30, 2025].
- [22] STMicroelectronics, “STM32CubeIDE,” ST.com. [Online]. Available: <https://www.st.com/en/development-tools/stm32cubeide.html>. [Accessed: May 30, 2025].
- [23] STM32duino Wiki, “Blue Pill,” [Online]. Available: <https://belchip.by/sitedocs/31025.pdf>. [Accessed: 31-May-2025].
- [24] M. Kozey, “Datasheet Esp32 Devkit V1 30 Pin,” *Schematic and Diagram Full List*, Apr. 30, 2024. [Online]. Available: <https://partdiagramdromecanoelq.z19.web.core.windows.net/datasheet-esp32-devkit-v1-30-pin.html>. [Accessed: May 31, 2025].
- [25] DFRobot, “ESP32-CAM Development Board,” *DFRobot*, Aug. 15, 2019. [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/DFR0602_Web.pdf. [Accessed: May 31, 2025].
- [26] Adafruit Industries, “Adafruit Optical Fingerprint Sensor,” Adafruit Learning System, Jun. 2014. [Online]. Available: <https://cdn-adafruit.s3.amazonaws.com/assets/docs/OpticalFingerprintSensor.pdf>.

learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf. [Accessed: May 31, 2025].

[27] A. M. Shojaei, “Interfacing FPM10A/AS608 Optical Fingerprint Reader Sensor Module with Arduino,” Electropeak, [Online]. Available: <https://electropeak.com/learn/interfacing-fpm10a-as608-optical-fingerprint-reader-sensor-module-with-arduino/>. [Accessed: May 31, 2025].

[28] ArduinoKIT.vn, “RFID là gì? Hướng dẫn sử dụng Module RFID RC522 với Arduino,” ArduinoKIT.vn, May 27, 2023. [Online]. Available: <https://arduinokit.vn/rfid-la-gi-huong-dan-su-dung-module-rfid-rc522-voi-arduino/>. [Accessed: May 31, 2025].

[29] UNIVERSAL-SOLDER Electronics Ltd., “CANADUINO® OLED 128x64 Pixel – I2C – 1.3 inch – SSD1306 or SH1106,” UNIVERSAL-SOLDER Electronics Ltd., [Online]. Available: https://universal-solder.ca/downloads/canaduino_oled_1.3_i2c.pdf. [Accessed: May 31, 2025].