

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN VIỄN THÔNG**

-----❖-----



## **ĐỒ ÁN 2**

# **THIẾT KẾ HỆ THỐNG KHÓA TỪ RFID TÍCH HỢP THEO DÕI HÀNH VI MỞ CỬA (THE DESIGN OF AN RFID DOOR LOCK SYSTEM INTEGRATED WITH BEHAVIOR TRACKING FOR DOOR ACCESS)**

**CÁN BỘ HƯỚNG DẪN**  
Thầy. Th.s Nguyễn Thanh Tuấn

**SINH VIÊN THỰC HIỆN:**  
**Bùi Minh Đức; MSSV: 2012983**  
Ngành: Điện tử - Viễn Thông – Khóa: K20

**• Học kỳ 232 •  
Năm học 2023-2024**



Số: \_\_\_\_\_/BKĐT  
Khoa: **Điện – Điện tử**  
Bộ Môn: **Viễn Thông**

## NHIỆM VỤ ĐỒ ÁN 2

- Họ và tên: Bùi Minh Đức MSSV: 2012983
- Ngành: Điện – Điện tử Chuyên ngành: Kỹ thuật Điện tử - Truyền thông
- Đề tài: Thiết kế hệ thống khóa từ RFID tích hợp theo dõi hành vi mở cửa  
(The design of an RFID door lock system integrated with behavior tracking for door access).
- Nhiệm vụ:
  - Thiết kế mạch PCB hệ thống khóa cửa RFID tích hợp theo dõi hành vi mở cửa.
  - Đo đạc thực nghiệm ... Mô phỏng hệ thống ...
  - Phân tích và so sánh kết quả ...
- Ngày giao nhiệm vụ đồ án: 20/02/2024
- Ngày hoàn thành nhiệm vụ: 27/05/2024
- Họ và tên người hướng dẫn: **Phản hướng dẫn**  
Th.s Nguyễn Thanh Tuấn,  
BM Viễn Thông, Khoa Điện – Điện Tử **100%**  
Nội dung và yêu cầu ĐA2 đã được thông qua Bộ Môn.

TP.HCM, ngày \_\_ tháng \_\_ năm 2024

**CHỦ NHIỆM BỘ MÔN**

**NGƯỜI HƯỚNG DẪN CHÍNH**

PGS. TS. Hà Hoàng Kha

Th.s Nguyễn Thanh Tuấn

### PHẦN DÀNH CHO KHOA, BỘ MÔN:

Người duyệt (chấm sơ bộ): .....

Đơn vị: .....

Ngày bảo vệ : .....

Điểm tổng kết:.....

Nơi lưu trữ luận văn: .....



## LỜI CẢM ƠN

Trước tiên em xin cảm ơn sâu sắc đến cha mẹ đã cổ vũ, động viên, hỗ trợ về tinh thần cũng như tạo mọi điều kiện thuận lợi cho em hoàn thành tốt tiểu luận này.

Em xin chân thành cảm ơn thầy Nguyễn Thanh Tuấn đã luôn quan tâm chỉ dạy, theo dõi, giúp đỡ tận tình trong suốt khoảng thời gian em thực hiện tiểu luận.

Và hơn hết, em xin bày tỏ lòng biết ơn trân trọng đến quý thầy cô Đại học Quốc gia TP.HCM – trường Đại học Bách Khoa đã tận tình chỉ dạy, truyền đạt những kiến thức quý báu cho em trong suốt thời gian theo học vừa qua để em có đủ kiến thức, điều kiện để thực hiện đề tài này.

Đồng cảm ơn đến các tác giả trong các quyển sách báo, internet, anh chị đi trước đã tìm tòi, nghiên cứu đúc kết kinh nghiệm làm tài liệu để em có thể tham khảo trong quá trình thực hiện đề tài.

Sau cùng tôi xin cảm ơn các bạn bè khoa Điện – Điện tử khóa K2020 đã tận tình giúp đỡ, hỗ trợ cho tôi trong quá trình thực hiện tiểu luận này.

TP. HCM, ngày 28, tháng 5 năm 2024

Bùi Minh Đức

## LỜI CAM ĐOAN

Tôi tên Bùi Minh Đức là sinh viên chuyên ngành Kỹ thuật Điện tử - Truyền thông, khóa 2020, tại Đại học Quốc gia thành phố Hồ Chí Minh – Trường Đại học Bách Khoa. Tôi xin cam đoan những nội dung sau đều là sự thật:

- (i) Công trình nghiên cứu này hoàn toàn do chính tôi thực hiện.
- (ii) Các tài liệu và trích dẫn trong luận văn này được tham khảo từ các nguồn thực tế, có uy tín và độ chính xác cao.
- (iii) Các số liệu và kết quả của công trình này được tôi tự thực hiện một cách độc lập và trung thực.

TP. HCM, ngày 28, tháng 5 năm 2024

Bùi Minh Đức

## TÓM TẮT LUẬN VĂN

Đề tài này tập trung vào việc xây dựng hệ thống khóa từ RFID tích hợp theo dõi hành vi đóng/mở cửa. Mục tiêu là đóng/ mở khóa sử dụng mật khẩu/thẻ từ RF. Để thực hiện điều này, em sử dụng STM32F103C8T6 làm vi điều khiển trung tâm điều khiển các ngoại vi.

Các thiết bị sẽ được kết nối giao tiếp với vi điều khiển qua các chân với phương thức lập trình nhúng phổ thông để điều khiển. Đề tài này xây dựng theo hướng có thể phát triển Đồ án tốt nghiệp thành một hệ thống IoT “Smart Home” tích hợp webserver quản lý và ứng dụng android theo dõi hoạt động.

Qua việc triển khai, đề tài hy vọng giúp người dùng linh hoạt trong việc sử dụng khóa từ, cũng là tăng tính bảo mật, mang lại tiện lợi và linh hoạt – là một phần không thể thiếu trong hệ thống Smart Home.

## **ABSTRACT**

This topic focuses on building an integrated RFID magnetic lock system that tracks door opening/closing behavior. The goal is to close/unlock using a password/RF card. To do this, I use STM32F103C8T6 as the central microcontroller to control the peripherals.

The devices will be connected to the microcontroller via pins with a common embedded programming method for control. This project is built in the direction of being able to develop the Graduation Project into an IoT "Smart Home" system integrating a management webserver and an Android application to monitor activities.

Through implementation, the project hopes to help users be flexible in using magnetic locks, also increase security, bring convenience and flexibility - an indispensable part of the Smart Home system.



## MỤC LỤC

LỜI CẢM ƠN.....	i
LỜI CAM ĐOAN.....	ii
TÓM TẮT LUẬN VĂN.....	iii
ABSTRACT.....	iv
DANH SÁCH HÌNH ẢNH.....	vii
<b>CHƯƠNG 1. GIỚI THIỆU .....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Phạm vi và phương pháp nghiên cứu .....	1
1.3 Các đóng góp của luận văn .....	2
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....</b>	<b>3</b>
2.1 Kit phát triển STM32F103C8T6 Blue Pill ARM Cortex-M3 .....	3
2.1.1 Giới thiệu .....	3
2.1.2 Thông số kỹ thuật.....	4
2.1.3 Cấu hình.....	5
2.2 Ngoại vi (ESP32 DEVKIT V1, RC522, LCD 1602, POWER, BUTTON).....	7
2.2.1 Esp32 DEVKIT V1 30pins .....	7
2.2.2 Module RFID RC522 NFC 13.56mhz .....	7
2.2.3 LCD 1602.....	11
2.2.4 Module I2C giao tiếp LCD1602 .....	14
2.2.5 Các ngoại vi khác.....	15
2.3 Phương thức truyền dữ liệu HTTP .....	16
2.4 Kết luận chương.....	20
<b>CHƯƠNG 3. THIẾT KẾ PHẦN CỨNG.....</b>	<b>22</b>
3.1 Yêu cầu thiết kế.....	22
3.2 Đặc tả hệ thống.....	22
3.3 Mô phỏng (Schematic Circuit).....	23
3.4 Mạch in (PCB Layout).....	23
3.5 Kết luận chương.....	25
<b>CHƯƠNG 4. THIẾT KẾ PHẦN MỀM.....</b>	<b>26</b>
4.1 Yêu cầu phần mềm.....	26
4.2 Lưu đồ giải thuật.....	26
4.3 Giao tiếp STM32F103C8T6 với ESP32 DEVKIT V1 truyền dữ liệu.....	27
4.3.1 STM32F103C8T6.....	27
4.3.2 ESP32.....	28

4.3.3	Google Apps Script API .....	29
<b>4.4</b>	<b>Kết luận chương.....</b>	<b>30</b>
<b>CHƯƠNG 5.</b>	<b>KẾT LUẬN.....</b>	<b>32</b>
<b>5.1</b>	<b>Hoàn thiện sản phẩm.....</b>	<b>32</b>
<b>5.2</b>	<b>Tóm tắt và kết luận chung .....</b>	<b>33</b>
5.2.1	Kết quả đạt được .....	33
5.2.2	Hạn chế .....	33
<b>5.3</b>	<b>Hướng phát triển.....</b>	<b>34</b>
<b>PHỤ LỤC A.....</b>	<b>.....</b>	<b>35</b>
<b>A.1</b>	<b>Phần cứng (hardware) .....</b>	<b>35</b>
<b>A.2</b>	<b>Phần mềm (software) .....</b>	<b>35</b>
<b>A.3</b>	<b>Kết quả (video) .....</b>	<b>35</b>
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>.....</b>	<b>36</b>

## DANH SÁCH HÌNH ẢNH

Hình 1: Kit STM32 Bluepill (STM32F103C8T6) .....	3
Hình 2: STM32 Bluepill GPIO Pin.....	4
Hình 3: ESP32 DEVKIT V1 30pins .....	7
Hình 4: giao tiếp UART giữa 2 MCU. ....	7
Hình 5: Module thẻ từ RFID RC522 13,56Mhz .....	8
Hình 6: RFID-RC522 pinout .....	9
Hình 7: LCD1602 pin out .....	11
Hình 8: Module I2C LCD 1602 .....	14
Hình 9: Khóa chốt điện từ LY-03 12VDC.....	16
Hình 10: Button input pull-up.....	16
Hình 11: Sơ đồ khối hệ thống .....	22
Hình 12: Sơ đồ nguyên lý .....	23
Hình 13: Mạch in 2 lớp .....	24
Hình 14: Mạch in 3D .....	24
Hình 15: Lưu đồ quét thẻ .....	26
Hình 16: Lưu đồ nút nhấn.....	27
Hình 17: Kết quả theo dõi hành vi mở cửa trên datasheet. ....	30
Hình 18: Mạch testboard.....	32

## CHƯƠNG 1. GIỚI THIỆU

### 1.1 Đặt vấn đề

Trong cuộc sống phát triển ngày nay, nhu cầu về tiện ích sống của mỗi gia đình và cá nhân đều đòi hỏi ngày càng cao. Một phần tử quan trọng trong hệ thống Smart Home chính là ổ khóa thông minh. Từ tính thiết yếu và trực quan của vấn đề, đề tài này nghiên cứu phát triển hệ thống khóa cửa tích hợp nhiều chức năng.

Câu hỏi nghiên cứu đặt ra của đề án là:

1. Hệ thống khóa tích hợp vi điều khiển có tốt hơn, nhiều ưu điểm hơn so với các bộ khóa truyền thống hay không?
2. Vi điều khiển ARM có thực sự phù hợp và tối ưu với đề tài?
3. Đề tài có nhiều cơ sở để mở rộng và phát triển không?

Trong chương 2, cơ sở lý thuyết sẽ được trình bày. Trong chương 3, các kết quả mô phỏng sẽ được so sánh và phân tích. Cuối cùng, chương 4 đưa ra kết luận chung.

### 1.2 Phạm vi và phương pháp nghiên cứu

- Nội dung 1: Tìm hiểu cách lập trình vi điều khiển STM32F103C8T6.
- Nội dung 2: Tìm hiểu giải thuật mở khóa bằng mật khẩu và thẻ từ, sắp xếp các trường hợp sử dụng.
- Nội dung 3: Tìm hiểu nguyên lý làm việc truyền dữ liệu từ ESP32 lên google trang tính với giao thức HTTP.
- Nội dung 4: Mô phỏng các kết quả thực hiện.
- Nội dung 5: Vẽ mạch in sử dụng phần mềm Altium.
- Nội dung 6: Lắp ráp hoàn thiện, đánh giá sản phẩm.

Công việc thực hiện:

1. Thiết kế phần cứng cho hệ thống khóa cửa, gia công mạch PCB.
2. Lập trình STM32F103C8T6 giao tiếp RC522 và LCD mở khóa bằng thẻ từ, các thao tác nút nhấn mở khóa mật khẩu, thay đổi dữ liệu thẻ từ.
3. Lập trình STM32F103C8T6 giao tiếp ESP32 và truyền dữ liệu lên google trang tính với giao thức HTTP.

### **1.3 Các đóng góp của luận văn**

Đề tài đồ án này có các đóng góp như sau:

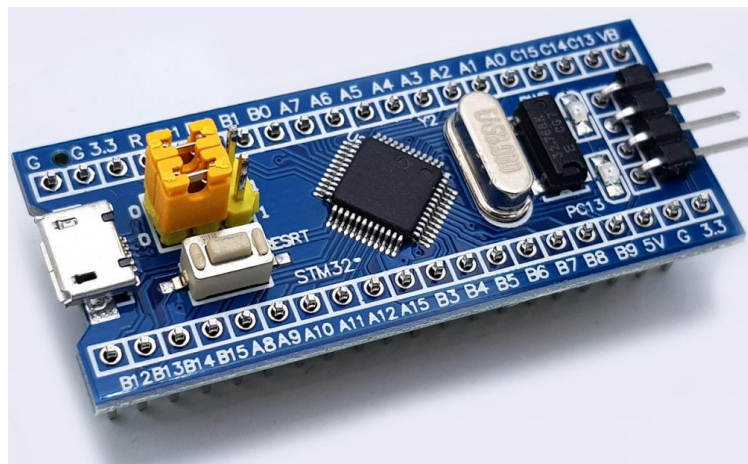
- Hiện thực hóa hệ thống khóa tích hợp vi điều khiển ARM.
- Tích hợp thành công dữ liệu theo dõi hành vi đóng/mở khóa cửa.
- So sánh, đánh giá ưu nhược điểm so với khóa truyền thống.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1 Kit phát triển STM32F103C8T6 Blue Pill ARM Cortex-M3

#### 2.1.1 Giới thiệu

STM32 là một trong những dòng chip phổ biến của STMicroelectronics với nhiều họ thông dụng như F0, F1, F2, F3, F4... STM32F103C8T6 thuộc họ F1 với lõi là ARM COTEX M3. STM32F103C8T6 là vi điều khiển 32 bit, tốc độ tối đa là 72Mhz.

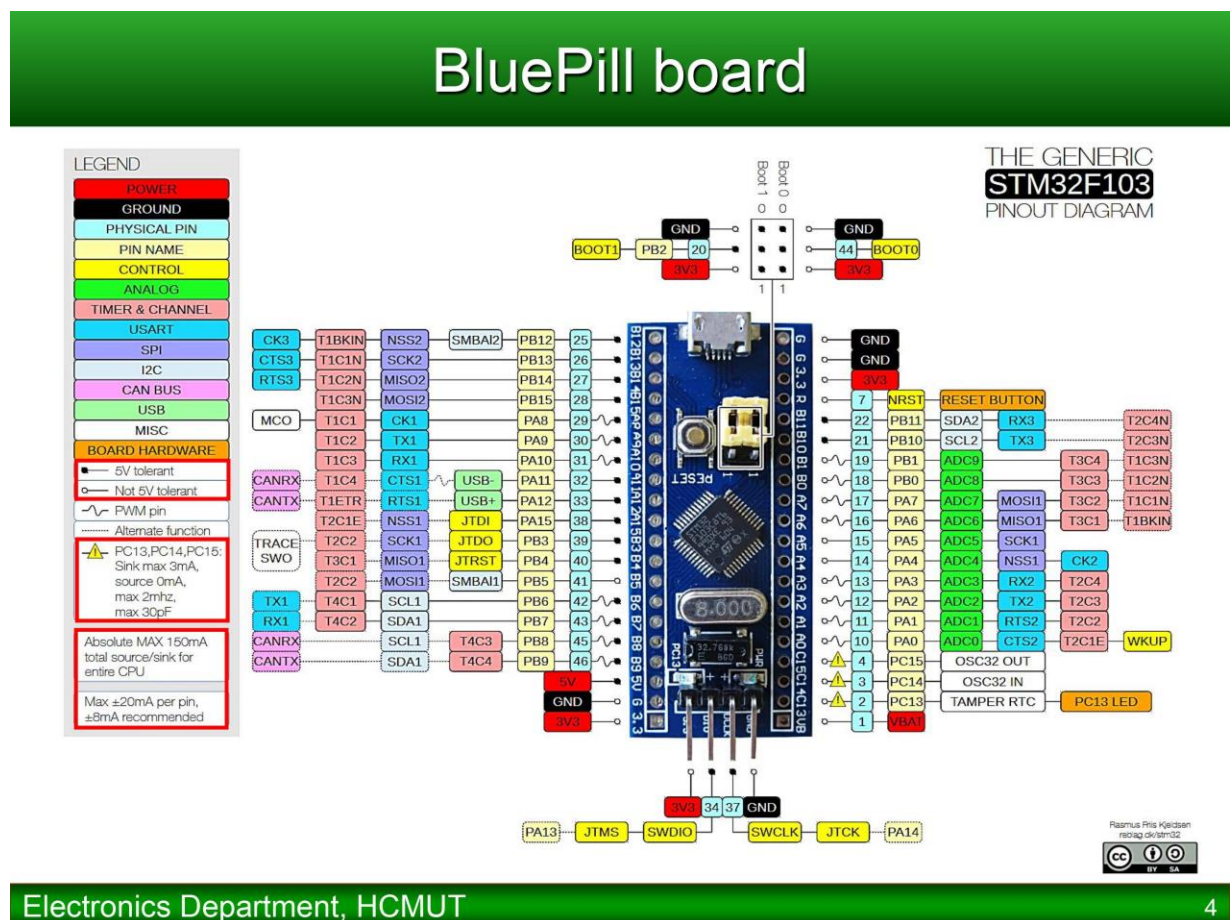


**Hình 1: Kit STM32 Bluepill (STM32F103C8T6)**

Kit phát triển STM32F103C8T6 Blue Pill ARM Cortex-M3 phù hợp với đề tài vì các tiêu chí: giá thành rẻ so với các loại vi điều khiển có chức năng tương tự; mạch nạp cũng như công cụ lập trình khá đa dạng và dễ sử dụng; một số ứng dụng chính (dùng cho driver để điều khiển ứng dụng, điều khiển ứng dụng thông thường, thiết bị cầm tay và thuốc, máy tính và thiết bị ngoại vi chơi game, GPS cơ bản, các ứng dụng trong công nghiệp, thiết bị lập trình PLC, biến tần, máy in, máy quét, hệ thống cảnh báo, thiết bị liên lạc nội bộ...).

Phần mềm lập trình: có khá nhiều trình biên dịch cho STM32 như STM32CubeIDE, IAR Embedded Workbench, Keil C... Trong đồ án này em sử dụng môi trường phát triển STM32CubeIDE vì tích hợp với STM32CubeMX giúp ích cho việc cấu hình ngoại vi, clock, tính toán dòng tiêu thụ, tạo project và là môi trường do chính ST company xây dựng phát triển.

### 2.1.2 Thông số kỹ thuật



Hình 2: STM32 Bluepill GPIO Pin

- Vi điều khiển: STM32F103C8T6.

- Điện áp cấp 5VDC qua cổng Micro USB sẽ được chuyển đổi thành 3.3VDC qua IC nguồn và cấp cho Vi điều khiển chính.
- Tích hợp sẵn thạch anh 8Mhz.
- Tích hợp sẵn thạch anh 32Khz cho các ứng dụng RTC.
- Ra chân đầy đủ tất cả các GPIO và giao tiếp: CAN, I2C, SPI, UART, USB, ...
- Tích hợp Led trạng thái nguồn, Led PC13, Nút Reset.
- Kích thước: 53.34 x 15.24mm.
- Sử dụng với các mạch nạp: ST-Link Mini/J-link/USB TO COM.
- Kết nối chân khi nạp bằng ST-Link Mini: Nạp theo chuẩn SWD.
  - TCK — SWCLK
  - TMS — SWDIO
  - GND — GND
  - 3.3V — 3.3V

### 2.1.3 Cấu hình

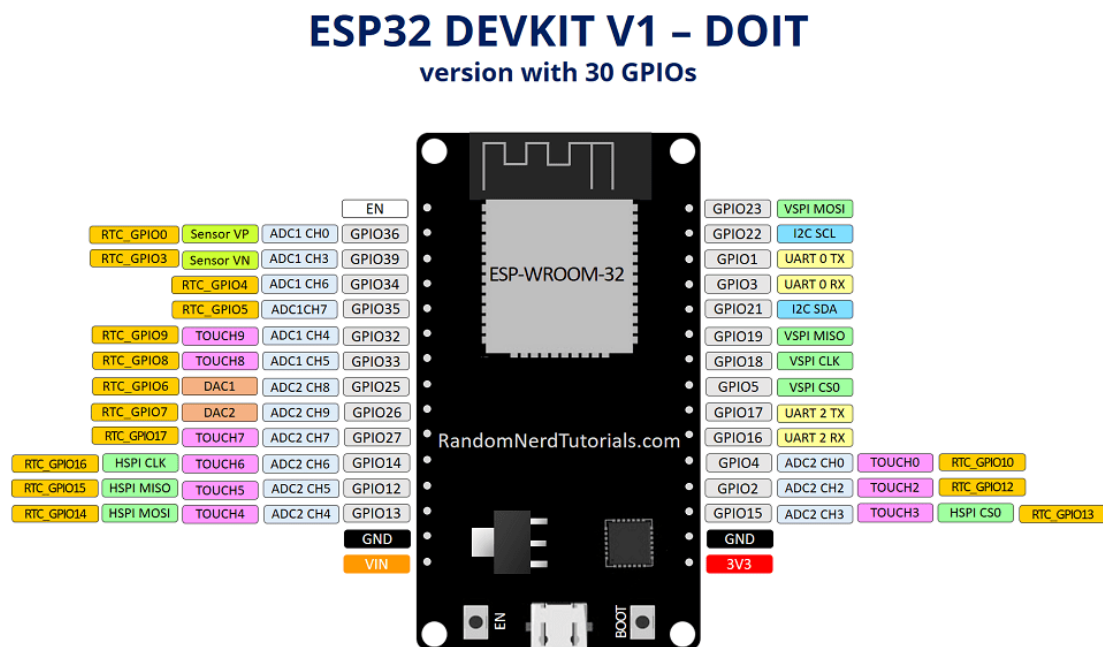
- ARM 32-bit Cortex M3 với clock max là 72Mhz.
- Bộ nhớ:
  - 64 kbytes bộ nhớ Flash (bộ nhớ lập trình).
  - 20kbytes SRAM.
- Clock, reset và quản lý nguồn:
  - Điện áp hoạt động 2.0V -> 3.6V.
  - Power on reset (POR), Power down reset (PDR) và programmable voltage detector (PVD).
  - Sử dụng thạch anh ngoài từ 4Mhz -> 20Mhz.
  - Thạch anh nội dùng dao động RC ở mode 8Mhz hoặc 40khz.
  - Sử dụng thạch anh ngoài 32.768khz được sử dụng cho RTC.



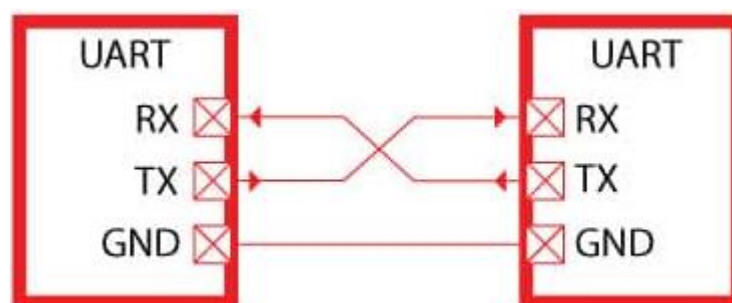
- Trong trường hợp điện áp thấp:
  - Có các mode: ngủ, ngừng hoạt động hoặc hoạt động ở chế độ chờ.
  - Cấp nguồn ở chân Vbat bằng pin để hoạt động bộ RTC và sử dụng lưu trữ data khi mất nguồn cấp chính.
- 2 bộ ADC 12 bit với 9 kênh cho mỗi bộ:
  - Khoảng giá trị chuyển đổi từ 0 – 3.6V.
  - Lấy mẫu nhiều kênh hoặc 1 kênh.
  - Có cảm biến nhiệt độ nội.
- DMA: cơ chế của hệ thống máy tính cho phép một thành phần phần cứng truy cập đến bộ nhớ dữ liệu chính (như RAM) một cách độc lập với CPU.
  - 7 kênh DMA.
  - Hỗ trợ DMA cho ADC, I2C, SPI, UART.
- 7 timer:
  - 3 timer 16 bit hỗ trợ các mode IC/OC/PWM.
  - 1 timer 16 bit hỗ trợ để điều khiển động cơ với các mode bảo vệ như ngắt input, dead-time...
  - 2 watchdog timer dùng để bảo vệ và kiểm tra lỗi.
  - 1 sysTick timer 24 bit đếm xuống dùng cho các ứng dụng như hàm Delay....
- Hỗ trợ 9 kênh giao tiếp bao gồm:
  - 2 bộ I2C(SMBus/PMBus).
  - 3 bộ USART (ISO 7816 interface, LIN, IrDA capability, modem control).
  - 2 SPIs (18 Mbit/s).
  - 1 bộ CAN interface (2.0B Active).
  - USB 2.0 full-speed interface
- Kiểm tra lỗi CRC và 96-bit ID.

## 2.2 Ngoại vi (ESP32 DEVKIT V1, RC522, LCD 1602, POWER, BUTTON)

### 2.2.1 Esp32 DEVKIT V1 30pins



Hình 3: ESP32 DEVKIT V1 30pins



Hình 4: giao tiếp UART giữa 2 MCU.

### 2.2.2 Module RFID RC522 NFC 13.56mhz

a. Giới thiệu.

Module RFID RC522 NFC 13.56mhz dùng để đọc và ghi dữ liệu cho thẻ NFC tần số 13.56mhz. Với mức thiết kế nhỏ gọn, linh hoạt module này là sự lựa chọn thích hợp cho các ứng dụng đọc – ghi thẻ NFC, đặc biệt khi sử dụng kết hợp với ARDUINO. RFID – Radio Frequency Identification Detection là công nghệ nhận dạng đối tượng bằng sóng vô tuyến. Là một phương pháp nhận dạng tự động dựa trên việc lưu trữ dữ liệu từ xa, sử dụng thiết bị Thẻ RFID và một Đầu đọc RFID.



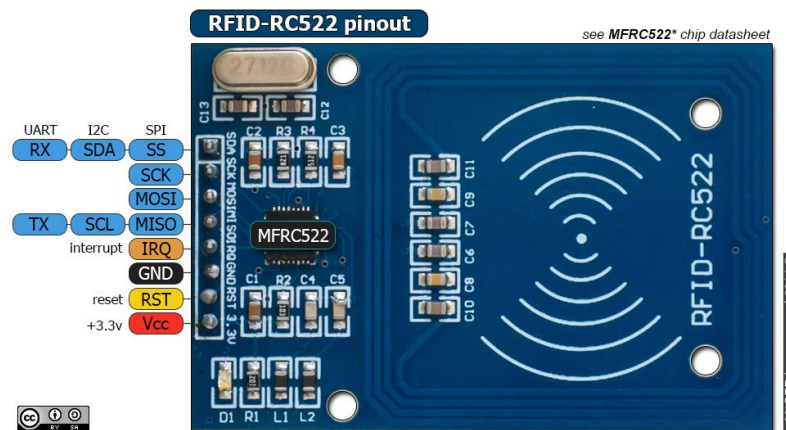
**Hình 5: Module thẻ từ RFID RC522 13,56Mhz**

b. Thông số kỹ thuật.

- Nguồn: 3.3VDC, 13 – 26mA.
- Dòng ở chế độ chờ: 10-13mA...
- Dòng ở chế độ nghỉ: <80uA.
- Tải tối đa: 30mA.

- Tần số sóng mang: 13.56MHz.
- Khoảng cách hoạt động: 0 ~ 60mm ( mifare1 card.
- Giao tiếp: SPI.
- Tốc độ truyền dữ liệu: tối đa 10Mbit/s.
- Các loại card RFID hỗ trợ: mifare1 S50, mifare1 S70, mifare UltraLight, mifare Pro, mifare Desfire.
- Kích thước: 40mm × 60mm.
- Nhiệt độ hoạt động: -20 đến 80 ° C.
- Độ ẩm hoạt động: 5% - 95%.
- Phụ kiện: móc khóa và thẻ.
- Có khả năng đọc và ghi
- Hỗ trợ: ISO / IEC 14443A /MIFAR.

c. Sơ đồ chân.



**Hình 6: RFID-RC522 pinout**

- SDA(SS) chân chọn lọc chip khi giao tiếp SPI (kích hoạt mức thấp).
- SCK: chân xung trong chế độ SPI.
- MOSI(SDI): Master Data Out – Slave In trong chế độ giao thiệp SPI.

- MISO(SDO): Master Data In – Slave Out trong chế độ giao thiệp SPI.
- IRQ: chân ngắt.
- GND: chân nối mass.
- RST: chân reset lại module.
- VCC: nguồn 3.3V.

d. Tính năng.

- MF RC522 vận dụng cho việc tích hợp cao việc đọc và viết dữ liệu.
- Giao tiếp với thẻ tại tần số 13.56Mhz.
- Là sự chọn lọc thấp cho sự lớn mạnh của những trang bị sáng tạo và trang bị di động cầm tay.
- MF RC552 dùng cho việc tăng điều chế và giải mã điều chế thông báo giao du thụ động bằng những bí quyết hoàn toàn phù hợp trong tần số 13.56Mhz.
- tương thích sở hữu bộ phát dấu hiệu 14443A.
- ISO 14443A xử lý kỹ thuật để phát hiện lỗi và những sườn hình.
- CRYPTO1 mau chóng tương trợ mã hóa thuật toán để công nhận sản phẩm là mafire.
- MF RC552 tương trợ mafire giao tiếp có các chuỗi bằng tốc độ cao, tốc độ truyền dữ liệu hai chiều lên tới 424kbit/s.
- MF RC552 cũng như vậy như MF RC500, MF RC530 nhưng cũng có các đặc điểm và sự khác biệt, giao tiếp giữa nó và máy chủ ở chế độ SPI giúp giảm thiểu các kết nối hạn hẹp của PCB, giảm mức giá đáng nhắc.
- Những MF 552 là những module được ngoài mặt để dễ dàng sử dụng mang các đầu đọc thẻ mạch.
- Nâng cao sự tăng trưởng của các vận dụng, đáp ứng nhu cầu về sử dụng các trang bị đầu/cuối tiêu dùng thẻ nhớ RF.
- Module này với thẻ được nạp trực tiếp vào những khuôn reader khác nhau, rất thuận tiện.

e. Ứng dụng.

- Ứng dụng quản lý lưu thông hàng hóa
- Ứng dụng quản lý kho hàng
- Ứng dụng quản lý thu phí đường bộ tự động
- Ngoài ra còn các ứng dụng như: quản lý nhà máy, quản lý thư viện, quản lý chấm công, quản lý bãi giữ xe, quản lý nhà ăn, quản lý sinh viên, quản lý bệnh viện, khóa cửa

### 2.2.3 LCD 1602

a. Giới thiệu.

Màn hình text LCD1602 xanh lá sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến.



**Hình 7: LCD1602 pin out**

b. Thông số kỹ thuật.

- Điện áp hoạt động là 5 V.
- Kích thước: 80 x 36 x 12.5 mm

- Chữ đen, nền xanh lá
- Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
- Tên các chân được ghi ở mặt sau của màn hình LCD hỗ trợ việc kết nối, đi dây điện.
- Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu
- Có bộ ký tự được xây dựng hỗ trợ tiếng Anh và tiếng Nhật, xem thêm HD44780 datasheet để biết thêm chi tiết.

c. Sơ đồ chân.

Số chân	Ký hiệu chân	Mô tả chân
1	Vss	Cấp điện 0V
2	Vcc	Cấp điện 5V
3	V0	Chỉnh độ tương phản
4	RS	Lựa chọn thanh ghi địa chỉ hay dữ liệu
5	RW	Lựa chọn thanh ghi đọc hay viết
6	EN	Cho phép xuất dữ liệu
7	D0	Đường truyền dữ liệu 0
8	D1	Đường truyền dữ liệu 1
9	D2	Đường truyền dữ liệu 2

10	D3	Đường truyền dữ liệu 3
11	D4	Đường truyền dữ liệu 4
12	D5	Đường truyền dữ liệu 5
13	D6	Đường truyền dữ liệu 6
14	D7	Đường truyền dữ liệu 7
15	A	Cực dương led nền
16	K	Cực âm led nền

d. Bộ nhớ.

Bộ điều khiển LCD có ba vùng nhớ nội, mỗi vùng có chức năng riêng. Bộ điều khiển phải khởi động trước khi truy cập bất kỳ vùng nhớ nào.

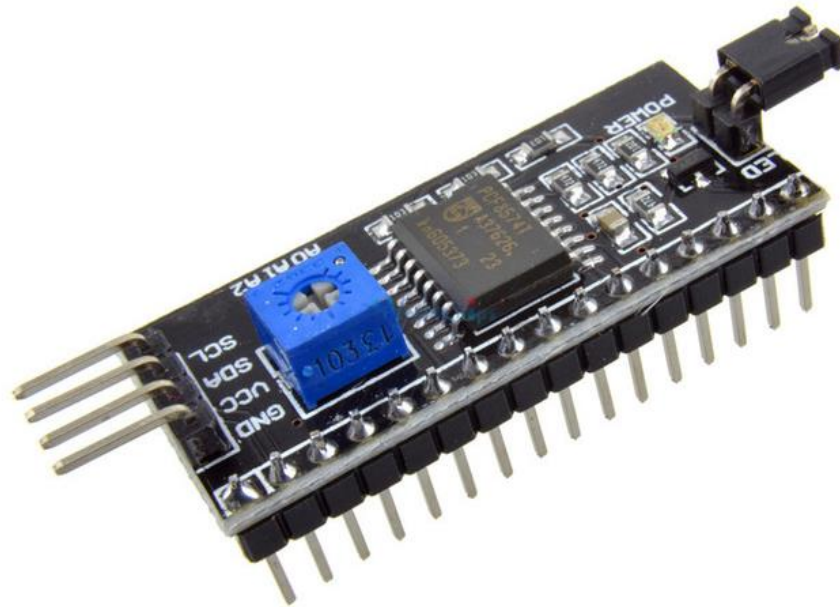
Bộ nhớ chứa dữ liệu để hiển thị (Display Data RAM: DDRAM) lưu trữ những mã ký tự để hiển thị lên màn hình. Mã ký tự lưu trữ trong vùng DDRAM sẽ tham chiếu với từng bitmap ký tự được lưu trữ trong CGROM đã được định nghĩa trước hoặc đặt trong vùng do người sử dụng định nghĩa.

Bộ phát ký tự ROM (Character Generator ROM: CGROM) chứa các kiểu bitmap cho mỗi ký tự được định nghĩa trước mà LCD có thể hiển thị, như được trình bày bằng mã ASCII. Mã ký tự lưu trữ trong DDRAM cho mỗi vùng ký tự sẽ được tham chiếu đến một vị trí trong CGROM. Ví dụ: mã ký tự số hex 0x53 lưu trữ trong DDRAM được chuyển sang dạng nhị phân 4 bit cao là  $DB[7:4] = "0101"$  và 4 bit thấp là  $DB[3:0] = "0011"$  chính là ký tự chữ 'S' sẽ hiển thị trên màn hình LCD.

Bộ phát ký tự RAM (Character Generator RAM: CG RAM) cung cấp vùng nhớ để tạo ra 8 ký tự tùy ý. Mỗi ký tự gồm 5 cột và 8 hàng.



### 2.2.4 Module I2C giao tiếp LCD1602



**Hình 8: Module I2C LCD 1602**

a. Giới thiệu

- LCD có quá nhiều nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển. Module I2C LCD ra đời và giải quyết vấn đề này.
- Thay vì phải mất 6 chân vi điều khiển để kết nối với LCD 16×2 (RS, EN, D7, D6, D5 và D4) thì module IC2 chỉ cần tốn 2 chân (SCL, SDA) để kết nối.
- Module I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 16×2, LCD 20×4, ...) và tương thích với hầu hết các vi điều khiển hiện nay.

b. Ưu điểm

- Tiết kiệm chân cho vi điều khiển.
- Dễ dàng kết nối với LCD.

c. Thông số kỹ thuật

- Điện áp hoạt động: 2.5-6V DC.
- Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780).
- Giao tiếp: I2C.
- Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).
- Tích hợp Jump chốt để cung cấp đèn cho LCD hoặc ngắt.
- Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

2.2.5 Các ngoại vi khác.

a. Khóa chốt điện từ LY-03 12VDC.

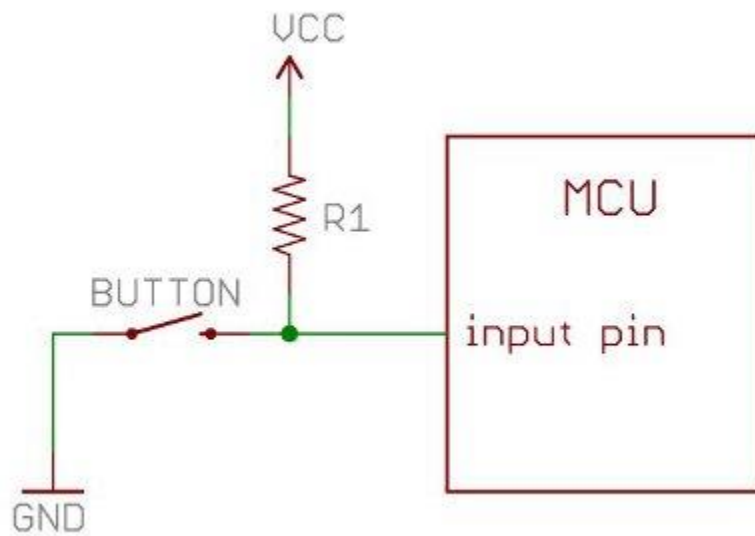
Thông số kỹ thuật:

- Điện áp sử dụng: 12 VDC
- Dòng điện tiêu thụ: 0.8A.
- Công suất tiêu thụ: 9.6W
- Sử dụng Solenoid từ.
- Tốc độ phản ứng: < 1s.
- Thời gian kích liên tục: < 10s



**Hình 9: Khóa chốt điện từ LY-03 12VDC**

b. Nút nhấn điện trở kéo lên (Pull-up Button).



**Hình 10: Button input pull-up**

## 2.3 Phương thức truyền dữ liệu HTTP

HTTP (Hypertext Transfer Protocol) là giao thức truyền tải siêu văn bản, được sử dụng chủ yếu để truyền tải dữ liệu giữa máy khách (client) và máy chủ (server) trên mạng internet. HTTP là một giao thức ứng dụng trong mô hình OSI, hoạt động ở lớp ứng dụng và dựa trên mô hình client-server.

Các thành phần của HTTP:

- Client: Thường là trình duyệt web hoặc ứng dụng gửi yêu cầu HTTP tới máy chủ.
- Server: Máy chủ web nhận yêu cầu HTTP từ client và phản hồi với dữ liệu yêu cầu.

Các phương thức HTTP cơ bản

HTTP có nhiều phương thức, mỗi phương thức phục vụ một mục đích cụ thể. Các phương thức phổ biến nhất bao gồm:

a. GET

- Mô tả: Lấy dữ liệu từ máy chủ.
- Sử dụng: Khi client muốn truy vấn thông tin hoặc tải xuống dữ liệu.
- Đặc điểm:
  - Tham số được truyền trong URL.
  - Không có phần thân yêu cầu (body).
  - Idempotent (kết quả của nhiều yêu cầu GET giống nhau là như nhau).

Ví dụ: Truy vấn thông tin người dùng

```
GET /users?name=John HTTP/1.1
Host: example.com
```

b. POST

- Mô tả: Gửi dữ liệu tới máy chủ để xử lý.
- Sử dụng: Khi client muốn gửi dữ liệu để tạo mới tài nguyên hoặc thực hiện xử lý nào đó.
- Đặc điểm:
  - Dữ liệu được gửi trong thân yêu cầu.
  - Không idempotent (mỗi yêu cầu POST có thể tạo ra kết quả khác nhau).

Ví dụ: Tạo mới một người dùng

```
POST /users HTTP/1.1
Host: example.com
Content-Type: application/json
Content-Length: 34
{
  "name": "John",
  "age": 30
}
```

#### c. PUT

- Mô tả: Cập nhật hoặc tạo mới tài nguyên trên máy chủ.
- Sử dụng: Khi client muốn cập nhật toàn bộ thông tin của tài nguyên hoặc tạo mới nếu tài nguyên không tồn tại.
- Đặc điểm:
  - Dữ liệu được gửi trong thân yêu cầu.
  - Idempotent.

Ví dụ: Cập nhật thông tin người dùng

```
PUT /users/1 HTTP/1.1
```

```
Host: example.com
Content-Type: application/json
Content-Length: 34
{
  "name": "John",
  "age": 31
}
```

#### d. DELETE

- Mô tả: Xóa tài nguyên trên máy chủ.
- Sử dụng: Khi client muốn xóa tài nguyên.
- Đặc điểm:
  - Idempotent.

Ví dụ: Xóa người dùng

```
DELETE /users/1 HTTP/1.1
Host: example.com
```

#### e. PATCH

- Mô tả: Cập nhật một phần tài nguyên trên máy chủ.
- Sử dụng: Khi client muốn cập nhật một phần của tài nguyên.
- Đặc điểm:
  - Dữ liệu được gửi trong thân yêu cầu.
  - Không idempotent.

Ví dụ: Cập nhật một phần thông tin người dùng

PATCH /users/1 HTTP/1.1

```
Host: example.com
Content-Type: application/json
```

```
Content-Length: 17
```

```
{  
  "age": 32  
}
```

f. HEAD

- Mô tả: Lấy phần đầu của phản hồi HTTP, tương tự như GET nhưng không có phần thân.
- Sử dụng: Khi client muốn kiểm tra metadata của tài nguyên mà không cần tải xuống toàn bộ.
- Đặc điểm:
  - Idempotent.

Ví dụ: Kiểm tra header của người dùng

```
HEAD /users/1 HTTP/1.1  
Host: example.com
```

g. OPTIONS

- Mô tả: Truy vấn các phương thức HTTP mà máy chủ hỗ trợ cho URL cụ thể.
- Sử dụng: Khi client muốn biết các phương thức có thể thực hiện trên tài nguyên.
- Đặc điểm:
- Idempotent.

Ví dụ: Kiểm tra các phương thức được hỗ trợ

```
OPTIONS /users HTTP/1.1  
Host: example.com
```

## 2.4 Kết luận chương

Trong chương này, chúng ta đã tìm hiểu về các thành phần cơ bản của kit phát triển STM32F103C8T6 Blue Pill ARM Cortex-M3, bao gồm giới thiệu, thông số kỹ thuật và cách cấu hình chúng. Chúng ta cũng đã xem xét các ngoại vi phổ biến như ESP32 DEVKIT V1, RC522, LCD 1602 và cách sử dụng chúng trong các ứng dụng IoT.

Ngoài ra, nội dung đã đi sâu vào phương thức truyền dữ liệu HTTP, một phần quan trọng trong việc giao tiếp giữa các thiết bị IoT và máy chủ. Chúng ta đã hiểu về các phương thức HTTP như GET, POST, PUT, DELETE, PATCH, HEAD và OPTIONS, cùng với các ví dụ minh họa.

Tổng cộng, chương này cung cấp một nền tảng lý thuyết cơ bản và quan trọng cho việc phát triển ứng dụng trong 2 chương tiếp theo “Thiết kế phần cứng” và “Thiết kế phần mềm” trên nền tảng STM32F103C8T6 Blue Pill ARM Cortex-M3.



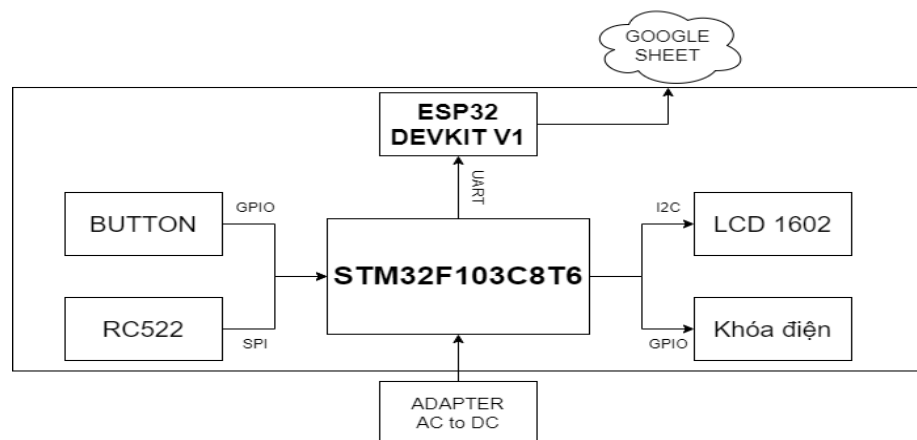
## CHƯƠNG 3. THIẾT KẾ PHẦN CỨNG

### 3.1 Yêu cầu thiết kế.

Các phần cứng yêu cầu gồm:

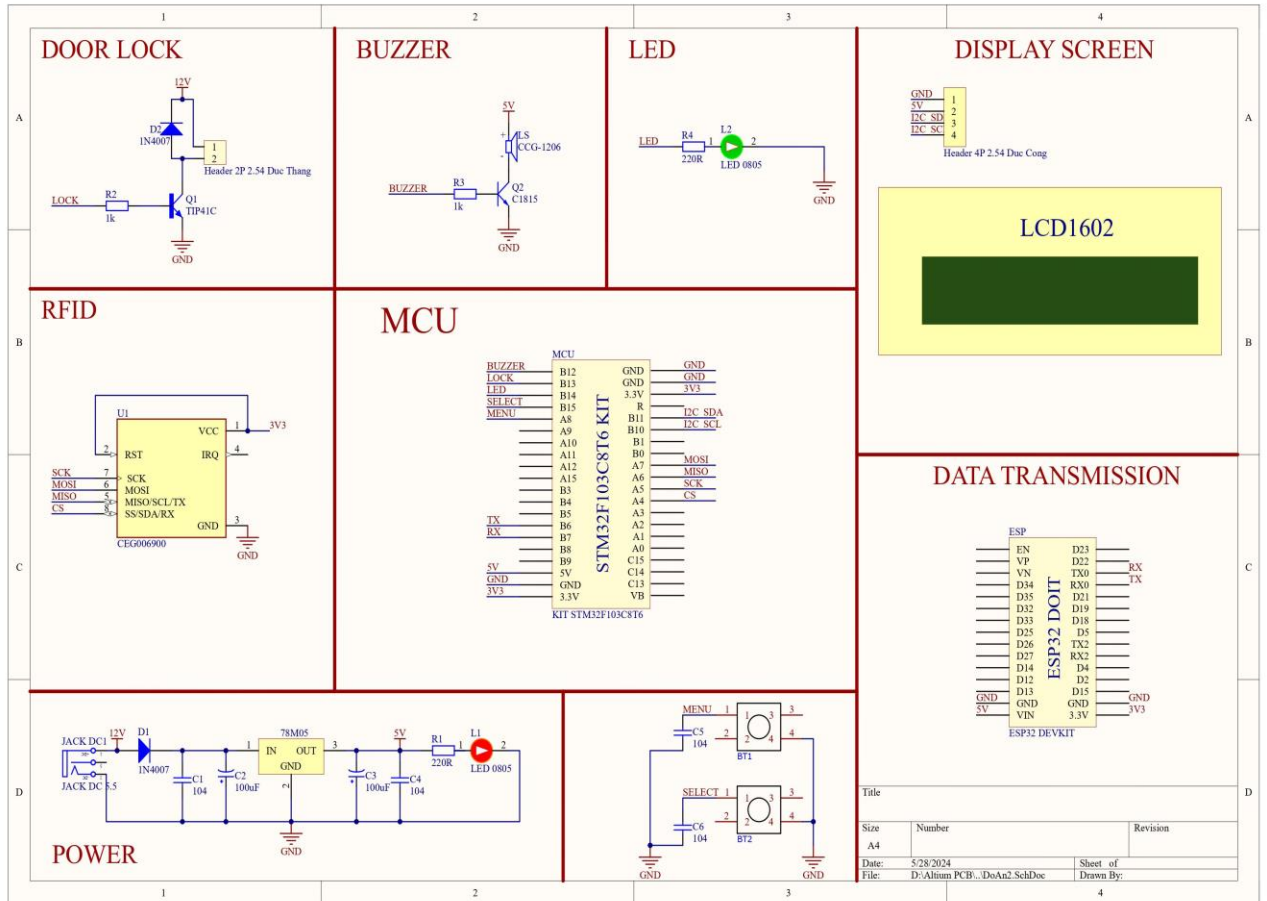
- Vi điều khiển để giao tiếp với thiết bị quét thẻ, nút nhấn và thiết bị hiển thị thông tin.
- Thiết bị dùng để quét thẻ đọc dữ liệu của thẻ từ và các tùy chọn (tùy chọn gồm có: thêm và xóa thẻ).
- Thiết bị hiển thị thông tin để hiển thị các lựa chọn cho người dùng và hiển thị quá trình quét thẻ cũng như trực quan hóa quá trình xử lý của khóa cửa.
- Nút nhấn dùng để điều khiển thực hiện các tính năng mở cửa, thêm, xóa, kiểm tra thẻ.
- Khóa điện từ dùng để đóng mở cửa, đóng ngắt thông qua GPIO.
- Nguồn Adapter cung cấp điện cho khóa cửa.

### 3.2 Đặc tả hệ thống.



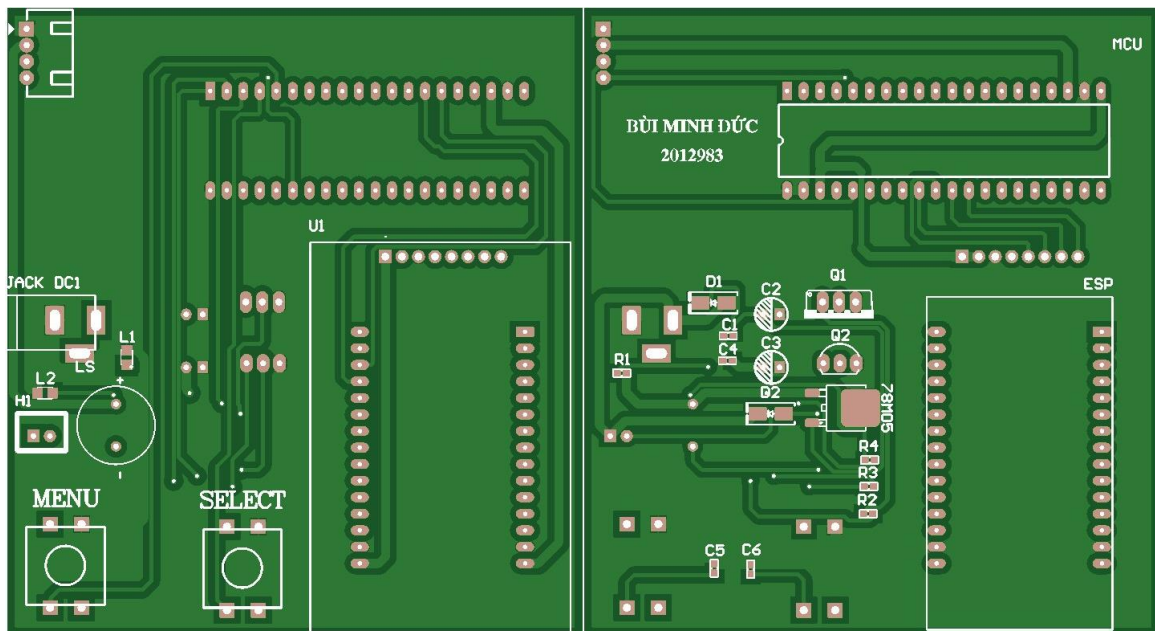
Hình 11: Sơ đồ khối hệ thống

### 3.3 Mô phỏng (Schematic Circuit).

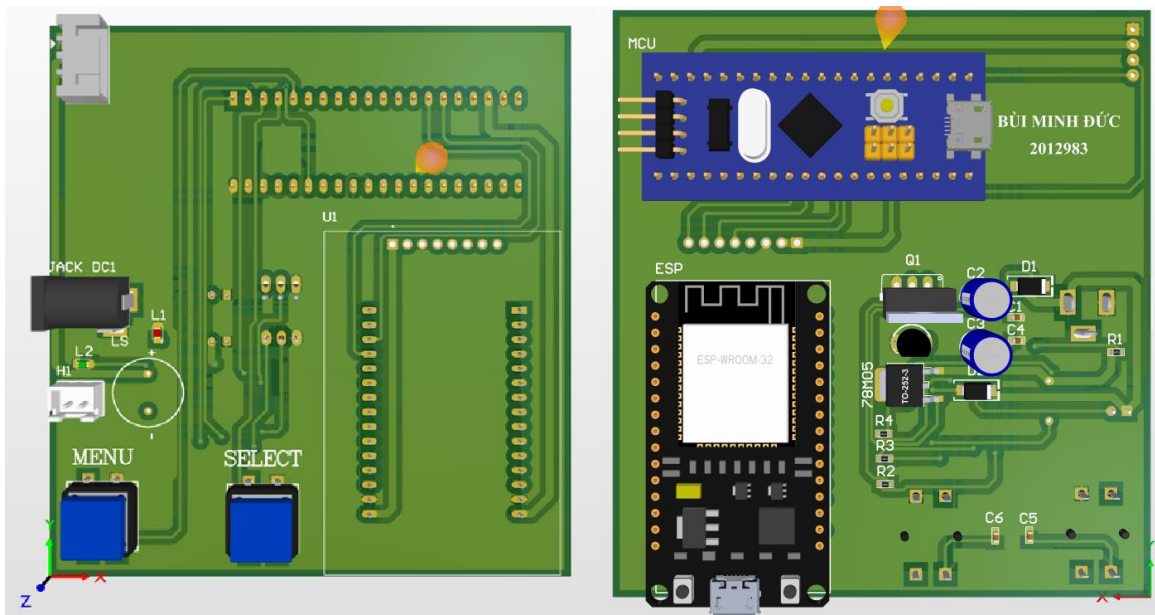


Hình 12: Sơ đồ nguyên lý

### 3.4 Mạch in (PCB Layout).



Hình 13: Mạch in 2 lớp



Hình 14: Mạch in 3D

### 3.5 Kết luận chương.

Trong chương này, em đã thực hiện việc thiết kế phần cứng cho khóa cửa sử dụng module RC522 và vi điều khiển STM32 Blue Pill. Dưới đây là những điểm chính:

**Yêu Cầu Thiết Kế:** Chúng ta đã xác định và phân tích các yêu cầu cần thiết cho việc thiết kế khóa cửa với khả năng đọc thẻ từ RFID và điều khiển bằng vi điều khiển STM32 Blue Pill và giao tiếp nút nhấn để điều khiển ngõ ra; thiết kế giao tiếp giữa 2 vi điều khiển để thực hiện chức năng truyền dữ liệu.

**Đặc Tả Hệ Thống:** Một bản đặc tả hệ thống đã được tạo ra để mô tả các chức năng cụ thể mà khóa cửa cần thực hiện, bao gồm việc đọc thẻ từ RFID, xác thực và điều khiển khóa.

**Mô Phỏng (Schematic Circuit):** em đã thực hiện mô phỏng mạch schematic của khóa cửa, bao gồm kết nối của module RC522 và vi điều khiển STM32 Blue Pill, cũng như các linh kiện khác như các phần tử led, điện trở, tụ.

**Mạch In (PCB Layout):** Dựa trên mạch schematic đã được mô phỏng, em đã thiết kế mạch in PCB để thực hiện kết nối vật lý giữa các linh kiện.

Kết quả của chương này là việc thiết kế một hệ thống hoàn chỉnh và sẵn sàng cho việc sản xuất khóa cửa RFID sử dụng vi điều khiển STM32 Blue Pill và module RC522. Các bước tiếp theo sẽ bao gồm việc phát triển phần mềm, thử nghiệm và triển khai hệ thống trong môi trường thực tế.

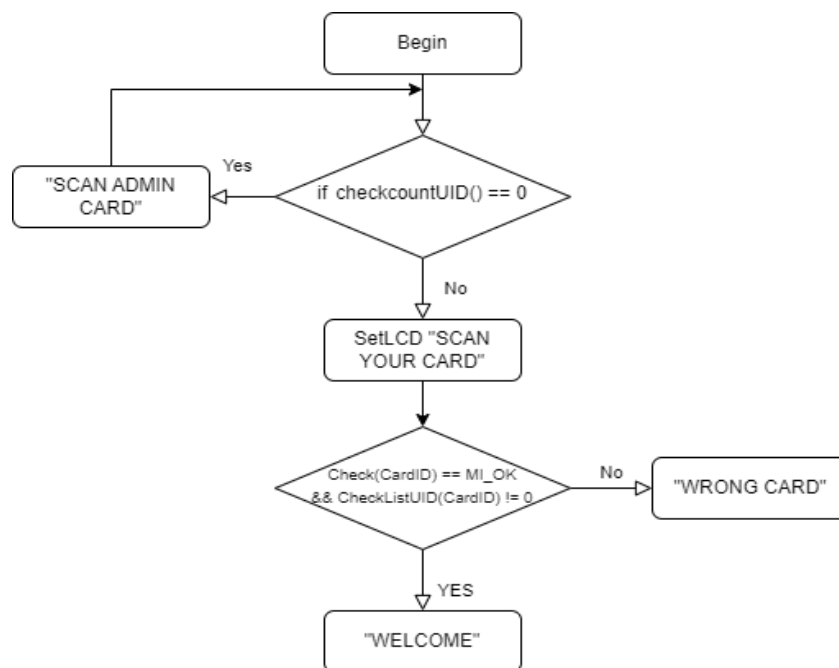
## CHƯƠNG 4. THIẾT KẾ PHẦN MỀM

### 4.1 Yêu cầu phần mềm.

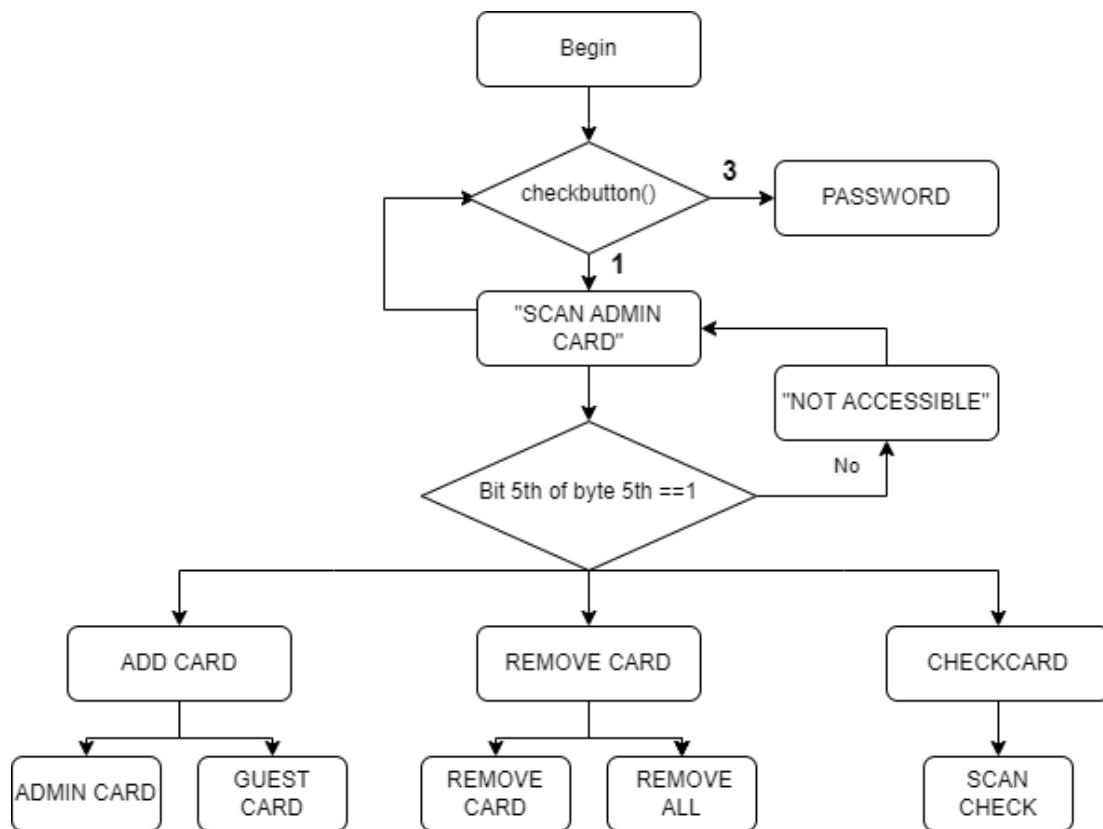
Vi điều khiển STM32F103C8T6 (sử dụng ngôn ngữ C):

- Giao tiếp với các ngoại vi: RC522, LCD.
- Xử lý so sánh dữ liệu đọc được của thẻ từ với dữ liệu được lưu trên flash.
- Xử lý thao tác các tính năng của hệ thống thông qua nút nhấn.
- Giao tiếp truyền dữ liệu bằng giao thức UART từ vi điều khiển trung tâm STM32F103C8T6 tới ESP32 DEVKIT V1, sau đó truyền dữ liệu theo dõi hành vi mở cửa lên googlesheet bằng giao thức HTTP.

### 4.2 Lưu đồ giải thuật.



Hình 15: Lưu đồ quét thẻ



**Hình 16: Lưu đồ nút nhấn**

### 4.3 Giao tiếp STM32F103C8T6 với ESP32 DEVKIT V1 truyền dữ liệu.

#### 4.3.1 STM32F103C8T6

- Cấu hình UART chỉ truyền dữ liệu tới ESP32:

```

huart1.Instance = USART1;
huart1.Init.BaudRate = 115200;
huart1.Init.WordLength = UART_WORDLENGTH_8B;
huart1.Init.StopBits = UART_STOPBITS_1;
huart1.Init.Parity = UART_PARITY_NONE;
huart1.Init.Mode = UART_MODE_TX;
huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart1.Init.OverSampling = UART_OVERSAMPLING_16;

```

- Truyền dữ liệu khi cửa mở, sai thẻ, nhập sai mật khẩu:

```
char opened[] = "OPENED";  
char wrong[]="WRONG-CARD";  
char wrongpass []="WRONG-PASSWORD";  
  
HAL_UART_Transmit(&huart1, opened, sizeof(opened)-1,  
HAL_MAX_DELAY);  
HAL_UART_Transmit(&huart1, wrong, sizeof(wrong)-1,  
HAL_MAX_DELAY);  
HAL_UART_Transmit(&huart1, wrongpass, sizeof(wrongpass)-1,  
HAL_MAX_DELAY);
```

### 4.3.2 ESP32

Cấu hình WiFi và Google Apps Script API: Trong hàm setup(), board sẽ lặp lại việc kiểm tra kết nối Wi-Fi mỗi 0.5 giây cho đến khi kết nối thành công.

Đọc dữ liệu từ UART: Trong hàm loop(), board sẽ kiểm tra xem có dữ liệu được nhận từ cổng UART không. Nếu có, nó sẽ đọc dữ liệu đó và kiểm tra xem dữ liệu có phải là "OPENED", "WRONG-CARD" hoặc "WRONG-PASSWORD" hay không.

```
void loop() {  
  if (Serial.available() > 0) {  
    String data = Serial.readString();  
    if (data.equals("OPENED") || data.equals("WRONG-CARD") ||  
data.equals("WRONG-PASSWORD")) {  
      // Gửi dữ liệu lên Google Sheets  
      sendToGoogleSheet(data);  
    }  
  }  
}
```

Gửi dữ liệu lên Google Sheets: Nếu dữ liệu được đọc từ UART khớp với một trong các điều kiện trên, hàm sendToGoogleSheet(data) sẽ được gọi để gửi dữ liệu lên Google Sheets. Hàm này sẽ tạo một URL với dữ liệu và ID của Google Script đã được cấu hình trước đó, sau đó sử dụng HTTPClient để gửi yêu cầu HTTP GET đến URL đó.

```
void sendToGoogleSheet(String data) {
    HTTPClient http;
    String url = "https://script.google.com/macros/s/" +
    GOOGLE_SCRIPT_ID + "/exec?data=" + data;
    Serial.println("Sending data to Google Sheets: " + url);
    http.begin(url);
    int httpCode = http.GET();
    if (httpCode > 0) {
        String payload = http.getString();
        Serial.println("HTTP Status Code: " + String(httpCode));
        Serial.println("Payload: " + payload);
    }
    http.end();
}
```

Xử lý phản hồi: Nếu yêu cầu được gửi thành công, mã trạng thái HTTP sẽ được nhận và được in ra Serial Monitor cùng với nội dung của phản hồi.

```
11:54:20.472 -> Sending data to Google Sheets:
https://script.google.com/macros/s/AKfycbzIM7Ztem4jFurzu8ji5rI4vWGs_UEi
AQcnERTMAjQU5tCKQ-3YL9-2jKc7A9b3CKg-/exec?data=OPENED
11:54:24.199 -> HTTP Status Code: 302
11:54:24.199 -> Payload: <HTML>
11:54:24.199 -> <HEAD>
11:54:24.199 -> <TITLE>Moved Temporarily</TITLE>
11:54:24.199 -> </HEAD>
11:54:24.199 -> <BODY BGCOLOR="#FFFFFF" TEXT="#000000">
11:54:24.199 -> <!-- GSE Default Error -->
11:54:24.199 -> <H1>Moved Temporarily</H1>
11:54:24.199 -> The document has moved <A
HREF="https://script.googleusercontent.com/macros/echo?user_content_key
=ifFEAdOLKvurjCSI5W5y7g6v-5ZWQjj84mWXHqXewv0TavAmq4yK9E0Tz-
eqXRmliMghudVMjMTqFiNdFiLUgQEa59x9Q_cVm5_BxDlH2jW0nuo2oDemN9CCS2h10ox_1
xSncGQajx_ryfhECjZEnIWu_1lHWyaLEjBnj_e64kPWR_oeXlPg-
rq8XPe2wb74kiSAVI5yWMwzwCJRd91j1fCvJv57iMpT54t7drn5i_PnZZBe_Spf4ys0x5uc
xG8SkOGtzZ5nbiM&lib=MLUffuM6Cb4lHYx12cdJNzhgCqzOd43Ir">here</A>.
11:54:24.246 -> </BODY>
11:54:24.246 -> </HTML>
11:54:24.246 ->
```

### 4.3.3 Google Apps Script API



Lưu lại đường dẫn của datasheet muốn làm địa chỉ đích lưu trữ. Sử dụng đoạn mã JavaScript mã nguồn mở hoạt động dựa trên một yêu cầu HTTP GET gửi từ máy khách (client) đến một ứng dụng Google Apps Script được triển khai trên Google Apps Script Engine.

	A	B	C
1	<b>DOOR LOCK INFORMATION MANAGEMENT</b>		
2	Date	Time	Action
3	28/05/2024	02:16:09	OPENED
4	28/05/2024	02:23:06	OPENED
5	28/05/2024	02:23:32	OPENED
6	28/05/2024	03:37:04	OPENED
7	28/05/2024	03:37:35	OPENED
8	28/05/2024	03:38:39	OPENED
9	28/05/2024	03:38:44	OPENED
10	28/05/2024	03:38:49	OPENED
11	28/05/2024	03:44:31	OPENED
12	28/05/2024	03:53:21	WRONG-PASSWORD
13	28/05/2024	03:54:01	WRONG-PASSWORD
14	28/05/2024	03:57:18	OPENED
15	28/05/2024	03:57:25	WRONG-CARD
16	28/05/2024	03:57:57	WRONG-PASSWORD
17	28/05/2024	07:46:17	OPENED
18	28/05/2024	07:46:29	WRONG-CARD
19	28/05/2024	07:47:34	OPENED
20	28/05/2024	07:47:56	OPENED
21	28/05/2024	07:48:01	OPENED
22	28/05/2024	07:50:22	OPENED
23	28/05/2024	09:31:38	OPENED
24	28/05/2024	11:54:23	OPENED

**Hình 17: Kết quả theo dõi hành vi mở cửa trên datasheet.**

#### **4.4 Kết luận chương.**

Chương 4 tập trung vào việc xây dựng phần mềm điều khiển và quản lý hệ thống, bao gồm giao tiếp với các ngoại vi, xử lý dữ liệu từ thẻ từ, xử lý thao tác của người dùng qua nút nhấn, và giao tiếp truyền dữ liệu lên Google Sheets thông qua giao thức HTTP.

*Giao tiếp với các ngoại vi:*

Phần mềm được phát triển để giao tiếp với hai ngoại vi chính là RC522 (đọc thẻ từ) và LCD (hiển thị thông tin). Qua việc sử dụng các thư viện phù hợp, việc giao tiếp và điều khiển các ngoại vi này đã được triển khai một cách hiệu quả.

*Xử lý so sánh dữ liệu thẻ từ với dữ liệu trên flash:*

Phần mềm thực hiện việc đọc dữ liệu từ thẻ từ và so sánh với dữ liệu đã được lưu trữ trước đó trong bộ nhớ flash của vi điều khiển. Quá trình so sánh này được thực hiện để xác định xem thẻ từ được đọc có được phép truy cập hay không.

*Xử lý thao tác của người dùng qua nút nhấn:*

Hệ thống được thiết kế để phản ứng với các thao tác của người dùng thông qua các nút nhấn. Các chức năng như mở cửa, quay lại hoặc các tùy chọn khác được điều khiển và thực hiện dựa trên sự tương tác với các nút nhấn.

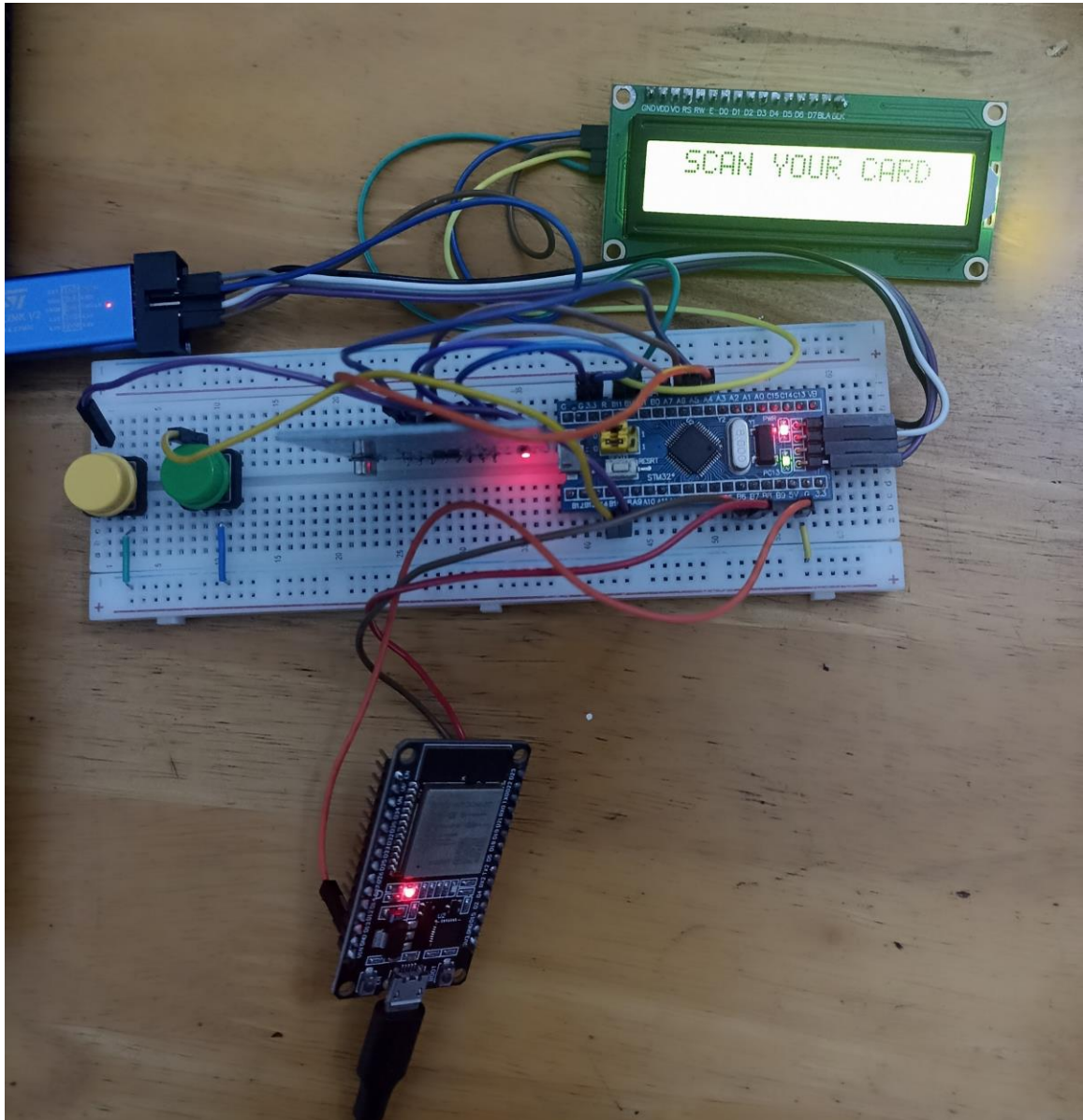
*Giao tiếp truyền dữ liệu lên Google Sheets:*

Dữ liệu về hành vi mở cửa được ghi lại và truyền lên Google Sheets thông qua giao thức HTTP. Qua việc sử dụng ESP32 DEVKIT V1, việc gửi dữ liệu này được thực hiện một cách nhanh chóng và đáng tin cậy.

Tóm lại, phần mềm được thiết kế và triển khai trong chương này đã đáp ứng được các yêu cầu chức năng và hiệu suất của hệ thống, giúp cải thiện quản lý và giám sát việc mở cửa và các hoạt động liên quan.

## CHƯƠNG 5. KẾT LUẬN

### 5.1 Hoàn thiện sản phẩm.



Hình 18: Mạch testboard

## 5.2 Tóm tắt và kết luận chung

### 5.2.1 Kết quả đạt được

Đề tài “Thiết kế hệ thống khóa từ RFID tích hợp theo dõi hành vi mở cửa” về cơ bản em đã hoàn thành các yêu cầu đặt ra:

Thiết lập ý tưởng đề tài: Hệ thống đã được thiết kế với sự tích hợp của vi điều khiển STM32F103C8T6 và các module ngoại vi như ESP32 DEVKIT V1, RC522 và I2C LCD. Các linh kiện tương thích với nhau cho độ phản hồi nhanh và dễ dàng giao tiếp, với nhiều mã nguồn mở trên internet.

Thiết kế hệ thống phần cứng: Hệ thống phản hồi tương đối ổn định (chỉ có trường hợp nhiễu LCD tần suất thấp do đặt gần nguồn Adapter), đáp ứng tốt các yêu cầu về bảo mật và chức năng.

Truyền dữ liệu lên Google Sheets: dữ liệu về hành vi mở cửa được truyền tải lên Google Sheets thông qua ESP32 và giao thức HTTP, giúp người quản lý dễ dàng theo dõi và giám sát từ xa.

### 5.2.2 Hạn chế

Hệ thống PCB còn sử dụng nhiều module, không tối ưu hóa về hiệu năng và chi phí thi công đề tài. Độ phản hồi thời gian thực không đạt được mức tuyệt đối.

Giải thuật lập trình trên STM32 chưa phải là tối ưu hóa thi hàm main và các hàm thư viện có độ dài chưa tối ưu.

Google Apps Script API là mã nguồn mở, phương án quản lý dữ liệu cá nhân bằng google sheet chưa thực sự là phù hợp.

### 5.3 Hướng phát triển.

Đối với phần cứng: tích hợp, chuyển đổi hệ thống PCB nhỏ gọn, tối ưu chi phí với các loại linh kiện dán SMT. Xây dựng hệ thống khóa hoàn chỉnh với vật liệu chắc chắn như là đóng vỏ bảo vệ, hướng tới thi công như một sản phẩm có thể thương mại hóa.

Đối với phần mềm: chuyển đổi sang cấu trúc hệ điều hành nhúng FREERTOS phân chia luồng chương trình (thread) giúp giảm độ chế và tối ưu hóa hiệu năng, công suất tiêu thụ.

Đối với phát triển đa chức năng: tích hợp thành hệ thống khóa bảo mật hộ gia đình với các bảo mật sinh trắc học (vân tay, gương mặt). Chuyển đổi luồng dữ liệu datasheet lên webserver và ứng dụng cài đặt trên điện thoại cá nhân.

## PHỤ LỤC A

### A.1 Phần cứng (hardware)

[https://drive.google.com/drive/folders/1kBeXTyDT3pOkx9eyIS\\_53H07kvpy80rV?usp=sharing](https://drive.google.com/drive/folders/1kBeXTyDT3pOkx9eyIS_53H07kvpy80rV?usp=sharing)

### A.2 Phần mềm (software)

<https://drive.google.com/drive/folders/1fcs1015bCC1PYU4lD3xqMwvkIau-g6hQ?usp=sharing>

### A.3 Kết quả (video)

<https://drive.google.com/drive/folders/1j9G4RupUx0KSoE5UFg37ju5frEJ5kIk8?usp=sharing>

## TÀI LIỆU THAM KHẢO

- [1] Th.s Bùi Quốc Bảo, “Giáo trình môn học Lập trình Hệ thống Nhúng”, Bộ môn Điện tử, Trường Đại học Bách khoa - ĐHQG TP.HCM, 2024.
- [2] T.s Hoàng Trang & Th.s Bùi Quốc Bảo, sách “Lập trình Hệ thống Nhúng”, NXB ĐHQG, 2019.
- [3] Long Automation. (2023, December 7). ESP32 - Google Sheets | Ghi dữ liệu cảm biến lên Google Sheets - Ultrasonic Sensor [Video]. YouTube.  
<https://www.youtube.com/watch?v=f3oSO1dDkiw>
- [4] Gravekamp, T. (n.d.). STM32F103C8T6 - Blue Pill. STM32-base Project.  
<https://stm32-base.org/boards/STM32F103C8T6-Blue-Pill.html>
- [5] Espressif Systems. (n.d.). ESP32Series Datasheet. 2.4 GHz Wi-Fi + Bluetooth® + Bluetooth LE SoC. <https://www.espressif.com/>
- [6] September 2023. Datasheet. Medium-density performance line Arm®-based 32-bit MCU with 64 or 128 KB Flash, USB, CAN, 7 timers, 2 ADCs, 9 com. Interfaces.  
<https://www.st.com/>