BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

ĐỒ ÁN MÔN HỌC 1

THIẾT KẾ VÀ THI CÔNG HỆ THỐNG MỞ CỬA BẰNG CẢM BIẾN VÂN TAY

SVTH: HỨA DUY BÌNH
MSSV: 21161286
Khóa: 2021
Ngành: CNKT Điện tử - Viễn thông
GVHD: ThS Nguyễn Ngô Lâm

Tp. Hồ Chí Minh, tháng 6 năm 2024

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
www.fhq.hcmute.edu.vn

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
--- *** ---

Tp. Hồ Chí Minh, ngày 18 tháng 06 năm 2021

NHIỆM VỤ ĐỒ ÁN MÔN HỌC

Họ và tên sinh viên: Hứa Duy Bình MSSV:21161286
Ngành: Công Nghệ Kỹ Thuật Điện tử - Viễn thông Lớp: 21161CLVT2B
Giảng viên hướng dẫn: ThS. Nguyễn Ngô Lâm
Ngày nhận đề tài: Ngày nộp đề tài:

1. Tên đề tài: Thiết kế và thi công mạch hệ thống mở cửa bằng cảm biến vân tay.
2. Các số liệu, tài liệu ban đầu:
Kiến thức cơ bản về các môn Mạch điện, Điện tử cơ bản, Vi xử lý, Arduino.
3. Nội dung thực hiện đề tài:
   - Thiết kế hệ thống
   - Lập trình cho hệ thống
   - Chỉnh sửa và kiểm tra mạch
   - Viết báo cáo
4. Sản phẩm:
Mạch điều khiển cửa tự động sử dụng cảm biến vân tay

GIẢNG VIÊN HƯỚNG DẪN

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
www.fhq.hcmute.edu.vn

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
--- *** ---

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên: Hứa Duy Bình MSSV:21161286

Ngành: Công Nghệ Kỹ Thuật Điện tử - Viễn thông

Tên đề tài: Thiết kế và thi công hệ thống mở cửa bằng cảm biến vân tay

Giáo viên hướng dẫn: ThS. Nguyễn Ngô Lâm

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

.....

3. Khuyết điểm:

.....

.....

.....

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:

.....

6. Điểm:.....(Bằng chữ:.....)

.....

Tp. *Hồ Chí Minh, ngày 18 tháng 06 năm 2024*

Giáo viên hướng dẫn

ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO
www.fhq.hcmute.edu.vn

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
--- *** ---

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên: Hứa Duy Bình MSSV:21161286

Ngành: Công Nghệ Kỹ Thuật Điện tử - Viễn thông

Tên đề tài: Thiết kế và thi công hệ thống mở cửa bằng cảm biến vân tay

Họ và tên Giáo viên phản biện:

.....

.....

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

.....

3. Khuyết điểm:

.....

.....

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:

.....

6. Điểm:……(Bằng chữ.....)

.....

Tp. *Hồ Chí Minh, ngày 15 tháng 06 năm 2024*

Giáo viên phản biện

LỜI CẢM ƠN

Để hoàn thành báo cáo đồ án môn học 1 chuyên ngành Công nghệ Kỹ thuật Điện tử - Viễn thông trước hết em xin gửi đến quý Thầy/Cô trong khoa Đào tạo Chất lượng cao, trường Đại học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh lời cảm ơn chân thành. Đặc biệt, thầy **Nguyễn Ngô Lâm** đã tận tình hướng dẫn, giúp đỡ và tạo điều kiện thuận lợi cho em trong suốt quá trình thực hiện đồ án. Em xin gửi đến thầy lời cảm ơn chân thành và sâu sắc nhất.

Đồng thời, em cũng xin cảm ơn đến các bạn bè đã hỗ trợ, đóng góp ý kiến cũng như chia sẻ kinh nghiệm để em hoàn thành tốt đề tài.

Mặc dù đã cố gắng hết sức, nhưng do lượng kiến thức còn eo hẹp nên không tránh khỏi những thiếu sót. Do vậy, em rất mong nhận được sự góp ý quý báu của Thầy/Cô để có thể hoàn thiện và tốt hơn nữa cũng như tích lũy kinh nghiệm để hoàn thành tốt báo cáo đồ án môn học 2 và báo cáo đồ án tốt nghiệp sau này.

Sau cùng, em kính chúc quý thầy cô thật dồi dào sức khỏe, luôn tràn đầy nhiệt huyết cùng với thành công trong sự nghiệp cao quý.

Em xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC CÁC BẢNG BIỂU

DANH MỤC CÁC HÌNH ẢNH

CHƯƠNG 1: TỔNG QUAN

1.1. GIỚI THIỆU

Hiện nay, công nghệ sinh trắc học đang không ngừng phát triển và được ứng dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là công nghệ nhận dạng vân tay. Với tính bảo mật cao, công nghệ này ngày càng trở nên phổ biến và được tích hợp vào nhiều thiết bị điện tử. Song song đó, các ứng dụng điều khiển thiết bị qua tin nhắn điện thoại và gọi điện cũng đang được phát triển mạnh mẽ nhờ vào tính tiện dụng và tiết kiệm chi phí của chúng.

Với mục đích tiếp cận các công nghệ đang phát triển này, thực hiện đồ án mong muốn chế tạo ra một mô hình hệ thống cửa thông minh. Hệ thống này sẽ cho phép đóng mở cửa bằng cách quét vân tay.

Đóng mở bằng cách sử dụng động cơ Servo và công tắc hành trình. Người dùng dễ dàng tương tác sử dụng thông qua cảm biến vân tay, các nút nhấn, hiển thị phần trăm pin có hướng dẫn cách sử dụng rõ ràng.

1.2. MỤC TIÊU NGHIÊN CỨU

Tìm hiểu và nghiên cứu về kit Arduino, module cảm biến vân tay R307 và cách kết nối giữa các module để thành mô hình hoàn thiện.

Xây dựng hệ thống quét dấu vân tay để điều khiển đóng mở cửa

Có thể đăng kí thêm hoặc xóa vân tay đã có để điều khiển đóng mở cửa. .

Xây dựng hệ thống có thể hiển thị phần trăm pin còn lại trên led 7 đoạn.

Có thể phát âm thanh thành công hoặc thất bại khi không đúng vân tay đã lưu.

Nghiên cứu, thực hành các thao tác kĩ thuật điện tử cơ bản (lắp ráp, test mạch, mô phỏng, thiết kế...).

1.3. ĐỐI TƯỢNG NGHIÊN CỨU

- Arduino Nano
- Cảm biến vân tay R307
- Led 7 đoạn Anode chung và Ic dịch 74HC595
- Buzzer

- Module Lcd I2C 16x2

1.4. BỐ CỤC ĐỒ ÁN

Chương 1: Tổng Quan:: Chương này trình bày về đặt vấn đề dẫn nhập lý do chọn đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bố cục đồ án

Chương 2: Cở sở lý thuyết:. Trong chương này trình bày về các lý thuyết có liên quan đến các vấn đề mà đề tài sẽ dùng để thực hiện thiết kế, thi công cho đề tài.

Chương 3: Tính Toán Và Thiết Kế: Chương này giới thiệu tổng quan về các yêu cầu của đề tài mà mình thiết kế và các tính toán, thiết kế gồm những phần nào. Như: thiết kế sơ đồ khối hệ thống, sơ đồ

Chương 4: Thi công hệ thống: Chương này trình bày về quá trình vẽ mạch in lắp ráp các thiết bị, đo kiểm tra mạch, lắp ráp mô hình. Thiết kế lưu đồ giải thuật cho chương trình và viết chương trình cho hệ thống. Hướng dẫn quy trình sử dụng hệ thống.

Chương 5: Kết quả nhận xét đánh giá: Chương này trình bày về những kết quả mà đồ án đạt được, những hạn chế, từ đó rút ra kết luận và hướng phát triển để giải quyết các vấn đề tồn đọng để đồ án hoàn thiện hơn

Chương 6: Kết Luận Và Hướng Phát Triển Chương này trình bày về những kết quả mà đồ án đạt được, những hạn chế, từ đó rút ra kết luận và hướng phát triển để giải quyết các vấn đề tồn đọng để đồ án hoàn thiện hơn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. CẢM BIẾN VÂN TAY

2.1.1. Định nghĩa:

Cảm biến vân tay hay còn có tên gọi khác là cảm biến nhận diện vân tay là một dạng lưu trữ vân tay của người dùng bằng công nghệ sinh trắc học với những loại sóng khác nhau, sau đó lưu lại những bề mặt lồi lõm và cả lớp da của tay để lưu lại nhằm đảm bảo tính bảo mật khi người dùng đăng nhập vào thiết bị.

Từ xa xưa, con người đã nhận ra mỗi cá nhân đều có một vân tay riêng nhưng chưa có một cơ sở khoa học nào để nghiên cứu và nhận dạng. Nhưng đến thế kỷ 16, các kỹ thuật vân tay khoa học hiện đại đã xuất hiện và từ đó các lí thuyết và chương trình mô tả, nhận dạng vân tay mới phát triển mau chóng. Năm 1888, Francis Galton giới thiệu các đặc trưng chi tiết phục vụ cho đối sánh vân tay. Nhưng đến đầu thế kỉ 20, nhận dạng vân tay chính thức được chấp nhận như một phương pháp nhận dạng cá nhân có giá trị và trở thành tiêu chuẩn trong pháp luật. Ví dụ, năm 1924 FBI đã thiết lập một cơ sở dữ liệu có 810.000 thẻ vân tay.

2.1.2. Nguyên lý hoạt động

Nguyên lý hoạt động của module cảm biến vân tay cơ bản có 2 phần:

- Lấy dữ liệu hình ảnh của vân tay: Khi lấy dữ liệu, người dùng cần phải thực hiện quét dấu vân tay hai lần thông qua cảm biến quang học. Hệ thống sẽ tiến hành thuật toán xử lý hình ảnh của 2 lần quét vân tay, tạo ra một khuôn mẫu của các vân tay dựa trên kết quả xử lý và lưu trữ lại các bản mẫu.

- So sánh dấu vân tay (có thể theo chế độ 1:1 hoặc theo 1:N): Khi người dùng thực hiện quét dấu vân tay, module sẽ chụp lại dữ liệu hình ảnh vân tay và so sánh với các mẫu vân tay đã được lưu trữ sẵn trong thư viện. Đối với 1:1, hệ thống sẽ so sánh trực tiếp vân tay với mẫu được chỉ định cụ thể trong module; đối với 1:N, hoặc tìm kiếm, hệ thống sẽ tìm kiếm trong thư viện để tìm vân tay phù hợp. Sau đó trả về kết quả đúng nếu trùng khớp hoặc kết quả sai nếu không trùng khớp dữ liệu đã được lưu trữ.

2.1.3. Các đặc tính

Module tích hợp nhiều loại chip xử lý trong cùng 1 module: cảm biến dấu vân tay quang học, bộ vi xử lý DSP tốc độ cao, bộ nhớ PLASH…

Dễ dàng sử dụng với các tính năng bảo mật cao, thông minh. Mức độ bảo mật điều chỉnh được: thích hợp cho các ứng dụng khác nhau, mức độ bảo mật có thể được thiết lập điều chỉnh bởi người sử dụng.

Người dùng có thể tiến hành phát triển kết hợp với các module khác để làm ra một loạt các sản phẩm cuối cùng, chẳng hạn như: kiểm soát quyền truy cập, điểm danh vào lớp học hoặc chấm công, két an toàn, khóa cửa nhà hay cửa xe…

Tiêu thụ điện năng thấp, giá thành không cao, kích thước nhỏ gọn, hiệu năng tuyệt vời. Khả năng chống tĩnh điện mạnh mẽ, chỉ số chống tĩnh điện đạt 15KV trở lên. Khả năng xử lý hình ảnh tốt, có thể chụp được hình ảnh có độ phân giải lên đến 500 dpi.

2.1.3. Các thông số kĩ thuật

Điện áp cung cấp: DC 4.2 ~ 6.0V.

Dòng cung cấp:

+ Dòng làm việc bình thường: 40 mA.

+ Dòng đỉnh: 150 mA.

Thời gian thu thập hình ảnh: < 0.5 giây.

Kích thước cửa sổ quét: 18x22 mm.

Chế độ quét:

+ So sánh với một mẫu duy nhất (1:1).

+ Tìm kiếm và so sánh với mẫu lưu trong bộ nhớ (1: N).

Bộ nhớ lưu trữ mẫu: 512 bytes.

Mức độ an toàn: năm (từ thấp đến cao: 1, 2, 3, 4, 5 (cao nhất)).

Tỷ lệ lỗi chấp nhận nhầm (FAR): < 0,001.

Tỷ lệ từ chối nhầm (FRR): < 1.0%.

Thời gian tìm kiếm: < 1.0 giây (1: 1000, trung bình).

Giao tiếp với máy tính: UART (TTL mức logic)

Tốc độ truyền thông tin liên lạc (UART): (9600 x N) bps đó N = 1 ~ 12 (giá trị

mặc định N = 6, tức là 57600bps)

Môi trường làm việc:

+ Nhiệt độ: -20 ℃ ~ + 40 ℃

+ Độ ẩm tương đối: 10% - 85%

Môi trường bảo quản:

+ Nhiệt độ: -40 ℃ ~ + 85 ℃

+ Độ ẩm tương đối: <85%

2.1.4. Giao tiếp phần cứng

Số chân	Tên chân	Kiểu vào / ra	Chức năng
1	Vt	In	Năng lượng cần để nhận diện vân tay (DC 4.2~ 6V, 5uA) (dây đỏ).
2	Vin	In	Cấp nguồn cho module (dây đen).
3	TXD	Out	Dữ liệu đầu ra. Kiểu TTL logic (dây vàng).
4	RXD	In	Dữ liệu đầu vào. Kiểu TTL logic (dây xanh lá).
5	GND	-	Dây nối đất (dây xanh dương).
6	Touch	Out	Xuất tín hiệu nhận diện dấu vân tay (dây trắng).

Bảng 2. 1: Các chân kết nối của module R308

Giao tiếp phần cứng của module R307 được thể hiện qua bảng 2.1

2.1.5. **Kiểm tra và xác nhận gói dữ liệu**

Lưu ý: lệnh chỉ được gửi từ VXL đến cảm biến, cảm biến chỉ trả về các gói xác nhận.

Định nghĩa bytes xác nhận: -

0x00h: thực thi hoàn tất.

0x01h: lỗi nhận dữ liệu.

0x02h: không phải vân tay.

0x03h: thất bại đăng ký vân tay.

0x06h: không tạo được đặt điểm nhân dạng.

0x07h: dấu vân quá nhỏ để lấy mẫu.

0x08h: dấu vân không trùng.

0x09h: thất bại tìm kiếm dấu vân tay.

0x0Ah: lỗi kết hợp đặc điểm dấu vân tay.

0x0Bh: đại chỉ ID vượt khung.

0xCh: lỗi đọc từ dữ liệu vân tay. Dữ liệu xấu.

0xDh: lỗi nạp dữ liệu.

0xEh: không thể nhận dữ liệu

0xFh: lỗi gửi hình ảnh.

0x10h: lỗi xoá dữ liệu.

0x11h: lỗi xoá một ID.

0x15h: lỗi tạo ảnh.

0x18h: lỗi ghi flash.

0x19h: không xác định được lỗi.

0x1Ah: số đăng ký không hợp lệ.

0x1Bh: gói dữ liệu sai.

0x1Ch: sai số trang.

0x1Dh: lỗi cổng giao tiếp.

others: dự phòng.

2.1.6. Giao thức truyền gói dữ liệu

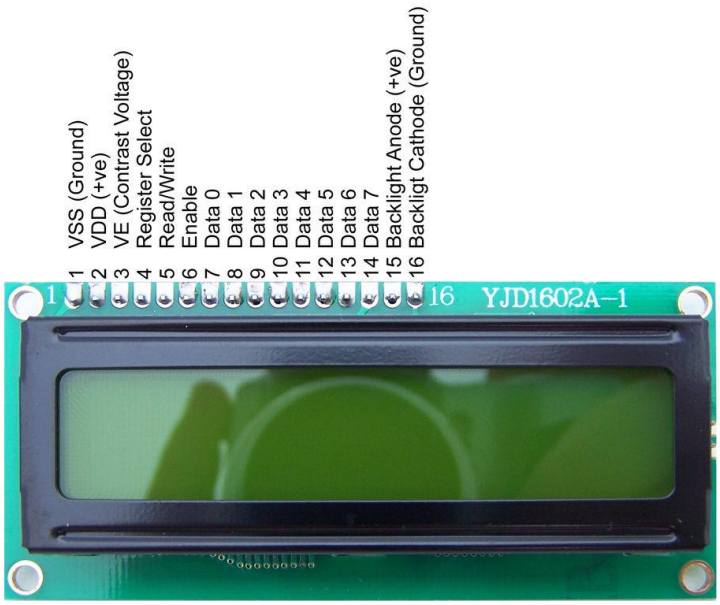
Khi module R308 thực hiện việc giao tiếp, truyền và nhận các câu lệnh/ dữ liệu/ kết quả thì tất cả được gói trong một định dạng gói dữ liệu được biểu diễn qua bảng 2.2:

Bảng 2. 2.. Định dạng gói dữ liệu

Tên	Ký hiệu	Độ dài	Mô tả
Header	START	2 bytes	Có 2 byte được truyền đầu tiên trong gói dữ liệu. Được mặc định giá trị 0xEF01.
Adder	ADDER	4 bytes	Có 4 byte địa chỉ của module. Giá trị mặc định ban đầu là 0XFFFFFFFF, nhưng nó có thể được sửa đổi bởi lệnh. Byte cao sẽ được chuyển vào đầu tiên và nếu giá trị adder sai, module sẽ từ chối để chuyển.
Package indentifier	PID	1 bytes	Định dạng loại gói dữ liệu 0x01 : Gói lệnh. 0x02 : Gói dữ liệu. 0x07 : Gói xác nhận. 0x08 : Gói kết thúc dữ liệu.
Package length	LENGTH	2 bytes	Chiều dài gói dữ liệu tính từ Package content đến Checksum. Đơn vị chiều dài là byte (tối đa 256 bytes).
Package contents	DATA	-	Nội dung dữ liệu. Có thể là lệnh, dữ liệu, kết quả được xác nhận…(như giá trị ký tự dấu vân tay).
Checksum	SUM	2 bytes	Là tổng số học của Package identifier, Package length, Package content. Bit tràn được bỏ qua, bit cao được truyền đầu tiên.

2.2. Màn hình LCD và module I2C

- **LCD 16x2**

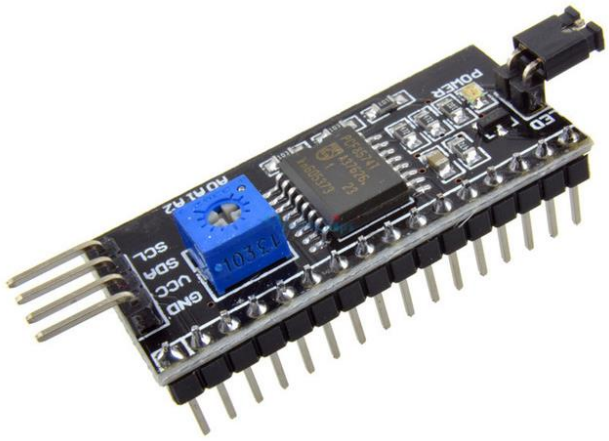


Hình 2. 1. Màn hình lcd 16x2

- Chân cấp nguồn: Vss nối mass (0V), VDD (nối nguồn +5V), V0 (điều chỉnh độ tương phản thường nối với biến trở).
- RS: Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi. + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read). + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.
- R/W: Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.
- E: Chân cho phép chốt xung kí tự (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào (chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.
- D0-D7: Chân dữ liệu dùng để trao đổi dữ liệu giữa thiết bị điều khiển và LCD.

- A, K: Chân điều khiển đèn nền.

- **Module I2C**



Hình 2. 2. Module I2C

LCD có quá nhiều nhiều chân gây khó khăn trong quá trình đấu nối và chiếm dụng nhiều chân trên vi điều khiển.

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 bạn chỉ cần tốn 2 chân (SCL, SDA) để kết nối.

Ưu điểm

Tiết kiệm chân cho vi điều khiển.

Dễ dàng kết nối với LCD.

Thông số kĩ thuật

Điện áp hoạt động: 2.5-6V DC.

Hỗ trợ màn hình: LCD1602,1604,2004 (driver HD44780).

Giao tiếp: I2C.

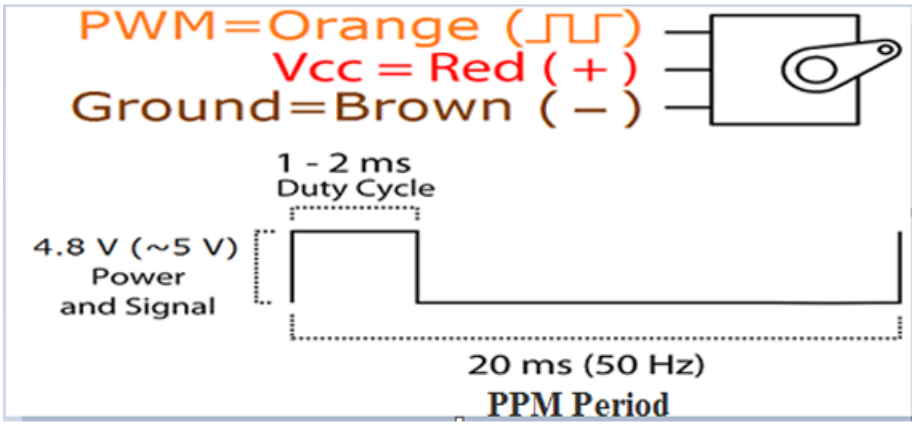
Địa chỉ mặc định: 0X27 (có thể điều chỉnh bằng ngắn mạch chân A0/A1/A2).

Tích hợp biến trở xoay điều chỉnh độ tương phản cho LCD.

2.3. Giới thiệu động cơ Servo SG90 9g Micro

Động cơ RC Servo 9G có kích thước nhỏ, là loại được sử dụng nhiều nhất để làm các mô hình nhỏ hoặc các cơ cấu kéo không cần đến lực nặng, động cơ RC Servo 9G có tốc độ

phản ứng nhanh, các bánh răng được làm bằng nhựa nên cần lưu ý khi nâng tải nặng vì có thể làm hư bánh răng, động cơ RC Servo 9G có tích hợp sẵn Driver điều khiển động cơ bên trong nên có thể dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM.



Hình 2. 3. Điều khiển servo bằng phương pháp điều độ rộng xung

Thông số kỹ thuật:

- Điện áp hoạt động: 4.8-5VDC
- Tốc độ: 0.12 sec/ 60 degrees (4.8VDC)
- Lực kéo: 1.6KG.CM
- Kích thước: 21x12x22mm
- Trọng lượng: 9g..

2.4. LED 7 đoạn và IC dịch 74HC595

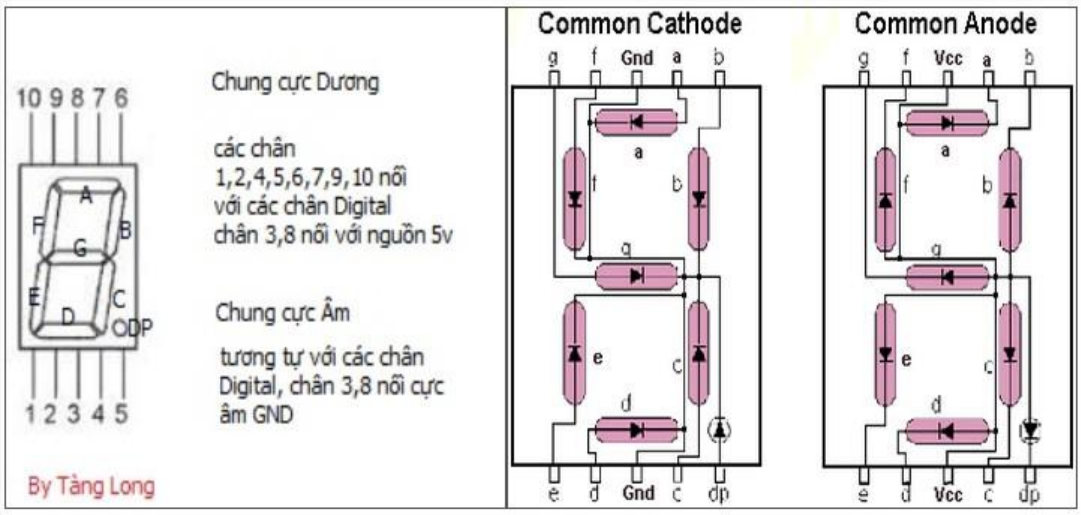
- **LED 7 Đoạn**

Cấu tạo**:** LED 7 đoạn bao gồm 8 LED được kết nối song song để có thể thắp sáng hiển thị số "0, 1, 2, 3, 4, 5, 7, 8, 9, A, b, C, d, E, F, …".Mỗi đoạn Led được đánh dấu từ A tới G.Đoạn thứ tám gọi là "chấm thập phân" (Decimal Point) ký hiệu DP được sử dụng khi hiển thị số không phải là số nguyên

Phân loại LED 7 đoạn: Dựa vào các cực được nối có thể phân loại như sau:

Loại dương chung (Common Anode): nếu cực dương (anode) của tất cả 8 LED được nối với nhau và các cực âm (cathode) đứng riêng lẻ.

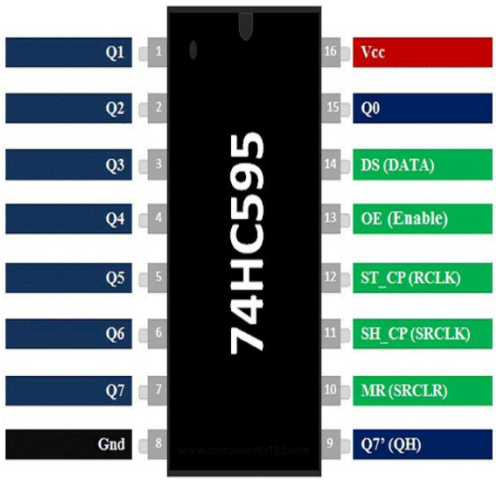
Loại âm chung (Common Cathode): nếu cực âm (cathode) của tất cả 8 LED được nối với nhau và các cực dương (anode) đứng riêng lẻ.



Hình 2. 4. LED 7 đoạn

- **IC 74HC595**

HC595 là một thanh ghi dịch (shift register) hoạt động trên giao thức nối tiếp vào song song ra (Serial IN Parallel OUT). Được sử dụng trong dự án này vì không có đủ chân GPIO trên vi điều khiển để kiểm soát số lượng đầu ra cần thiết. Nó nhận dữ liệu nối tiếp từ vi điều khiển và sau đó gửi dữ liệu này qua các chân song song.



Hình 2. 5. Sơ đồ chân 74HC595

Có thể tăng 8 chân đầu ra bằng cách sử dụng chip đơn.

Cũng có thể kết nối song song nhiều hơn 1 thanh ghi dịch.

Giả sử đã kết nối ba thanh ghi dịch với bộ vi điều khiển, các chân đầu ra được tăng lên 8 x 2 = 24.

Bảng 2. 3. Sơ đồ chân ic 74HC595

Số chân	**Tên chân**	**Mô tả**
1, 2, 3, 4, 5, 6, 7	Chân output (Q1 đến Q7)	74hc595 có 8 chân đầu ra, trong đó có 7 chân này. Chúng có thể được kiểm soát nối tiếp
8	Ground	Nối đất
9	(Q7') Serial Output	Chân này được sử dụng để kết nối nhiều hơn một 74hc595 dưới dạng xếp tầng
10	(MR) Master Reset	Reset tất cả các đầu ra ở mức thấp. Phải giữ ở mức cao để hoạt động bình thường
11	(SH_CP) Clock	Đây là chân đồng hồ mà tín hiệu đồng hồ phải được cung cấp từ vi điều khiển hoặc vi xử lý
12	(ST_CP) Latch	Chân Latch dùng để cập nhật dữ liệu vào các chân đầu ra. Nó kích hoạt mức cao
13	(OE) Output Enable	Chân OE được sử dụng để tắt đầu ra. Phải giữ ở mức thấp để hoạt động bình thường
14	(DS) Serial Data	Đây là chân mà dữ liệu được gửi đến, dựa trên đó 8 đầu ra được điều khiển
15	(Q0) Output	Chân đầu ra đầu tiên
16	Vcc	Chân này cấp nguồn cho IC, thường sử dụng + 5V

2.5. Buzzer

Tính Năng: Buzzer hay còn gọi là còi chíp hoặc còi xung, là thiết bị phát ra âm thanh (tiếng bíp bíp) hay dùng trong các mạch điện tử.

Cấu tạo của Buzzer gồm 2 chân, chân dài là chân (+) và chân ngắn là chân (-). Trong quá trình sử dụng cần chú ý mắc đúng chân để tránh làm hỏng buzzer.

Ứng dụng này sử dụng Buzzer 5V, có phát ra âm thanh có tần số tối đa 2.5kHz.

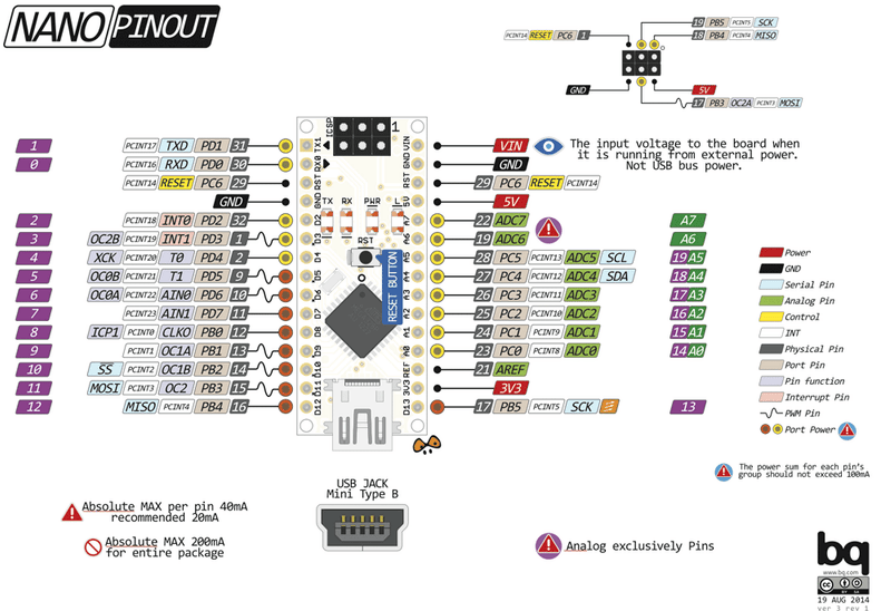
Thông Số Kỹ Thuật

- Điện áp làm việc: 5V.
- Kích thước: 9.5 x 12 mm.

Hình 2. 6. Buzzer

2.6. Khối xử lý trung tâm

Mạch arduino Nano CH340: Arduino Nano USB Mini là board mạch sử dụng vi điều khiển ATmega328P tích hợp các chân I/O đơn giản nhỏ gọn dựa trên mã nguồn mở được phát triển bởi Arduino, có lợi thế lớn về kích thước so với phiên bản Arduino Uno và Arduino Mega. Arduino Nano có thể hoạt động độc lập và tương tác hiệu quả với các thiết bị điện tử, cũng có thể giúp những người mới tìm hiểu về Arduino có thể kết nối với PC, phối hợp với Flash, Xử lý, Max / Msp, PD, và các phần mềm khác một cách dễ dàng. Điều này giúp Arduino Nano là sự lựa chọn ưa thích khi muốn thực hiện một projects mà yêu cầu kết nối với các thiết bị ngoại vi ít và đơn giản. Các chức năng rất giống giống với phiên bản Arduino Uno nhưng kích thước nhỏ gọn hơn.



Hình 2. 7. Arduino Nano

Bảng 2. 4. Thông số kĩ thuật Arduino Nano

STT	Thông số	Chuẩn	Đơn vị
1.	Điện áp hoạt động (mức logic)	5	V
2.	Điện áp đầu vào	7 - 12	V
3.	Điện áp đầu vào (giới hạn)	6 - 20	V
4.	Các chân I/O số (digital I/O)	14 (6 chân PWM)	chân
5.	Chân đầu vào tương tự	8	chân
6.	Dòng DC trên mỗi chân I/O	40	mA
7.	Bộ nhớ Flash	32	KB
8.	SRAM	2	KB
9.	EEPROM	1	KB
10.	Tốc độ xung clock	16	MHz
11.	Dòng ra tối đa (5V)	500	mA
12.	Dòng ra tối đa (3.3V)	50	mA

Bảng 2. 5. Bảng thông số các chân của Arduino Nano

Pin No.	Name	Type	Description
1-2, 5-16	D0 – D13	I/O	Ngõ vào/ra số từ port 0-13
3, 28	Reset	Input	Reset
4, 29	GND	PWR	Ground
17	3V3	Output	Ngõ ra +3,3V
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Ngõ vào Analog kênh 0-7
27	+5V	Output/Input	+ 5V từ nguồn cung cấp bên ngoài
30	VIN	PWR	Cung cấp điện thế (Supply voltage)

CHƯƠNG 3: TÍNH TOÁN VÀ THIẾT KẾ

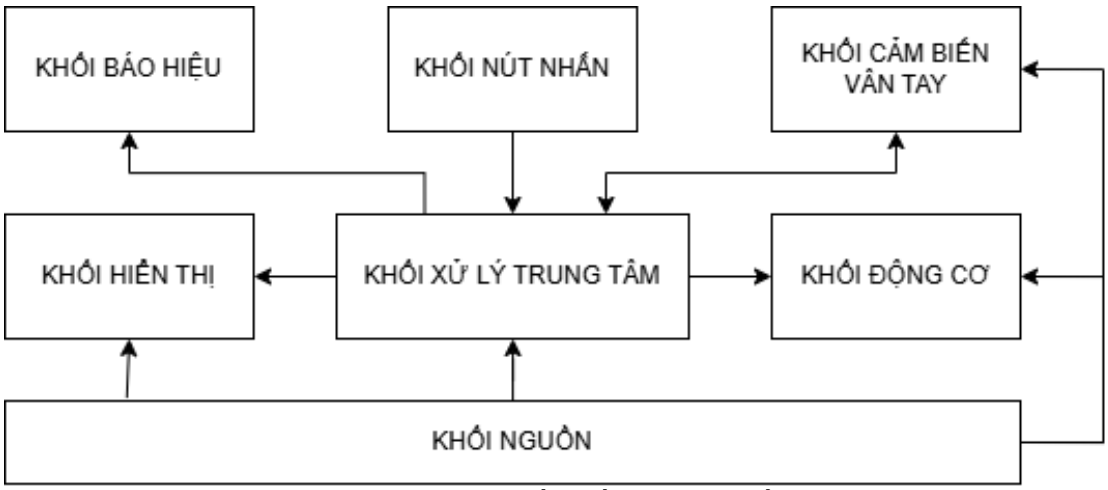
3.1. YÊU CẦU VÀ SƠ ĐỒ KHỐI CỦA HỆ THỐNG

3.1.1. Yêu cầu của hệ thống

Hệ thống có các chức năng sau:

- Nhận diện dấu vân tay đã lưu để mở cửa
- Đăng kí thêm dấu vân tay mới
- Xóa dấu vân tay cũ
- Hiển thị chỉ dẫn cho người dùng trên LCD
- Hiển thị phần trăm pin còn lại của mạch

3.1.2. Sơ đồ khối và chức năng của mỗi khối



Hình 3. 1. Sơ đồ khối của hệ thống

Chức năng từng khối

Khối nguồn: cung cấp nguồn cho các khối cảm biến vân tay, khối xử lý trung tâm và khối động cơ, khối báo hiệu

Khối cảm biến vân tay: quét nhận dấu vân tay để đóng mở cửa và nơi lưu trữ dấu vân tay. Khối này do module cảm biến vân tay R307 thực hiện.

Khối xử lý trung tâm: thu thập dữ liệu từ các thiết bị sau đó xử lý và điều khiển khối chấp hành và khối hiển thị. Khối này do Arduino Nano thực hiện.

Khối động cơ: sử dụng để đóng mở chốt cửa do động cơ servo 9g thực hiện.

Khối nút nhấn: sử dụng để đăng kí thêm dấu vân tay hoặc xóa dấu vân tay đã lưu

Khối hiển thị: hiển thị phần trăm pin trên led 7 đoạn, và lcd để chỉ trạng thái hoạt động của thiết bị

Khối báo hiệu: tiếp nhận dữ liệu từ khối xử lý trung tâm để báo hiệu đúng hay sai khi nhận vân tay, do buzzer thực hiện

3.1.3. Hoạt động của hệ thống

Đăng kí vân tay: Nhấn nút Enroll để đăng kí vân tay, chọn địa chỉ lưu ID vân tay bằng cách nhấn Up hoặc Down sau đó nhấn Ok. Đặt ngón tay cần đăng kí vào cảm biến cho đến khi màn hình báo Stored!.

Xóa vân tay: Nhấn nút Del để xóa vân tay, chọn địa chỉ ID mà bạn cần xóa bằng cách nhấn UP hoặc Down sau đó nhấn Ok. Màn hình báo Figer Deleted Successfully là thành công.

Mở khóa động cơ: Nhấn giữ Up/Down 2s để bắt đầu kiểm tra, lúc này đặt ngón tay vào cảm biến, nếu vân tay đúng khóa điện từ sẽ tự động mở 5s sau đó đóng, và mở buzzer kêu khóa đã mở thành công; nếu vân tay không đúng màn hình sẽ báo Finger Not Found Try Later buzzer sẽ báo không thành công và khóa động cơ sẽ không mở

Hiển thị led 7 đoạn: hiện thị phần trăm pin, hiển thị địa chỉ id lên led 7 đoạn khi đang ở trong chế độ đăng kí vân tay hoặc chế độ xóa vân tay

3.2. THIẾT KẾ HỆ THỐNNG PHẦN CỨNG

3.2.1. Khối cảm biến vân tay

Hình 3. 2. Cảm biến vân tay R307

Cảm biến vân tay R307 trên hình 3.2 tích hợp xử lý hình ảnh và thuật toán xử lý trên cùng một chip. Khả năng xử lý ảnh chụp tốt với độ phân giải lên đến 500dpi. Chuẩn giao tiếp: USB - UART (TTL logical logic) từ 9600 – 115200bps, sử dụng tốc độ mặc định là 57600 bps đảm bảo truyền nhận chính xác dữ liệu. Tiêu thụ công suất thấp,

+ Điện áp cung cấp: 3.6V ~ 6V DC.

+ Dòng điện tiêu thụ: 100mA, đỉnh 150mA.

Trong bảng 3.1 là chức năng của các chân của module cảm biến vân tay R308.

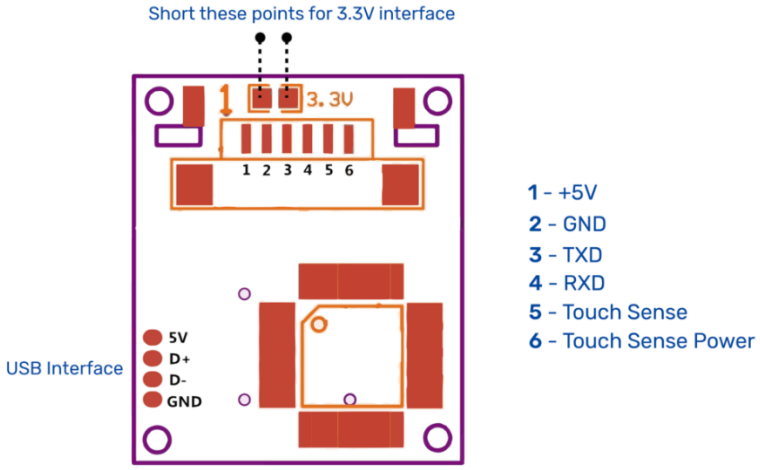
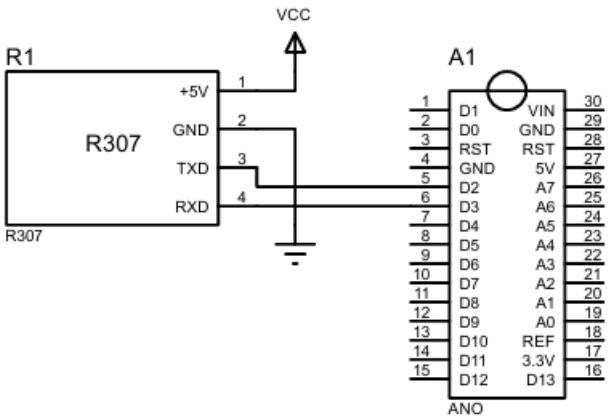


Fig. R307 Fingerprint Scanner Pinout

Hình 3. 3. Các chân cảm biến vân tay

Tiến hành kết nối cảm biến với Arduino theo sơ đồ sau:



Hình 3. 4. Sơ đồ kết nối chân giữ r307 với arduino nano

3.2.2. Khối hiển thị

Để hiển thị thông tin từ từ khối trung tâm sau khi xử lý

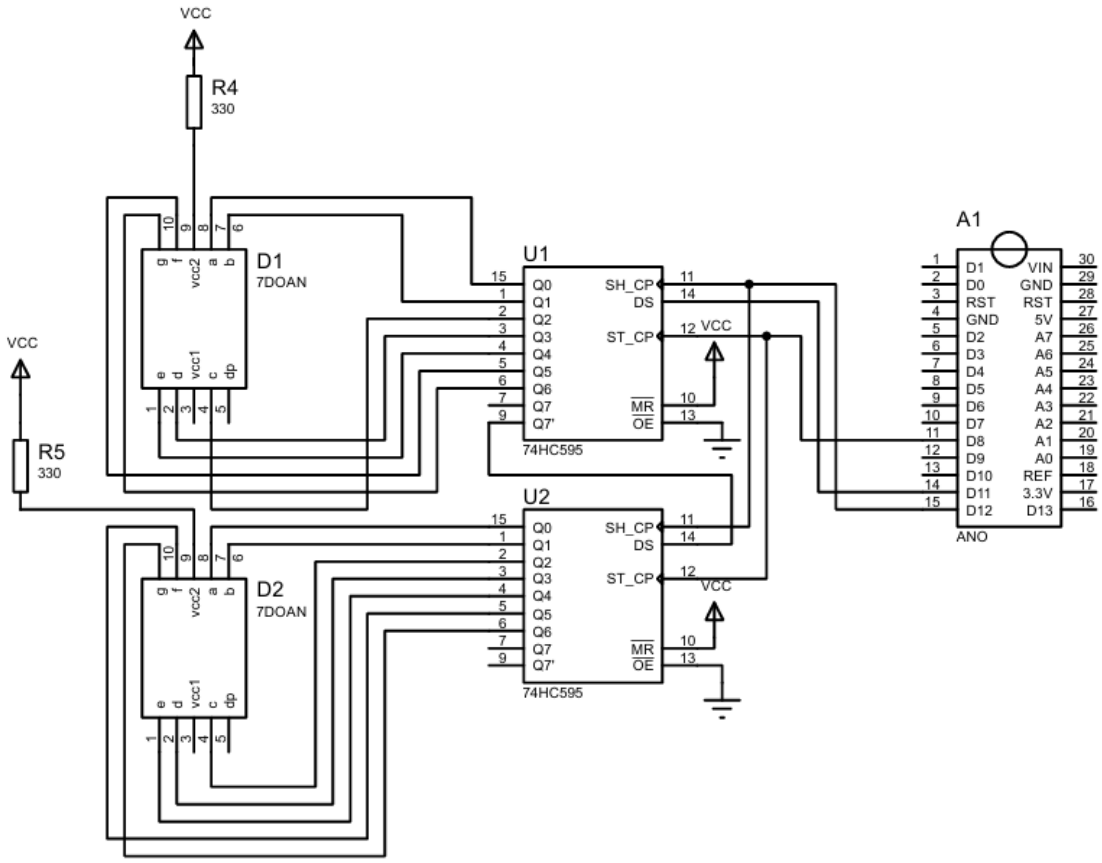
- LED 7 đoạn

Hiển thị phần trăm pin của mạch nguồn. Do để hiển thị 2 led 7 đoạn ta sẽ cần tới 20 chân digital nối với vi điều khiển để điều khiển cho 2 led hoạt động. Giải pháp là sử dụng phương pháp ShiftOut có thể rút gọn lại còn 3 chân digtal để kết nối với vi điều khiển.

ShiftOut là việc gửi tín hiệu cho 1 IC có hỗ trợ shiftOut (ví dụ HC 595 này), cứ mỗi lần gửi nó gửi 1 byte (không hơn không kém), mỗi 1 bit (có tổng cộng 8 bit trong 1 byte) sẽ quản lý giá trị điện tại chân tín hiệu của HC 595 (các chân có tên là Q0-Q7).

Cứ mỗi lần shiftOut, thì byte đầu tiên sẽ đến IC HC 595 cuối cùng, byte thứ hai sẽ đến IC HC595 thêm trước IC HC595 đó

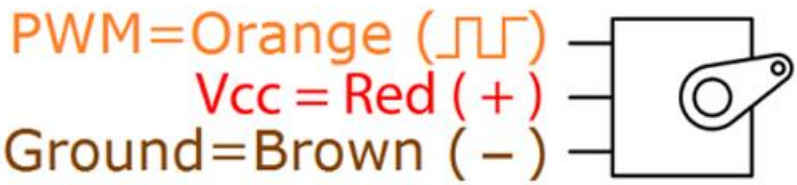
- GND (pin 8) nối đến cực âm
- Vcc (pin 16) nối đến chân 5V
- OE (pin 13) nối đến cực âm
- MR (pin 10) nối đến chân 5V
- DS (pin 14) đến Arduino DigitalPin 11
- SH_CP (pin 11) đến Arduino DigitalPin 12
- ST_CP (pin 12) đến Arduino DigitalPin 8



Hình 3. 5. Sơ đồ kết nối Led 7 đoạn anode chung kết hợp với 74HC595

3.3.3. Khối động cơ

Điều khiển và kết nối Sơ đồ chân kết nối với động cơ Servo 9g biểu diễn qua hình 3.5 bên dưới.

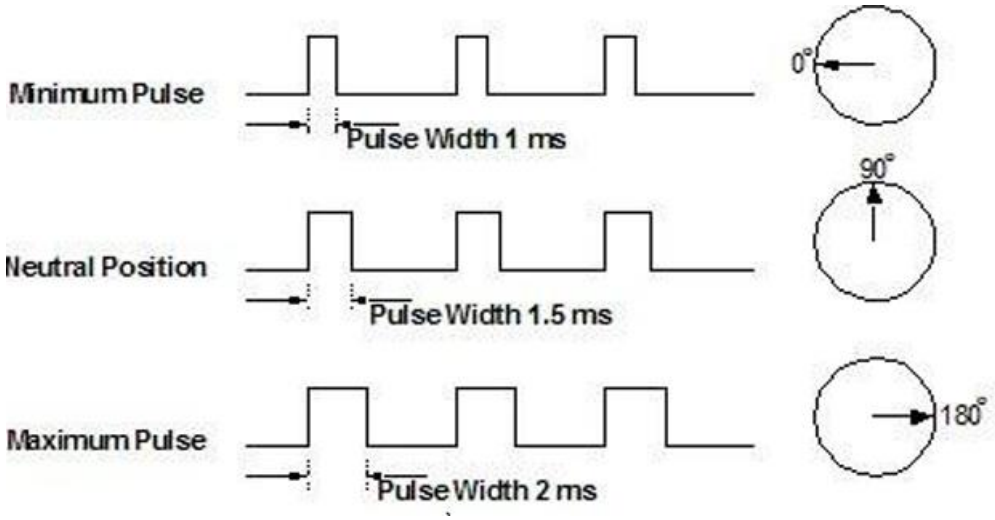


Hình 3. 6. Sơ đồ chân động cơ servo 9g

Servo là một loại động cơ điện đặc biệt. Không giống như những động cơ thông thường cứ cắm điện vào là chạy liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoản 0o - 180o

Động cơ servo được thiết kế những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ được nối với một mạch điều khiển. Khi động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bất kỳ lý do gì nào ngăn cản chuyển động quay của dộng cơ, cơ cấu hồi tiếp sẽ tiếp nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển sẽ tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác.

Kết nối dây màu đỏ với 5V, dây màu nâu với 0V, dây màu cam với chân phát xung (PPM) của vi điều khiển như hình 3.5. Tùy vào góc quay mà ta cấp thời gian xung ở mức cao tương ứng. Phương điều chế PPM (Pulse Position Modulation) là phương pháp điều chỉnh điện áp ra tải hay nói cách khác là vị trí xung thay đổi theo biên độ tín hiệu tương tự trong một khe thời gian. Đặc tính của xung PPM: - Tần số thông thường có giá trị trong khoảng 50Hz (20 ms) - Thời gian xung ở mức cao chỉ từ 1ms đến 2ms. - Có thể có nhiều hơn 1 sự thay đổi trạng thái điện cao/thấp.



Hình 3. 7. Điều chỉnh động rộng xung

3.3.4. Khối nguồn

Sử dụng ổn áp 7805 để hạ áp từ nguồn pin 11.1V thành 5V để phù hợp với các cảm biến của mạch. Sơ đồ nguyên lý của khối nguồn (hình 3.9)

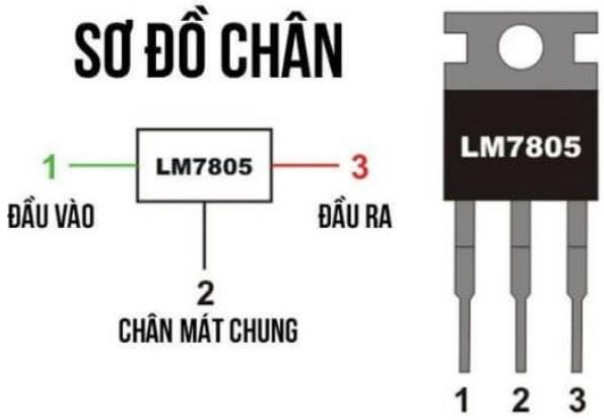
IC 7805, còn được gọi là LM7805, là một loại linh kiện điện tử thuộc dòng ổn áp dương LM78xx. Đây là một IC điều chỉnh điện áp dương, được thiết kế để cung cấp một điện áp đầu ra ổn định là 5V DC từ một nguồn điện áp đầu vào DC trong khoảng từ 7V đến 25V. IC 7805 có nhiều tính năng tích hợp, bao gồm khả năng cung cấp dòng điện đầu ra lên đến 1.5A, bảo vệ quá tải, bảo vệ quá nhiệt, và dòng điện tĩnh thấp. Nó thường được sử dụng rộng rãi trong các ứng dụng điện tử để cung cấp điện áp ổn định cho vi mạch và linh kiện khác.

- Dưới đây là thông số kỹ thuật cơ bản của IC LM7805:
- Điện áp đầu vào tối thiểu: 7V
- Điện áp đầu vào tối đa: 25V
- Điện áp đầu ra ổn định: 5V

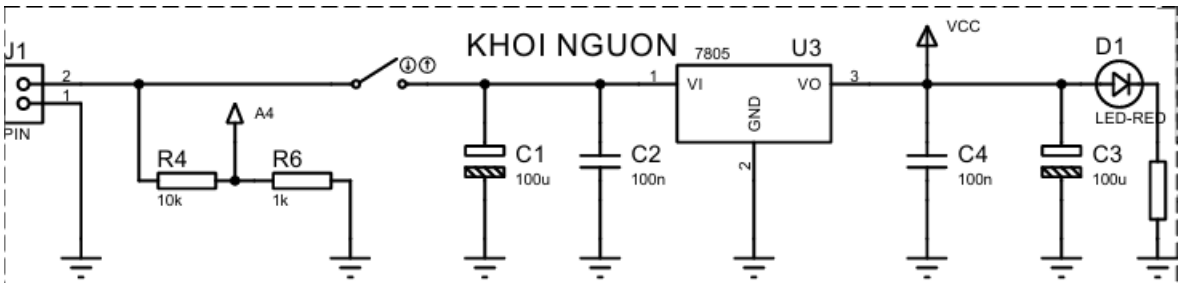
  Dòng điện đầu ra tối đa: 1.5A (1500mA)
- Chức năng bảo vệ (Protection Features):
- Bảo vệ quá tải (Overcurrent Protection): Có
- Bảo vệ quá nhiệt (Overheat Protection): Có

- Dòng điện tĩnh (Quiescent Current):
- Dòng điện tĩnh khi không có tải: Khoảng 5mA
- Nhiệt độ hoạt động (Operating Temperature Range): Từ 0°C đến +125°C
- Chế độ bảo quản (Storage Temperature Range): Từ -65°C đến +150°C
- Gói (Package):
- Gói tiêu chuẩn: TO-220
- Điện áp dropout (Dropout Voltage):
- Điện áp dropout tiêu chuẩn: Khoảng 2V (tùy thuộc vào dòng điện tải)
- Độ chính xác đầu ra (Output Voltage Accuracy): Độ chính xác đầu ra: ±5%

LM7805 thường có ba chân: đầu vào (Input), đầu ra (Output), và chân GND (Ground). Nó có thể được kết nối vào mạch điện tử bằng cách cung cấp một nguồn điện áp đầu vào DC và sẽ cung cấp một nguồn điện áp ổn định 5V DC ở đầu ra.

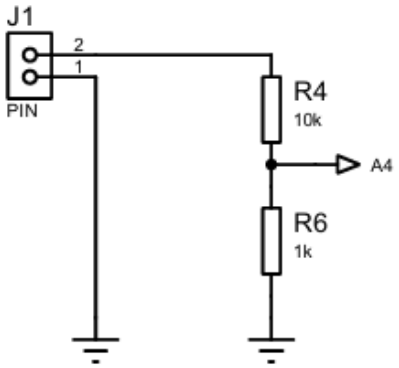


Hình 3. 8. Sơ đồ chân ổn LM7805



Hình 3. 9. Sơ đồ nguyên lý khối nguồn

- Ứng dụng mạch Thevenin để đo áp pin của nguồn



Hình 3. 10. Mạch ứng dụng thevenin để đo điện áp của pin

Áp dụng định lý Thevenin ta được công thức:

$$V_{A4} = \frac{Vcc\ .\ R6}{R4+ R6} \quad \rightarrow \quad \text{Vcc} = \frac{V_{A4}}{\frac{R6}{R4+ R6}}$$

Từ đó ta sẽ cho tín hiệu từ V_{A4} vào chân ananlog của Arduino Nano để đọc điện áp còn lại trong pin

- **Dòng tiêu thụ và điện áp các linh kiện**

- Board Arduino Nano sử dụng hết 13 chân (N).

- Dòng DC trên mỗi chân I/O I_{1C} là 40mA.

$I = I_{1C} \times N$

$= 40 \times 13 = 520$ mA

Dòng LCD I_{LCD} tiêu thụ tối đa là 50 mA.

Dòng tiêu thụ cảm biến vân tay I_{CB} là 150 mA.

Dòng tiêu thụ Servo 9g: 100mA

Dòng tiêu thụ của Led 7 đoạn anode chung: 150mA

Tổng dòng tiêu thụ cho khối nguồn là: I_A=520 + 50 + 150 + 100 + 150x2 = 1120 mA

Theo tính toán thì sẽ cấp dòng tối thiểu là 1120 mA. Thực tế chọn khối nguồn có ic LM7805 có dòng ra tối đa là 1500mA để mạch hoạt động ổn định

CHƯƠNG 4: THI CÔNG HỆ THỐNG

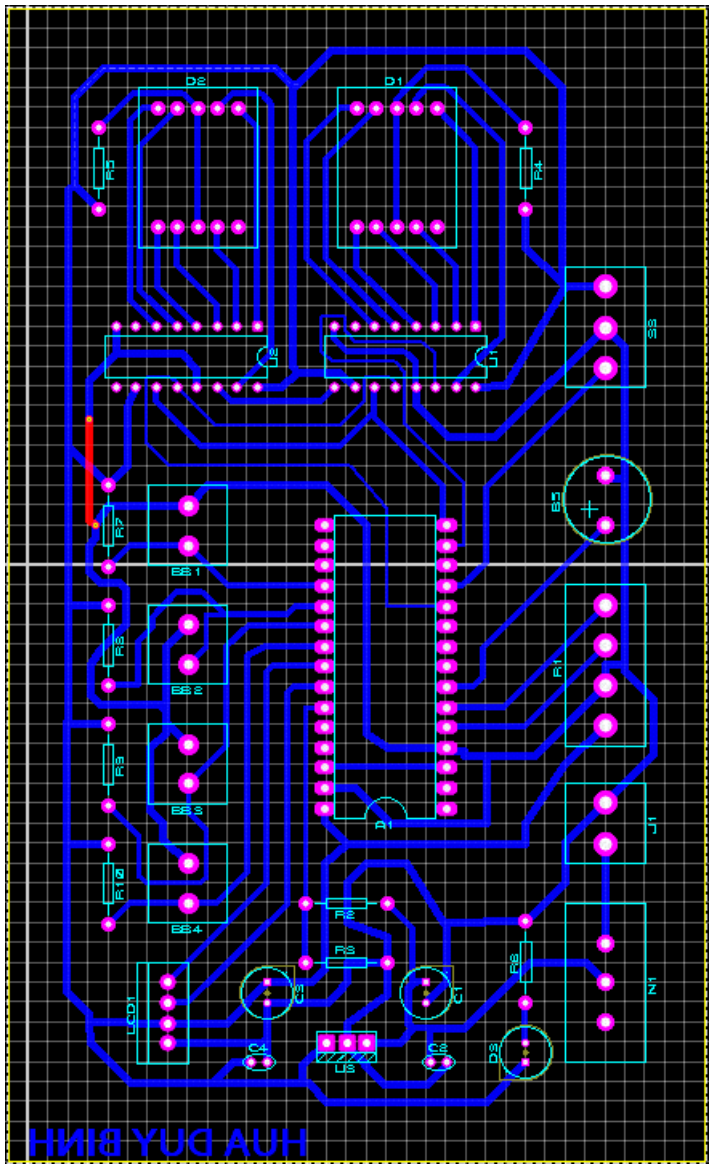
4.1. GIỚI THIỆU

Sau quá trình tính toán và thiết kế chọn các thiết bị hợp lý nay tiến hành thi công thi công PCB, lắp ráp và test mạch.

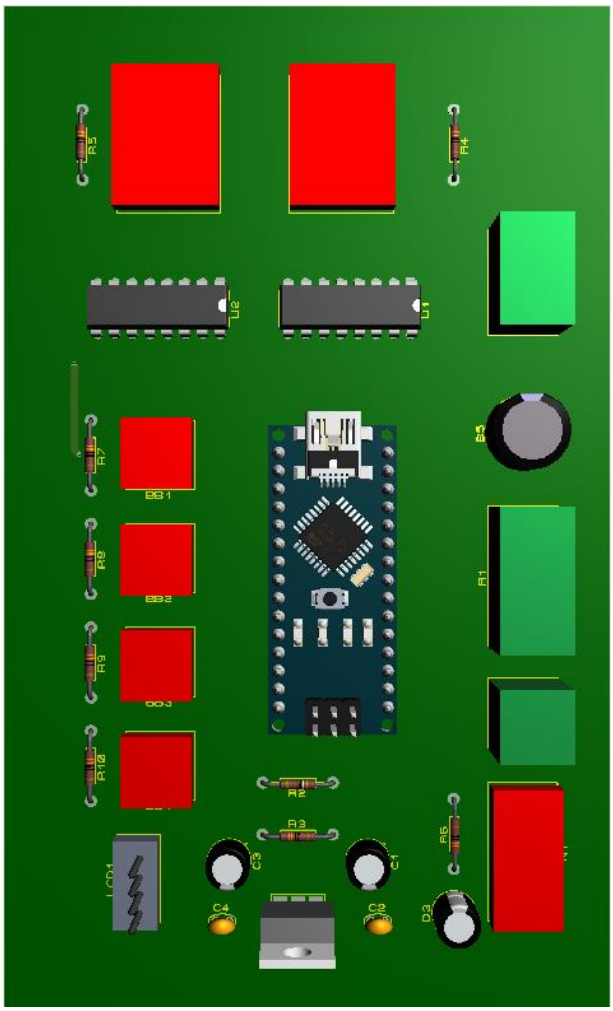
4.2. THI CÔNG HỆ THỐNG

4.2.1. Thi công bo mạch

Sau khi thiết kế xong sơ đồ nguyên lý và tiến hành vẽ mạch PCB 1 lớp thủ công. Với kích thước board là 100 x 150 mm.



Hình 4. 1. Sơ đồ đi dây lớp bottom



Hình 4. 2. Hình dạng 3D của mạch

4.2.2. Láp ráp và kiểm tra

Quy trình lắp ráp – kiểm tra mạch :

- Bước 1: In mạch ra giấy in lên miếng đồng đã được chà nhám để loại bỏ lớp đồng bị oxi hóa bên ngoài
- Bước 2: Rửa board đồng sạch sẽ bằng thuốc rửa mạch sau khi ủi mạch
- Bước 3: Tiến hành khoan lỗ và lắp linh kiện lên mạch chuẩn bị hàn
- Bước 4: Dùng đồng hồ chỉnh thang đo điện trở x1 để kiểm tra ngắn mạch trên ngõ vào từ adapter 5V và 2 chân nguồn từ adduino cắm vào. Kiểm tra GND giữa adapter 5V và arduino có nối với nhau chưa.
- Bước 5: Tiến hành hàn trở và led vào, đo kiểm tra led còn sáng hay không,
- Bước 5: Hàn tất cả các hàng rào, tụ điện.

- Bước 6: Gắn board arduino vào mạch vừa hàn xong. Đo kiểm tra từng chân từ arduino ra port đã kết nối hết chưa.
- Bước 7: Cuối cùng cắm điện vào arduino và nguồn 11.1V trên mạch. Dùng đồng hồ đo áp ở ngõ vào và ngõ ra servo, LCD, cảm biến vân tay, buzzer, led 7 đoạn. Tiếp tục chỉnh biến trở để tinh chỉnh độ tương phản LCD.

4.3. LẬP TRÌNH HỆ THỐNG

4.3.1. Lưu đồ giải thuật

Hệ thống có các chức năng như sau: Cho phép mở cửa sử dụng nhận dạng vân tay bằng cảm biến vân tay cho phép nạp thêm vân tay, xóa bớt số lượng vân tay.

Khi cấp điện vào hệ thống, khởi động arduino, cảm biến vân tay và LCD. Sau khi khởi động xong mặc định sẽ vào chế độ đóng mở cửa, nhấn phím enroll để đổi chế độ đăng kí vân tay, nhấn phim delelect để đổi chế độ xóa bớt số lượng vân tay

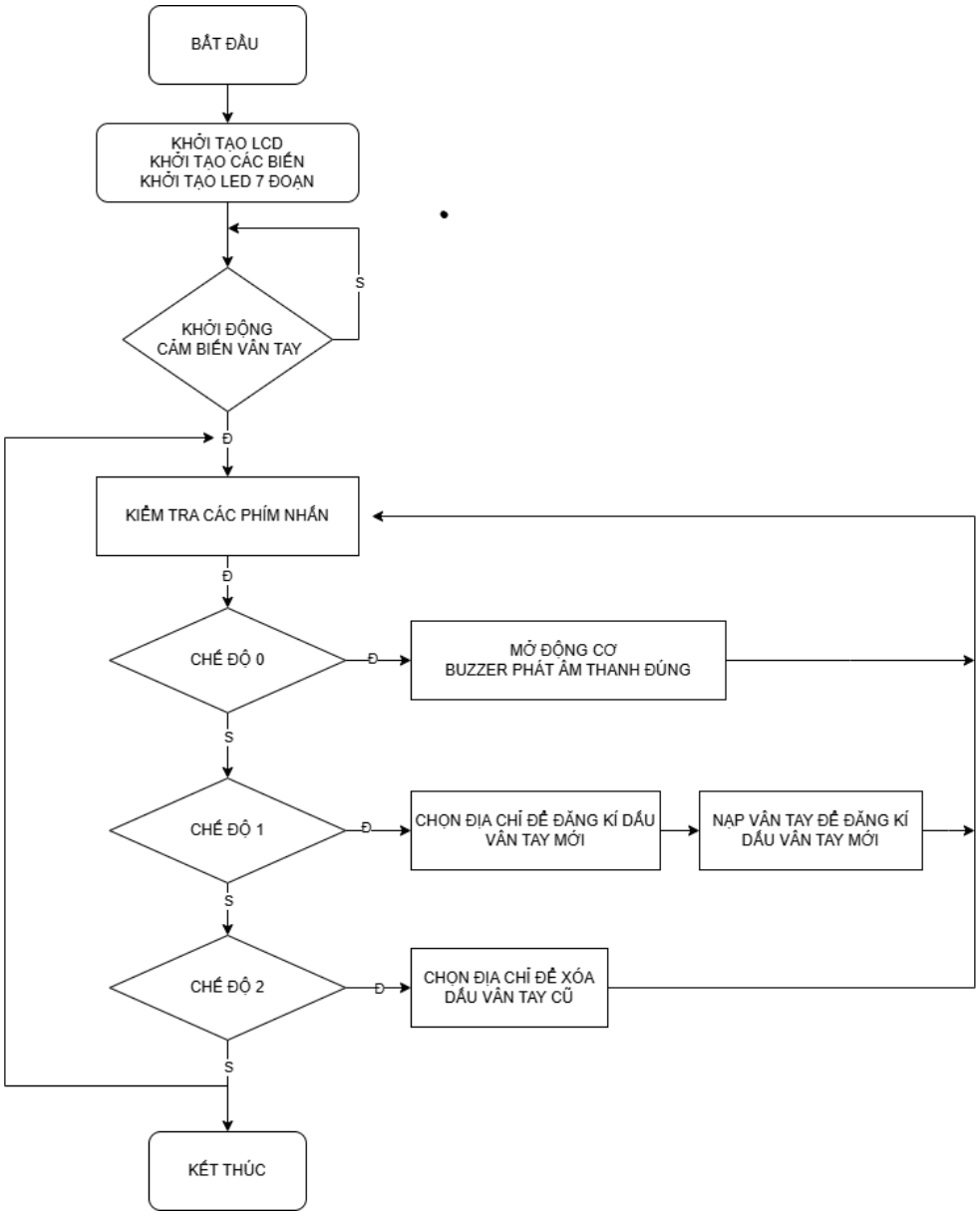
LƯU ĐỒ CHÍNH

Quy trình làm việc của hệ thống như thể hiện trong sơ đồ hình 4.3. Chương trình bắt đầu vào khởi động, LCD, các biến. Sau đó kiểm tra có cảm biến vân tay kết nối chưa. Nếu chưa kết nối thì chương trình sẽ dừng đợi đến khi nào có kết nối. Khi phát hiện có kết nối vân tay thì sẽ khởi động chương trình ngắt 100ms kiểm tra các nút nhấn, để xem chương trình đang hoạt động ở chế độ nào. Mặc định chương sẽ vào chế độ 0. Nếu Nút UP/DW được nhấn giữ 2s thì sẽ vào chế độ 0, nút nhân ENROLL được nhấn thì sẽ vào chế độ 1, nếu phím DEL được nhấn thì sẽ vào chế độ 2..

Chế độ 0: Nhấn giữ Up/Down 2s để bắt đầu kiểm tra, lúc này đặt ngón tay vào cảm biến, nếu vân tay đúng khóa điện từ sẽ tự động mở 5s sau đó đóng; nếu vân tay khôngđúng màn hình sẽ báo Finger Not Found Try Later và khóa chốt sẽ không mở.

Chế độ 1: Đăng kí vân tay: Nhấn nút Enroll để đăng kí vân tay, chọn địa chỉ lưu ID vân tay bằng cách nhấn Up hoặc Down sau đó nhấn Ok. Đặt ngón tay cần đăng kí vào cảm biến cho đến khi màn hình báo Stored!.

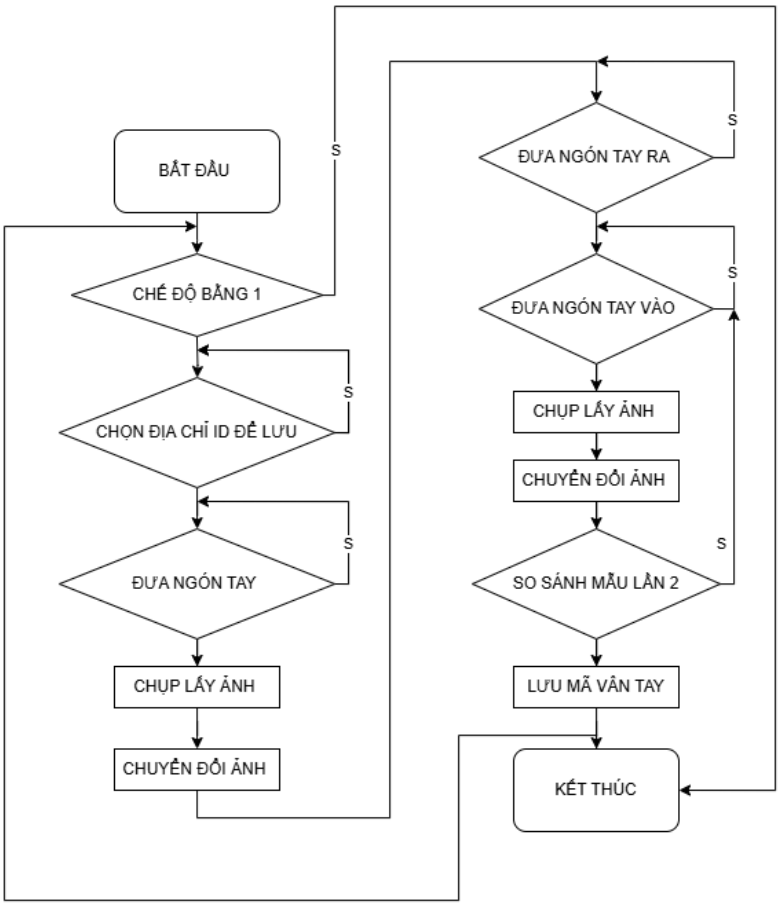
Chế độ 2: Xóa vân tay: Nhấn nút Del để xóa vân tay, chọn địa chỉ ID mà bạn cần xóa bằng cách nhấn UP hoặc Down sau đó nhấn Ok. Màn hình báo Figer Deleted Successfully là thành công.



Hình 4. 3. Lưu đồ chính

CHẾ ĐỘ NẠP VÂN TAY

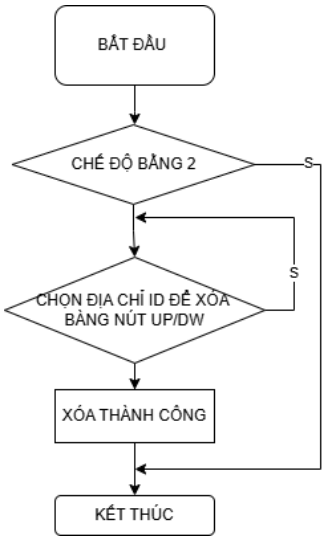
Quy trình lấy mẫu vân tay thêm người dùng thực hiện theo lưu đồ hình 4.4. Nhập ID vị trí lưu dấu vân tay sau đó ta đưa ân tay cần lưu vào cảm biến để tiến hành lấy mẫu lần một và tương tự lấy mẫu lần hai. Nếu hai lần mà không giống nhau thì ta sẽ lấy mẫu lại lần hai cho đến khi giống thì ta lưu dấu vân tay lại. Tiếp tục lưu vân tay với những mẫu khác tương tự như trên cho đến khi kết thúc việc lấy mẫu



Hình 4. 4. Lưu đồ chế độ nạp vân tay

CHẾ ĐỘ XÓA VÂN TAY

Quy trình xóa vân tay người dùng như lưu đồ hình 4.5. Chương trình kiểm tra nếu đang ở chế độ 2. Nhập ID cần xoá bằng 2 nút nhấn điều khiển UP/DW nhấn nút DEL lần nữa để xóa dấu vân tay tại ID đã chọn



Hình 4. 5. Lưu đồ chế độ xóa vân tay

4.3.2. Phần mềm lập trình cho vi điều khiển

- Giới thiệu phần mềm lập trình Arduino IDE

Môi trường phát triển tích hợp Arduino IDE là một ứng dụng đa nền tảng được viết bằng Java, và được dẫn xuất từ IDE cho ngôn ngữ lập trình xử lý và các dự án lắp ráp. Do có tính chất mã nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm. Người sử dụng chỉ cần định nghĩa hai hàm để thực hiện một chương trình hoạt động theo chu trình:

setup(): hàm chạy một lần duy nhất vào lúc bắt đầu của một chương trình dùng để khởi tạo các thiết lập.

loop(): hàm được gọi lặp lại liên tục cho đến khi bo mạch được tắt.

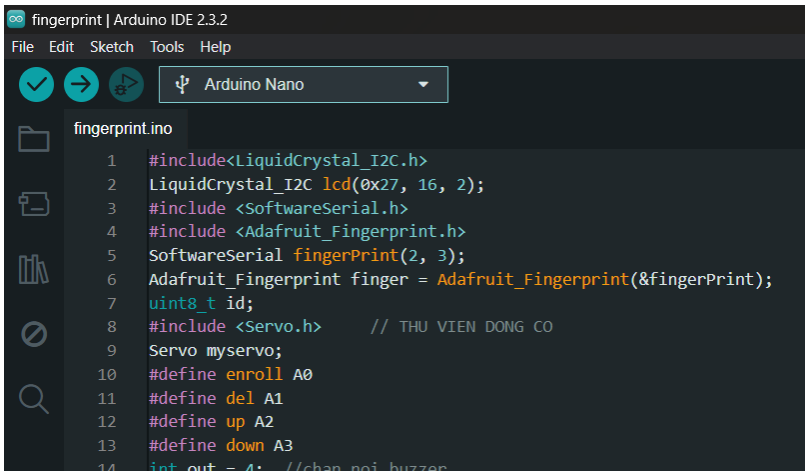
- Giao diện lập trình

1: Menu lệnh: Dùng để thêm thư viện, lưu, tạo project mới,….

2: Nút kiểm tra chương trình (built): Dùng để kiểm tra xem chương trình được viết có lỗi không. Nếu chương trình bị lỗi thì phần mềm Arduino IDE sẽ hiển thị thông tin lỗi ở vùng thông báo thông tin.

3: Nút nạp chương trình xuống board Arduino: Dùng để nạp chương trình được viết xuống mạch Arduino. Trong quá trình nạp, chương trình sẽ được kiểm tra lỗi trước sau đó mới thực hiện nạp xuống mạch Arduino.

4: Vùng lập trình: Vùng này để người lập trình thực hiện việc lập trình cho chương trình của mình.



Hình 4. 6. Giao diện lập trình Arduino

CHƯƠNG 5: KẾT QUẢ THỰC HIỆN

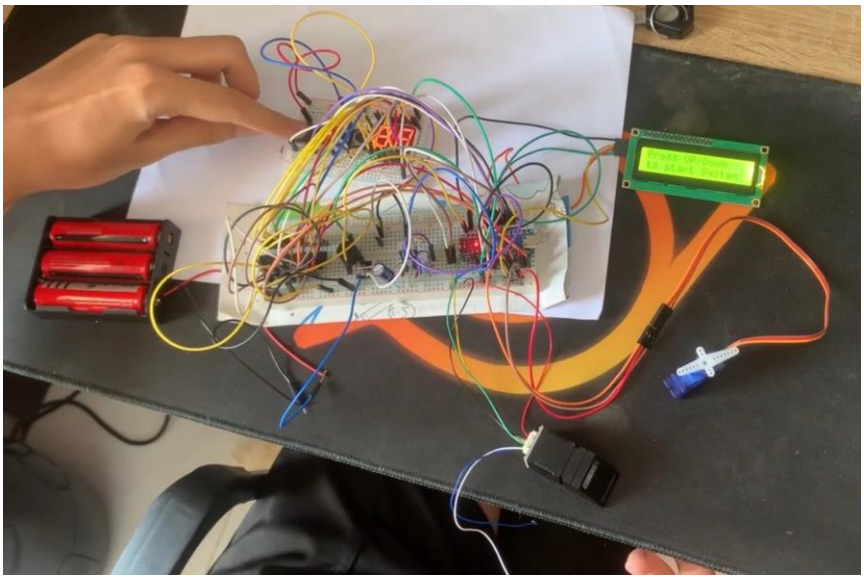
5.1. KẾT QUẢ

Sau quá trình nghiên cứu, thi công thì đồ án "Thiết kế và thi công hệ thống mở cửa bằng cảm biến vân tay" đã hoàn thành và thực hiện được tính năng sau:

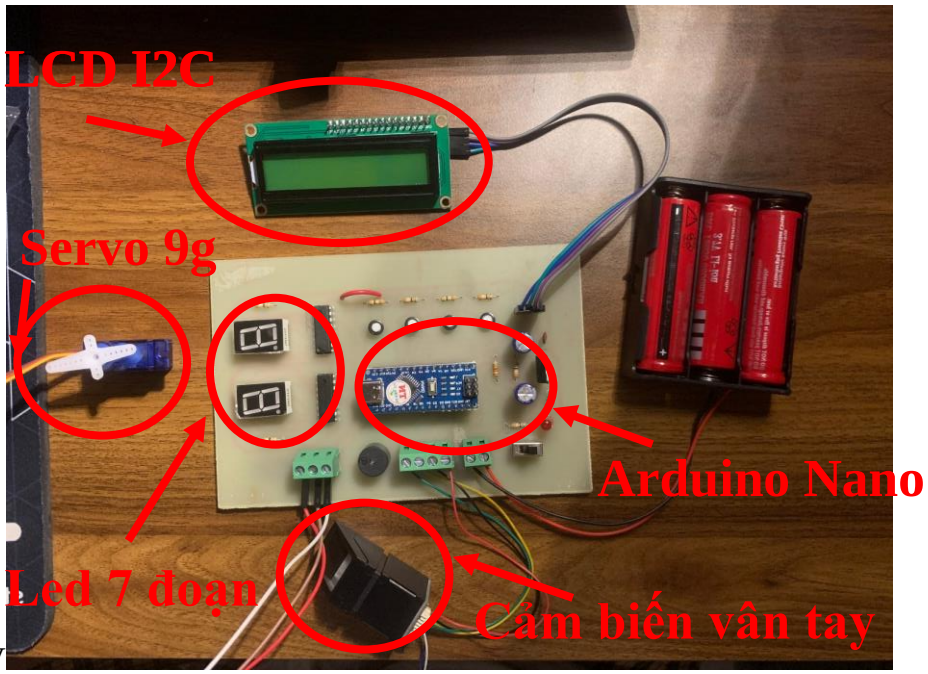
+ Điều khiển đóng mở khóa thông qua nhận dạng dấu vân tay. Khi người dùng muốn khóa hoặc mở cửa, thì tiến hành quét dấu vân tay qua cảm biến vân tay R307. Cảm biến sẽ tiến hành quét dấu vân tay và so sánh với các mẫu vân tay đã được lưu trong bộ nhớ. Nếu là dấu vân tay được chấp nhận thì cho phép điều khiển động cơ Servo để mở chốt cửa và phát âm thanh thành công. Nếu sai thì phát âm thanh thất bại bằng buzzer

+ Hiển thị được trạng thái khóa hoặc mở khóa cửa trên màng hình LCD

+ Có thể kiểm tra phần trăm pin và hiển thị qua led 7 đoạn

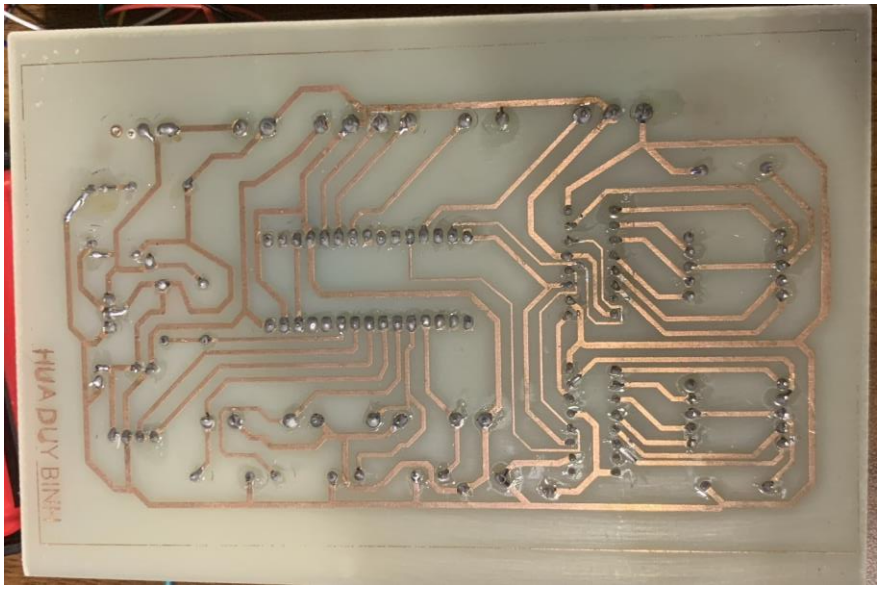


Hình 5. 1. Ảnh test mạch



V

Hình 5. 2. Ảnh kết quả sau khi thực hiện gắn linh kiện



Hình 5. 3. Ảnh mạch mặt sau khi đã hàn xong hoàn chỉnh

Mô hình cửa có các chế độ hoạt động cơ bản như: Nạp thêm vân tay mới vào để cấp thêm quyền truy cập; Xóa dấu vân tay cũ đang được lưu trữ, hiển thị phần trăm pin trên led 7 đoạn (hình 5.4);

Hình 5. 4. Chế độ xóa vân tay, chế độ nạp thêm dấu vân tay

Để mô tả rõ hơn hoạt động của sản phẩm em có đính kèm video clip được đính kèm link sau: *https://youtu.be/x9aV1xu8m3s*

5.2. NHẬN XÉT

Sau thời gian nghiên cứu, thi công thì đồ án tốt nghiệp của em với đề tài "Thiết kế và thi công hệ thống mở cửa bằng cảm biến vân tay" đã cơ bản hoàn tất. Nhìn chung, mô hình đã hoạt động ổn định, đạt được khoảng 95% yêu cầu đề ra ban đầu. Hệ thống cửa dễ sử dụng, an toàn, bảo mật cho người dùng. Sử dụng công nghệ nhận dạng vân tay nên tính bảo mật của mô hình khá cao.

Tuy nhiên, do sự hạn chế về kiến thức và thời gian thực hiện, nguồn tài liệu tham khảo chủ yếu thông qua internet nên đề tài không tránh khỏi sai sót và còn một số hạn chế:

Mô hình hình hoạt động chưa thực sự trơn tru, việc tiến hành quét mã vân tay thực hiện hơi chậm có thể do đáp ứng phần cứng.

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. KẾT LUẬN

Toàn bộ hệ thống chạy tương đối ổn định, đạt kết quả khá tốt. Tuy nhiên đôi lúc vẫn bị ảnh hưởng bởi đáp ứng tác động của hệ thống phần cứng.

Giao tiếp và truyền dữ liệu thành công giữa Arduino Nano với các module cảm biến vân tay R307, buzzer, màng hình LCD, nút nhấn, động cơ Servo SG90 9g.

- Hệ thống nhận, trả lời yêu cầu từ người dùng hoạt động khá ổn định, chính xác.

- Module cảm biến vân tay R307 thực hiện quét nhận diện dấu vân tay hoạt động ổn định, độ chính xác rất cao.

- Điều khiển đóng mở khóa cửa thành công qua nhận dạng dấu vân tay.

- Hiển thị được trạng thái khóa hoặc mở khóa cửa trên màng hình LCD.

- Có thể thay đổi được hoặc thêm vân tay khi cần thiết

Tuy nhiên, do sự hạn chế về kiến thức và thời gian thực hiện, nguồn tài liệu tham khảo chủ yếu thông qua internet nên đề tài không tránh khỏi sai sót và còn một số hạn chế:

- Vì điều kiện kinh phí và thời gian có hạn, nên không thực hiện được mô hình hoàn chỉnh

- Mạch PCB còn khá lớn so với hệ thống đóng mở cửa tự động

- Không có chế độ sleep dẫn đến việc mau hết pin khi sử dụng

6.2. HƯỚNG PHÁT TRIỂN

Tăng cường bảo mật và an toàn: có thể tăng cường thêm yếu tố bảo mật như nhập mật khẩu từ keyboard.

Tối ưu hóa hiệu suất và tiết kiệm năng lượng nguồn điện Pin lâu dài: Nghiên cứu và phát triển các giải pháp tiết kiệm năng lượng để kéo dài thời gian sử dụng pin của khóa cửa.

Có thể thi công mạch điện 2 lớp để thu nhỏ diện tích mạch

Từ đề tài này cũng có thể ứng dụng phát triển lên hệ thống nhà thông minh, tăng hệ thống bảo mật két sắt hoặc bảo mật trên cửa xe hơi…

TÀI LIỆU THAM KHẢO

[1] Lê Văn Thảo, Nguyễn Quang Trí "THIẾT KẾ VÀ THI CÔNG HỆ THỐNG MỞ CỬA BẰNG CẢM BIẾN VÂN TAY" Đồ Án Tốt Nghiệp ĐH, Trường ĐH Sư Phạm Kỹ Thuật TP.HCM, 2016.

[2] "HƯỚNG DẪN SỬ DỤNG CẢM BIẾN VÂN TAY VỚI ARDUINO" https://nshopvn.com/blog/huong-dan-su-dung-cam-bien-van-tay-voi-arduino/

[3] "Arduino Fingerprint Lock", http://www.instructables.com

[4] "Điều khiển Led 7 đoạn", http://arduino.vn/bai-viet/113-dieu-khien-8-den-led-sang-nhap-nhay-theo-y-muon-cua-ban-de-hay-kho

[5] Thư viện cảm biến vân tay - https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library/tree/master

[6] "Arduino | Mạch đo điện áp DC 0 - 50V đơn giản",

https://www.youtube.com/watch?v=P3Fs7DfMBCE

DATASHEET

[7] Cảm biến vân tay,

https://www.rajguruelectronics.com/Product/1276/R307%20Fingerprint%20Module.pdf,

[8] LCD 16x2, https://handsontec.com/dataspecs/module/I2C_1602_LCD.pdf

[9] Động cơ Servo, http://www.micropik.com/PDF/SG90Servo.pdf

[10] IC 74HC595, https://www.ti.com/lit/ds/symlink/sn74hc595.pdf

[11] LED 7 đoạn anode chung, https://docs.rs-online.com/b51e/0900766b801bf827.pdf

[12] Buzzer,

https://components101.com/sites/default/files/component_datasheet/Buzzer%20Datasheet.pdf

[13] IC 7805, https://www.ti.com/lit/ds/symlink/lm7800.pdf

PHỤ LỤC

```
#include<LiquidCrystal_I2C.h> // Thư viện LCD I2C
LiquidCrystal_I2C lcd(0x27, 16, 2); // Khai báo địa chỉ I2C của LCD
#include <SoftwareSerial.h> // Thư viện giao tiếp serial qua các chân số của Arduino
#include <Adafruit_Fingerprint.h> // Thư viện cảm biến vân tay của Adafruit
SoftwareSerial fingerPrint(2, 3); // Khai báo các chân serial cho cảm biến vân tay
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&fingerPrint); // Khai báo cảm biến
vân tay
uint8_t id; // Biến lưu ID của vân tay
#include <Servo.h> // Thư viện điều khiển servo
Servo myservo; // Khai báo servo
#define enroll A0 // Chân A0 dùng để enroll (đăng ký) vân tay
#define del A1 // Chân A1 dùng để xóa vân tay
#define up A2 // Chân A2 dùng để tăng giá trị
#define down A3 // Chân A3 dùng để giảm giá trị
int out = 4; // Chân 4 nối với buzzer
int latchPin = 8; // Chân 8 nối với chân latch của LED 7 đoạn
int clockPin = 12; // Chân 12 nối với chân clock của LED 7 đoạn
int dataPin = 11; // Chân 11 nối với chân data của LED 7 đoạn

int cambien = A6; // Chân A6 đọc điện áp từ cảm biến
int gtcambien; // Biến lưu giá trị từ cảm biến
float vol_out; // Biến lưu điện áp đầu ra từ cảm biến
float vol_in; // Biến lưu điện áp đầu vào tính toán từ điện áp đầu ra

// Mảng chứa các giá trị mã hóa của các số 0-9 cho LED 7 đoạn
const byte Seg[11] = {
  0b11000000, // 0
  0b11111001, // 1
  0b10100100, // 2
  0b10110000, // 3
  0b10011001, // 4
  0b10010010, // 5
  0b10000010, // 6
  0b11111000, // 7
  0b10000000, // 8
  0b10010000, // 9
};

// Hàm hiển thị giá trị lên LED 7 đoạn
void HienThiLED7doan(unsigned long Giatri, byte SoLed = 2) {
  byte *array = new byte[SoLed]; // Tạo mảng lưu các chữ số của giá trị
  for (byte i = 0; i < SoLed; i++) {
    array[i] = (byte)(Giatri % 10UL); // Tách từng chữ số từ phải sang trái
    Giatri = (unsigned long)(Giatri / 10UL); // Chia giá trị cho 10 để lấy chữ số tiếp theo
  }
```

```
  digitalWrite(latchPin, LOW); // Tắt latch để chuẩn bị gửi dữ liệu
  for (int i = SoLed - 1; i >= 0; i--)
    shiftOut(dataPin, clockPin, MSBFIRST, Seg[array[i]]); // Gửi từng byte dữ liệu ra LED
7 đoạn
  digitalWrite(latchPin, HIGH); // Bật latch để hiển thị dữ liệu
  free(array); // Giải phóng bộ nhớ mảng
}

// Hàm setup khởi tạo các thiết bị
void setup() {
  lcd.init(); // Khởi tạo LCD
  lcd.backlight(); // Bật đèn nền LCD
  Serial.begin(9600); // Khởi tạo serial với tốc độ 9600
  myservo.attach(9); // Kết nối servo với chân số 9
  myservo.write(180); // Đặt vị trí ban đầu cho servo (đóng cửa)
  pinMode(enroll, INPUT_PULLUP); // Đặt chân enroll ở chế độ INPUT_PULLUP
  pinMode(up, INPUT_PULLUP); // Đặt chân up ở chế độ INPUT_PULLUP
  pinMode(down, INPUT_PULLUP); // Đặt chân down ở chế độ INPUT_PULLUP
  pinMode(del, INPUT_PULLUP); // Đặt chân del ở chế độ INPUT_PULLUP
  pinMode(latchPin, OUTPUT); // Đặt chân latchPin ở chế độ OUTPUT
  pinMode(clockPin, OUTPUT); // Đặt chân clockPin ở chế độ OUTPUT
  pinMode(dataPin, OUTPUT); // Đặt chân dataPin ở chế độ OUTPUT
  pinMode(out, OUTPUT); // Đặt chân out (buzzer) ở chế độ OUTPUT
  pinMode(cambien, INPUT); // Đặt chân cảm biến ở chế độ INPUT

  lcd.print("HUA DUY BINH"); // Hiển thị tên lên LCD
  lcd.setCursor(0,1);
  lcd.print("21161286"); // Hiển thị mã số lên LCD
  delay(2000);
  finger.begin(57600); // Khởi tạo cảm biến vân tay với tốc độ 57600
  lcd.clear();
  lcd.print("Finding Module"); // Hiển thị thông báo tìm kiếm module vân tay
  lcd.setCursor(0,1);
  delay(1000);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
    lcd.clear();
    lcd.print("Found Module ");
    delay(1000);
  } else {
    lcd.clear();
    lcd.print("Module not Found");
    lcd.setCursor(0,1);
    lcd.print("Check Connections");
    while (1); // Dừng chương trình nếu không tìm thấy module vân tay
  }
```

```
}

// Hàm loop chạy liên tục
void loop() {
  lcd.setCursor(0,0);
  lcd.print("Press UP/Down ");
  lcd.setCursor(0,1);
  lcd.print("to start System");
  battery();
  if(digitalRead(up) == 0 || digitalRead(down) == 0) {
    for(int i = 0; i < 5; i++) {
      lcd.clear();
      lcd.print("Place Finger");
      delay(2000);
      int result = getFingerprintIDez(); // Kiểm tra ID vân tay
      if(result >= 0) {
        winsound(); // Phát âm thanh khi mở cửa thành công
        myservo.write(45); // Mở cửa
        lcd.clear();
        lcd.print("Allowed");
        lcd.setCursor(0,1);
        lcd.print("Gate Opened");
        delay(5000);
        myservo.write(180); // Đóng cửa
        lcd.setCursor(0,1);
        lcd.print("Gate Closed");
        return;
      }
    }
  }
  checkKeys(); // Kiểm tra các phím bấm
  delay(1000);
}

// Hàm đo điện áp pin
void battery() {
  gtcambien = analogRead(cambien); // Đọc giá trị từ cảm biến
  vol_out = (gtcambien * 5) / 1024.0; // Chuyển đổi giá trị từ analog sang điện áp
  vol_in = float(vol_out / float(390.0/float(3100.0 + 390.0))); // Tính toán điện áp nguồn
  int percent = vol_in / 11.1 * 100; // Tính phần trăm pin
  HienThiLED7doan(percent, 2); // Hiển thị phần trăm pin lên LED 7 đoạn
  delay(2000);
}

// Hàm kiểm tra các phím bấm
void checkKeys() {
```

```
  if(digitalRead(enroll) == 0) {
    lcd.clear();
    lcd.print("Please Wait");
    delay(1000);
    while(digitalRead(enroll) == 0);
    Enroll(); // Gọi hàm enroll
  } else if(digitalRead(del) == 0) {
    lcd.clear();
    lcd.print("Please Wait");
    delay(1000);
    delet(); // Gọi hàm delete
  }
}

// Hàm đăng ký vân tay
void Enroll() {
  int count = 0;
  lcd.clear();
  lcd.print("Enroll Finger");
  lcd.setCursor(0,1);
  lcd.print("Location:");
  while(1) {
    lcd.setCursor(9,1);
    lcd.print(count);
    lcd.print("  ");
    HienThiLED7doan(count, 2); // Hiển thị vị trí vân tay lên LED 7 đoạn
    if(digitalRead(up) == 0) { //nếu có nhấn nút UP
      count++; //biến đếm count lên 1 đơn vị
      if(count > 25) count = 0;
      delay(500);
    } else if(digitalRead(down) == 0) {
      count--;
      if(count < 0) count = 25;
      delay(500);
    } else if(digitalRead(del) == 0) {
      id = count;
      getFingerprintEnroll(); // Gọi hàm getFingerprintEnroll để đăng ký vân tay
      return;
    } else if(digitalRead(enroll) == 0) {
      return;
    }
  }
}

// Hàm xóa vân tay
void delet() {
```

```
  int count = 0;
  lcd.clear();
  lcd.print("Delete Finger");
  lcd.setCursor(0,1);
  lcd.print("Location:");
  while(1) {
    lcd.setCursor(9,1);
    lcd.print(count);
    lcd.print("  ");
    HienThiLED7doan(count, 2); // Hiển thị vị trí vân tay lên LED 7 đoạn
    if(digitalRead(up) == 0) {
      count++;
      if(count > 25) count = 0;
      delay(500);
    } else if(digitalRead(down) == 0) {
      count--;
      if(count < 0) count = 25;
      delay(500);
    } else if(digitalRead(del) == 0) {
      id = count;
      deleteFingerprint(id); // Gọi hàm deleteFingerprint để xóa vân tay
      return;
    } else if(digitalRead(enroll) == 0) {
      return;
    }
  }
}

// Hàm lấy ID vân tay để đăng ký
uint8_t getFingerprintEnroll() {
  int p = -1;
  lcd.clear();
  lcd.print("finger ID:");
  lcd.print(id);
  lcd.setCursor(0,1);
  lcd.print("Place Finger");
  delay(2000);
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
      case FINGERPRINT_OK: // Nếu quá trình chuyển đổi thành công
        lcd.clear();
        lcd.print("Image taken"); // hiện thị lên màn hình LCD ảnh đã lấy thành công
        break;
      case FINGERPRINT_NOFINGER: //nếu chưa có ngón tay để lấy ảnh
        lcd.clear();
```

```
      lcd.print("No Finger");
      break;
    case FINGERPRINT_PACKETRECIEVEERR: //nếu có lỗi giao tiếp với cảm biến
      lcd.clear();
      lcd.print("Comm Error");
      break;
    case FINGERPRINT_IMAGEFAIL: //nếu hình ảnh vân tay không hợp lệ
      lcd.clear();
      lcd.print("Imaging Error");
      break;
    default: // các trường hợp lỗi không xác định khác
      lcd.clear();
      lcd.print("Unknown Error");
      break;
   }
  }
p = finger.image2Tz(1);
switch (p) {  // Kiểm tra kết quả của quá trình chuyển đổi hình ảnh vân tay
  case FINGERPRINT_OK:  // Nếu quá trình chuyển đổi thành công
    lcd.clear();  // Xóa nội dung trên màn hình LCD
    lcd.print("Image converted");  // Hiển thị thông lên màn hình LCD
    break;  // Thoát khỏi switch-case

  case FINGERPRINT_IMAGEMESS:  // Nếu hình ảnh vân tay quá lộn xộn để chuyển đổi
    lcd.clear();  // Xóa nội dung trên màn hình LCD
    lcd.print("Image too messy");  // Hiển thị thông  lên màn hình LCD
    return p;  // Trả về giá trị lỗi và kết thúc hàm

  case FINGERPRINT_PACKETRECIEVEERR:  // Nếu có lỗi giao tiếp với cảm biến
    lcd.clear();  // Xóa nội dung trên màn hình LCD
    lcd.print("Comm Error");  // Hiển thị thông báo "Comm Error" lên màn hình LCD
    return p;  // Trả về giá trị lỗi và kết thúc hàm

  case FINGERPRINT_FEATUREFAIL:  // Nếu không tìm thấy đặc trưng vân tay
    lcd.clear();  // Xóa nội dung trên màn hình LCD
    lcd.print("Feature Not Found");  // Hiển thị thông báo lên màn hình LCD
    return p;  // Trả về giá trị lỗi và kết thúc hàm

  case FINGERPRINT_INVALIDIMAGE:  // Nếu hình ảnh vân tay không hợp lệ
    lcd.clear();  // Xóa nội dung trên màn hình LCD
    lcd.print("Feature Not Found");  // Hiển thị thông báo lên màn hình LCD
    return p;  // Trả về giá trị lỗi và kết thúc hàm

  default:  // Các trường hợp lỗi không xác định khác
    lcd.clear();  // Xóa nội dung trên màn hình LCD
    lcd.print("Unknown Error");  // Hiển thị thông báo "Unknown Error" lên màn hình LCD
```

```
    return p;  // Trả về giá trị lỗi và kết thúc hàm
}  lcd.clear();
lcd.print("Remove Finger");
delay(2000);  // Chờ 2 giây để người dùng có thời gian nhấc ngón tay khỏi cảm biến
p = 0;  // Đặt giá trị ban đầu của biến p
while (p != FINGERPRINT_NOFINGER) {  // Lặp lại cho đến khi cảm biến không còn
phát hiện ngón tay nào
  p = finger.getImage();  // Cập nhật giá trị p với kết quả từ hàm getImage()
}
p = -1;  // Đặt lại giá trị của biến p
lcd.clear();
lcd.print("Place Finger");
lcd.setCursor(0,1);
lcd.print("   Again");
while (p != FINGERPRINT_OK) {  // Lặp lại cho đến khi vân tay nhận diện được hình
ảnh vân tay
  p = finger.getImage();  // Cập nhật giá trị p với kết quả từ hàm getImage()
}
p = finger.image2Tz(2);
p = finger.createModel();
p = finger.storeModel(id);  // Lưu mẫu vân tay vào vị trí id trong bộ nhớ của cảm biến
if (p == FINGERPRINT_OK) {  // Kiểm tra xem quá trình lưu trữ có thành công không
  lcd.clear();
  lcd.print("Stored!");
  delay(2000);  // Chờ 2 giây để người dùng có thời gian đọc thông báo
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {  // Nếu có lỗi giao tiếp
  return p;  // Trả về giá trị lỗi và kết thúc hàm
} else if (p == FINGERPRINT_BADLOCATION) {  // Nếu vị trí lưu trữ không hợp lệ
  return p;  // Trả về giá trị lỗi và kết thúc hàm
} else if (p == FINGERPRINT_FLASHERR) {  // Nếu có lỗi khi ghi vào bộ nhớ flash
  return p;  // Trả về giá trị lỗi và kết thúc hàm
} else {  // Các lỗi khác
  return p;  // Trả về giá trị lỗi và kết thúc hàm
}
}
// Hàm lấy ID vân tay
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) // Nếu quá trình chuyển đổi thành công
    return -1; //trả hàm về giá trị âm
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)
    return -1; //trả hàm về giá trị âm
  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK) {
    failsound(); // Phát âm thanh khi không tìm thấy vân tay
```

```
    lcd.clear();
    lcd.print("Finger Not Found");
    lcd.setCursor(0,1);
    lcd.print("Try Later");
    delay(2000);
    return -1;
   }
  return finger.fingerID;
}

// Hàm xóa vân tay theo ID
uint8_t deleteFingerprint(uint8_t id) {
  uint8_t p = -1;
  lcd.clear();
  lcd.print("Please wait");
  p = finger.deleteModel(id);
  if (p == FINGERPRINT_OK) { // Nếu quá trình chuyển đổi thành công
    lcd.clear();
    lcd.print("Finger Deleted");
    lcd.setCursor(0,1);
    lcd.print("Successfully");
    delay(1000);
  } else {
    lcd.clear();
    lcd.print("Something Wrong");
    lcd.setCursor(0,1);
    lcd.print("Try Again Later");
    delay(2000);
    return p;
   }
}

// Hàm phát âm thanh khi thành công
void winsound() {
  tone(out, 3000);
  delay(100);
  noTone(out);
  delay(30);
  tone(out, 4000);
  delay(100);
  noTone(out);
  delay(30);
  tone(out, 5000);
  delay(150);
  noTone(out);
  delay(30);
```

```
}

// Hàm phát âm thanh khi thất bại
void failsound() {
  unsigned char f;
  for (f = 0; f < 3; f++) {
    tone(out, 1000);
    delay(100);
    noTone(out);
    delay(30);
  }
}
```