

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM  
KHOA ĐIỆN - ĐIỆN TỬ



ĐỒ ÁN TỐT NGHIỆP  
NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ - TRUYỀN THÔNG

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

GVHD: ThS. NGUYỄN VIỆT HÙNG  
SVTH: LÊ VĂN THẢO  
MSSV: 12141215  
SVTH: NGUYỄN QUANG TRÍ  
MSSV: 12141236



Tp. Hồ Chí Minh, tháng 07/2016

BỘ GIÁO DỤC & ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN ĐIỆN TỬ-CÔNG NGHIỆP



# ĐỒ ÁN TỐT NGHIỆP

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐỀ TÀI:

THIẾT KẾ VÀ THI CÔNG  
HỆ THỐNG MỞ CỦA BẰNG  
CẨM BIẾN VÂN TAY

GVHD: ThS. Nguyễn Việt Hùng

SVTH 1: Lê Văn Thảo

MSSV: 12141215

SVTH 2: Nguyễn Quang Trí

MSSV: 12141236

Tp. Hồ Chí Minh – 7/2016

BỘ GIÁO DỤC & ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH  
KHOA ĐIỆN – ĐIỆN TỬ  
BỘ MÔN ĐIỆN TỬ-CÔNG NGHIỆP

# ĐỒ ÁN TỐT NGHIỆP

NGÀNH KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐỀ TÀI:

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

GVHD: ThS. Nguyễn Việt Hùng

SVTH 1: Lê Văn Thảo

MSSV: 12141215

SVTH 2: Nguyễn Quang Trí

MSSV: 12141236

Tp. Hồ Chí Minh – 7/2016

# MỤC LỤC

Nhiệm vụ đồ án.....	i
Lịch trình thực hiện đồ án tốt nghiệp.....	iii
Cam đoan .....	iv
Lời cảm ơn .....	v
Mục lục .....	vi
Liệt kê hình vẽ .....	viii
Liệt kê bảng vẽ.....	xii
Tóm tắt .....	xii
<b>CHƯƠNG 1. TỔNG QUAN .....</b>	<b>1</b>
1.1 ĐẶT VÂN ĐÈ .....	1
1.2 MỤC TIÊU .....	2
1.3 NỘI DUNG NGHIÊN CỨU .....	2
1.4 GIỚI HẠN.....	2
1.5. BỐ CỤC .....	3
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....</b>	<b>4</b>
2.1 TỔNG QUAN VỀ ARDUINO .....	4
2.2 TỔNG QUAN VỀ TIN NHẮN SMS .....	5
2.2.1 Sơ lược về SMS .....	5
2.2.2 Cấu trúc một tin nhắn SMS .....	5
2.3 NHẬN DẠNG VÂN TAY .....	6
2.3.1 Giới thiệu sơ lược về dấu vân tay và nhận dạng vân tay .....	6
2.3.2 Việc ứng dụng công nghệ nhận dạng vân tay hiện nay .....	6
2.3.3 Nguyên lý hoạt động cơ bản của nhận dạng vân tay .....	7
2.3.4 Các bước xử lý trong quá trình nhận dạng vân tay.....	7
2.4 GIỚI THIỆU PHẦN CỨNG.....	9
2.4.1 Arduino Mega 2560 .....	9
2.4.2 SIM 900A .....	12
2.4.3 Module cảm biến vân tay R308 (Fingerprint R308).....	19
2.4.4 Màn hình LCD.....	27

2.4.5 Giới thiệu động cơ Servo SG90 9g Micro .....	29
2.4.6 Giới thiệu bàn phím ma trận 4x4 (Keypad 4x4) .....	30
<b>CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ.....</b>	<b>32</b>
3.1 GIỚI THIỆU .....	32
3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG.....	32
3.2.1 Thiết kế sơ đồ khối hệ thống.....	32
3.2.2 Tính toán và thiết kế mạch.....	34
3.2.3 Sơ đồ nguyên lý của toàn mạch .....	43
<b>CHƯƠNG 4. THI CÔNG HỆ THỐNG .....</b>	<b>46</b>
4.1 GIỚI THIỆU .....	46
4.2 THI CÔNG HỆ THỐNG .....	46
4.2.1 Thi công bo mạch .....	46
4.2.2 Lắp ráp và kiểm tra.....	49
4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH .....	49
4.3.1 Đóng gói bộ điều khiển .....	49
4.3.2 Thi công mô hình.....	50
4.4 LẬP TRÌNH HỆ THỐNG .....	51
4.4.1 Lưu đồ giải thuật.....	51
4.4.2 Phần mềm lập trình cho vi điều khiển .....	58
4.5 KẾT QUẢ MÔ PHỎNG.....	62
4.6 VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC .....	63
4.6.1 Viết tài liệu hướng dẫn sử dụng .....	63
4.6.2 Quy trình thao tác .....	64
<b>CHƯƠNG 5. KẾT QUẢ_NHẬN XÉT_ĐÁNH GIÁ .....</b>	<b>65</b>
5.1 KẾT QUẢ.....	65
5.2 NHẬN XÉT – ĐÁNH GIÁ .....	72
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>73</b>
6.1 KẾT LUẬN .....	73
6.2 HƯỚNG PHÁT TRIỂN .....	74
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>75</b>
<b>PHỤ LỤC.....</b>	<b>76</b>

# LIỆT KÊ HÌNH VẼ

Hình	Trang
Hình 2.1 Các loại Board Arduino .....	4
Hình 2.2 Sơ đồ quá trình xử lý ảnh .....	7
Hình 2.3 Sơ đồ quá trình so sánh vân tay .....	8
Hình 2.4 Sơ đồ các chân kết nối trên Arduino Mega 2560.....	10
Hình 2.5 Vị trí chân Arduino Mega .....	11
Hình 2.6 Vị trí chân sim 900A .....	13
Hình 2.7 Giao diện phần mềm Terminal.....	18
Hình 2.8 Mạch chuyển USB UART PL2303.....	18
Hình 2.9 Module cảm biến vân tay R308 .....	20
Hình 2.10 Giao thức truyền thông của R308 .....	23
Hình 2.11 Định nghĩa của thanh ghi .....	24
Hình 2.12 Định dạng gói dữ liệu.....	25
Hình 2.13 Màn hình LCD 16x2 .....	27
Hình 2.14 Động cơ Servo SG90 9g Micro.....	29
Hình 2.15 Điều khiển và kết nối của động cơ Servo SG90 9g Micro .....	30
Hình 2.16 Sơ đồ nối dây và hình ảnh thực tế của bàn phím ma trận 4x4 .....	30
Hình 3.1 Sơ đồ khái hệ thống .....	32
Hình 3.2 Sơ đồ khái hệ thống thiết bị thực tế .....	33
Hình 3.3 Cảm biến vân tay R308 .....	34
Hình 3.4 Sơ đồ nguyên lý cảm biến vân tay kết nối arduino mega .....	35
Hình 3.5 Khối xử lý trung tâm sử dụng board arduino mega 2560 .....	36
Hình 3.6 Động cơ servo 9g .....	36
Hình 3.7 Sơ đồ chân động cơ servo 9g .....	37
Hình 3.8 Điều chỉnh độ rộng xung PPM.....	38
Hình 3.9 Sim 900A thực tế .....	38
Hình 3.10 Sơ đồ chân Sim 900A thực tế.....	39
Hình 3.11 Sơ đồ nguyên lý kết nối sim 900A vào arduino mega .....	40
Hình 3.12 LCD .....	40
Hình 3.13 Sơ đồ nguyên lý kết nối arduino mega với LCD .....	41
Hình 3.14 Sơ đồ nối dây và hình ảnh thực tế của bàn phím ma trận 4x4 .....	42
Hình 3.15 Adapter 9VDC 2A.....	43

Hình 3.16 adapter 5VDC 3A.....	43
Hình 3.17 Kết nối module ngoại vi.....	43
Hình 3.18 LCD và LED báo trạng thái .....	44
Hình 3.19 Khối nguồn.....	44
Hình 3.20 Shield kết nối Arduino mega 2560 .....	45
Hình 3.21 Sơ đồ nguyên lý toàn mạch.....	45
Hình 4.1 Sơ đồ đi dây lớp top .....	46
Hình 4.2 Sơ đồ đi dây lớp bottom .....	46
Hình 4.3 Sơ đồ đi dây lớp bottom và top chưa phủ đồng .....	47
Hình 4.4 Hình dạng 3D lớp top.....	47
Hình 4.5 Hình dạng 3D lớp bottom .....	48
Hình 4.6 Sơ đồ bố trí linh kiện mặt trước mô hình .....	49
Hình 4.7 Hình dạng khung cửa thực tế .....	50
Hình 4.8 Hình dạng phía trước cửa thực tế .....	50
Hình 4.9 Chi tiết cơ khí ráp khung.....	50
Hình 4.10 Lưu đồ chính làm việc của hệ thống .....	52
Hình 4.11 Lưu đồ chế độ nạp vân tay .....	53
Hình 4.12 Lưu đồ chế độ xóa vân tay .....	54
Hình 4.13 Lưu đồ chế độ mở cửa bằng vân tay .....	55
Hình 4.14 Lưu đồ chế độ mở cửa bằng điện thoại .....	56
Hình 4.15 Lưu đồ chương trình ngắn.....	57
Hình 4.16 Quy trình làm việc của arduino .....	58
Hình 4.17 Giao diện lập trình arduino .....	58
Hình 4.18 Giao diện menu arduino IDE .....	59
Hình 4.19 Giao diện file menu arduino IDE .....	59
Hình 4.20 Giao diện Examples menu arduino IDE .....	59
Hình 4.21 Giao diện Sketch menu arduino IDE .....	60
Hình 4.22 Giao diện edit menu arduino IDE .....	60
Hình 4.23 Giao diện tool menu arduino IDE .....	61
Hình 4.24 Board Arduino sử dụng .....	61
Hình 4.25 Hiển thị Board và Serial Port đã kết nối .....	62
Hình 4.26 Arduino Toolbar.....	62
Hình 4.27 Xung PPM mở cửa.....	62
Hình 4.28 Xung PPM đóng cửa.....	63

Hình 4.29 Quy trình thao tác .....	64
Hình 5.1 Chế độ đóng mở bằng vân tay.....	66
Hình 5.2 Chế độ đóng mở bằng cách gọi điện thoại .....	67
Hình 5.3 Kiểm tra trạng thái cửa.....	67
Hình 5.4 Mặt trên của mạch in.....	68
Hình 5.5 Mặt dưới của mạch in.....	68
Hình 5.6 Sơ đồ mạch in.....	68
Hình 5.7 Hình thực tế gắn linh kiện cho mặt trên .....	69
Hình 5.8 Hình ảnh thực tế gắn linh kiện cho mặt dưới .....	69
Hình 5.9 Hình ảnh hệ thống sau khi kết nối.....	70
Hình 5.10 Mặt bên trong mô hình cửa .....	70
Hình 5.11 Mặt ngoài mô hình cửa .....	71
Hình 5.12 Nạp vân tay .....	71
Hình 5.13 Xóa vân tay .....	71
Hình 5.14 Đóng mở cửa .....	71

# **LIỆT KÊ BẢNG**

<b>Bảng</b>	<b>Trang</b>
Bảng 2.1: Các tính năng chính của SIM 900A .....	14
Bảng 2.2: Một số lệnh AT cơ bản .....	15
Bảng 2.3: Các chân kết nối của module R308 .....	22
Bảng 2.4: Định dạng gói dữ liệu .....	26
Bảng 2.5: Các chân của LCD .....	28
Bảng 3.1: Chức năng chân cảm biến vân tay R308 .....	34
Bảng 3.2: Thông số kỹ thuật arduino mega 2560 .....	36
Bảng 3.3: Kết nối chân LCD với arduino .....	41
Bảng 4.1. Danh sách các linh kiện. ....	48

## TÓM TẮT

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

Với mục đích muốn tiếp cận với các công nghệ đang phát triển trên, nên nhóm thực hiện đồ án với mong muốn chế tạo ra mô hình hệ thống cửa được đóng mở bằng cách tiến hành quét vân tay hoặc sử dụng chức năng nhắn tin và gọi điện để điều khiển.

Mô hình sử dụng kit Arduino Mega 2560 làm vi điều khiển trung tâm để điều khiển các module mở rộng như cảm biến vân tay R308, Sim 900A...

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, sử dụng tin nhắn sms và gọi điện, có hướng dẫn cách sử dụng rõ ràng.

# Chương 1. TỔNG QUAN

## 1.1 ĐẶT VĂN ĐỀ

Trong thời đại ngày nay, sự phát triển không ngừng của khoa học kỹ thuật đã giúp cho con người thuận tiện hơn trong các công việc hàng ngày, các thiết bị và đồ dùng trong nhà ngày càng được số hóa, tự động và thông minh hơn. Nền kinh tế cũng phát triển nhanh chóng, kéo theo chất lượng cuộc sống được cải thiện rất nhiều, số lượng người giàu ngày càng tăng. Nhiều gia đình bắt đầu có điều kiện hơn và có nhu cầu mua các thiết bị công nghệ hiện đại, đồ dùng cao cấp, đắt tiền hiện nay khá cao.

Tuy nhiên, tỷ lệ tội phạm trộm cắp vẫn còn rất cao, chúng thường hay nhầm vào các tài sản đắt tiền và các gia đình giàu có. Để bảo vệ an toàn cho tài sản trong nhà là vấn đề khá đau đầu của nhiều gia đình. Bể khóa được các cánh cửa sử dụng chìa khóa và ổ khóa thông thường hiện nay trở nên dễ dàng hơn rất nhiều cho bọn trộm vì bọn chúng ngày càng chuyên nghiệp, tinh vi. Do hệ thống cửa chính là nơi dễ đột nhập vào nhà nhất để trộm cắp tài sản. Nhận thức được tính cấp thiết đó, nhóm chúng em đã quyết định chọn đề tài “THIẾT KẾ VÀ THI CÔNG HỆ THỐNG MỞ CỦA BẦNG CẨM BIẾN VÂN TAY” nhằm tăng tính năng bảo mật và bảo vệ an toàn tài sản cho người sử dụng.

Công nghệ sinh trắc [1] ra đời đáp ứng được các yêu cầu về sự bảo mật riêng tư thông tin cá nhân cũng như để nhận biết một người nào đó trong hàng tỉ người trên thế giới. Các đặc trưng sinh trắc thường được sử dụng là vân tay, gương mặt, mống mắt, tiếng nói. Trong đó, vân tay đã được biết tới với tính phân biệt (tính chất cá nhân) và ổn định theo thời gian cao nhất, vì vậy nó là đặc trưng sinh trắc được sử dụng rộng rãi nhất, được xem là một trong những kỹ thuật nhận dạng hoàn thiện và đáng tin cậy nhất. Song song đó, mạng di động ngày càng phát triển rộng rãi, gắn liền với công việc và cuộc sống hàng ngày của con người. Mặt khác, việc phát triển không ngừng của vi xử lý đã cho ra đời nhiều loại sản phẩm thông minh nhỏ gọn, các Kit nhúng xuất hiện với nhiệm vụ hỗ trợ học tập, nghiên cứu ngày càng nhiều. Trong đó kit Arduino là một sản phẩm mang tính ứng dụng cao với một cộng đồng phát triển mạnh mẽ trên quy mô

## **CHƯƠNG 1. TỔNG QUAN**

---

toàn cầu. Và ta có thể kết hợp giữa mạng di động và Kit Arduino để điều khiển các thiết bị bằng cách nhắn tin SMS hoặc gọi điện thoại...Dựa vào các công nghệ trên, mô hình Cửa thông minh của nhóm sử dụng Adruino và module cảm biến vân tay [9], module Sim 900A [12] làm hệ thống điều khiển đóng mở cửa.

### **1.2 MỤC TIÊU**

Tìm hiểu và nghiên cứu về kit Arduino, module cảm biến vân tay R308, module SIM 900A 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 qua cảm biến vân tay.

Xây dựng hệ thống nhận tin nhắn và cuộc gọi để điều khiển đóng mở cửa qua tin nhắn SMS và cuộc gọi.

### **1.3 NỘI DUNG NGHIÊN CỨU**

- NỘI DUNG 1: Tìm hiểu và nghiên cứu về cấu tạo phần cứng, nguyên lý hoạt động, tính năng của các module Arduino, Sim 900A, R308, động cơ Servo, ma trận phím....
- NỘI DUNG 2: Các giải pháp thiết kế hệ thống, thi công mô hình cửa thông minh.
- NỘI DUNG 3: Thiết kế hệ thống điều khiển, lưu đồ giải thuật và chương trình điều khiển mô hình.
- NỘI DUNG 4: Thiết kế hoàn chỉnh mô hình thực tế.
- NỘI DUNG 5: Chạy thử nghiệm hệ thống cửa thông minh.
- NỘI DUNG 6: Cân chỉnh hệ thống.
- NỘI DUNG 7: Viết sách luận văn.
- NỘI DUNG 8: Bảo vệ đề tài tốt nghiệp

### **1.4 GIỚI HẠN**

- Thiết kế mô hình cửa có kích thước dài, rộng, cao là 30 x 30 x 40 cm.
- Cảm biến vân tay R308 có thể quét và lưu trữ hơn 900 mẫu vân tay khác nhau.
- Module Sim 900A có thể dùng để gọi điện và nhắn tin.
- Nguồn cấp cho cảm biến vân tay và module Sim 900A sử dụng adapter 5VDC-3A

- Hệ thống chỉ sử dụng được khi được cấp điện, không có nguồn điện dự phòng.
- Đề tài chỉ xây dựng mô hình cửa có kích thước nhỏ bằng Mica.

### 1.5. BỘ CỤC

- **Chương 1: Tổng Quan**

Chương này trình bày về đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bộ cục đồ án.

- **Chương 2: Cơ 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 khái niệm, sơ đồ nguyên lý toàn mạch, tính toán thiết kế mạch.

- **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**

Trình bày về những kết quả đã được mục tiêu đề ra sau quá trình nghiên cứu thi công. Từ những kết quả đạt được để đánh giá quá trình hoàn thành được bao nhiêu phần trăm.

- **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 TỔNG QUAN VỀ ARDUINO**



**Hình 2.1.** Các loại Board Arduino

➤ **Giới thiệu:**

Arduino [2] là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Đi kèm với nó là một môi trường phát triển tích hợp (IDE) chạy trên các máy tính cá nhân thông thường và cho phép người dùng viết các chương trình cho Arduino bằng ngôn ngữ C hoặc C++. Trong hình 2.1 bên trên là các loại board Arduino khác nhau có trên thị trường.

➤ **Phần cứng:**

Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8-bit, hoặc ARM Atmel 32-bit. Board Arduino sẽ đưa ra hầu hết các chân I/O của vi điều khiển để sử dụng cho những mạch ngoài.

Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau giúp dễ

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

dàng lập trình và có thể mở rộng với các mạch khác, các module thêm vào có thể dễ dàng chuyển đổi, được gọi là shield. Vài shield kết nối với board Arduino trực tiếp thông qua các chân khác nhau, nhưng nhiều shield được định địa chỉ thông qua serial bus I2C, nhiều shield có thể được xếp chồng và sử dụng dưới dạng song song. Arduino chính thức thường sử dụng các dòng chip MegaAVR, đặc biệt là ATMega8, ATMega168, ATMega328, ATMega1280, và ATMega2560.

Theo nguyên tắc, khi sử dụng phần mềm Arduino, tất cả các board được lập trình thông qua một kết nối RS-232, nhưng cách thức thực hiện lại tùy thuộc vào đời phần cứng. Các board Serial Arduino có chứa một mạch chuyển đổi giữa RS-232 sang TTL. Các board Arduino hiện tại được lập trình thông qua cổng USB, thực hiện thông qua chip chuyển đổi USB-to-serial như là FTDI FT232.

### 2.2 TỔNG QUAN VỀ TIN NHẮN SMS

#### 2.2.1 Sơ lược về SMS

SMS là từ viết tắt của Short Message Service có thể thấy được là dữ liệu có thể được lưu giữ bởi một tin nhắn SMS là rất giới hạn. Một tin nhắn SMS có thể chứa tối đa là 140 byte (1120 bit) dữ liệu. Vì vậy, một tin nhắn SMS chỉ có thể chứa được:

- Gồm 160 ký tự nếu như mã hóa ký tự 7 bit được sử dụng phù hợp với chữ latin.
- Gồm 70 ký tự nếu như mã hóa ký tự 16 bit Unicode UCS2 được sử dụng.

Tin nhắn SMS dạng text hỗ trợ nhiều ngôn ngữ khác nhau. Nó có thể hoạt động tốt với nhiều ngôn ngữ mà có hỗ trợ mã Unicode, bao gồm cả Arabic, Trung Quốc, Nhật bản và Hàn Quốc.

Các tin nhắn SMS có thể được gửi và đọc tại bất kỳ thời điểm nào.

Tin nhắn SMS có thể được gửi tới các điện thoại khi tắt nguồn.

Các tin nhắn SMS ít gây phiền phức trong khi ta vẫn có thể liên lạc với được người khác.

#### 2.2.2 Cấu trúc một tin nhắn SMS

Nội dung của một tin nhắn SMS khi được gửi đi sẽ được chia làm 5 phần như sau:

- **Instructions to air interface:** chỉ thị dữ liệu kết nối với air interface (giao diện không khí) .

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

- **Instructions to SMSC:** chỉ thị dữ liệu kết nối với trung tâm tin nhắn SMSC (short message service centre).
- **Instructions to handset:** chỉ thị dữ liệu kết nối bắt tay.
- **Instructions to SIM (optional):** chỉ thị dữ liệu kết nối, nhận biết SIM (Subscriber Identity Modules).
- **Message body:** nội dung tin nhắn SMS.

### 2.3 NHẬN DẠNG VÂN TAY

#### 2.3.1 Giới thiệu sơ lược về dấu vân tay và nhận dạng vân tay

##### a. Khái niệm về dấu vân tay

Vân tay [1] là do các gai da đội lớp biểu bì lên mà thành. Đó là nơi tập kết miệng các tuyến mồ hôi, tuyến bã nhòn... Nó đã định hình khi con người còn là cái thai 4 tháng trong bụng mẹ. Khi đứa bé ra đời, lớn lên, vân tay được phóng đại nhưng vẫn giữ nguyên dạng cho đến khi về già. Nếu tay có bị bỏng, bị thương, bị bệnh thì khi lành, vân tay lại tái lập y hệt như cũ. Chỉ khi có tổn thương sâu huỷ hoại hoàn toàn, sẹo chằng chít mới xoá mất vân tay.

Vân tay không ai giống ai, đặc sắc nhất là vân ngón cái và ngón trỏ.

##### b. Giới thiệu về nhận dạng vân tay

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 lý 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.

#### 2.3.2 Việc ứng dụng công nghệ nhận dạng vân tay hiện nay

Trên thế giới hiện nay đã xuất hiện nhiều sản phẩm công nghệ cao sử dụng phương pháp nhận dạng vân tay như khóa vân tay, máy chấm công vân tay, máy tính xách tay, điện thoại thông minh... Tuy nhiên đây vẫn là vấn đề còn chưa được nghiên cứu nhiều ở Việt Nam. Ở nước ta, phương pháp này mới chỉ phổ biến ở việc quản lý nhân sự thông qua chứng minh thư nhân dân và phục vụ điều tra phá án. Các sản phẩm công nghệ cao

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

nói trên chúng ta vẫn phải nhập khẩu với giá thành khá cao, do đó chúng vẫn chưa được phổ biến rộng rãi.

### 2.3.3 Nguyên lý hoạt động cơ bản của nhận dạng vân tay

Nguyên lý hoạt động của công nghệ nhận dạng vân tay là khi đặt ngón tay lên trên một thiết bị nhận dạng dấu vân tay, ngay lập tức thiết bị này sẽ quét hình ảnh ngón tay đó và đối chiếu các đặc điểm của ngón tay đó với dữ liệu đã được lưu trữ trong hệ thống. Quá trình xử lý dữ liệu sẽ được thiết bị chuyển sang các dữ liệu số và ra thông báo rằng dấu vân tay đó là hợp lệ hay không hợp lệ để cho phép hệ thống thực hiện các chức năng tiếp theo. Hệ thống sinh trắc học sẽ ghi nhận mẫu vân tay của người dùng và lưu trữ tất cả những dữ liệu đặc biệt này thành một mẫu nhận diện được số hóa toàn phần. Có hai phương pháp để lấy dấu vân tay.

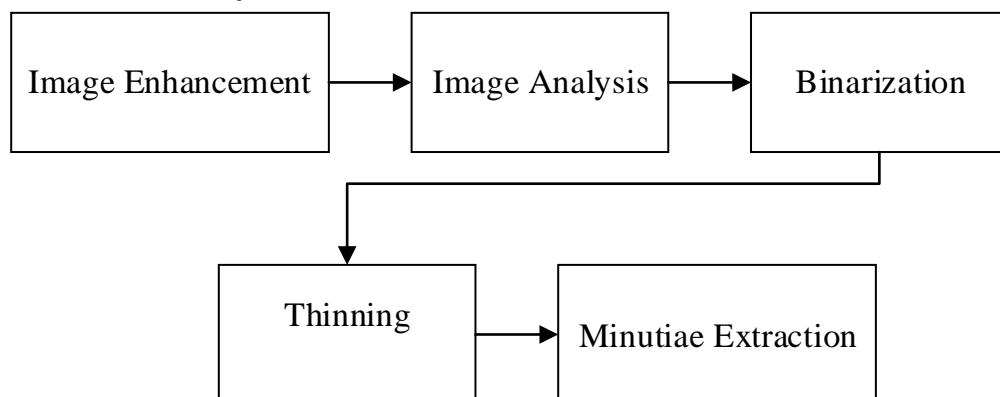
Cách thứ nhất (cô điểm) là sao chép lại hình dạng vân tay (như lăn tay bằng mực, hay chạm vào một vật gì đó) thông qua máy quét ghi nhận và xử lý.

Cách thứ hai, hiện tại đa số các nước đều sử dụng phần mềm hoặc thiết bị quét vân tay để nhận dạng vân tay.

### 2.3.4 Các bước xử lý trong quá trình nhận dạng vân tay

Quá trình xử lý nhận dạng vân tay được chia làm hai quá trình lớn: quá trình xử lý ảnh và quá trình so sánh vân tay.

#### a. Quá trình xử lý ảnh



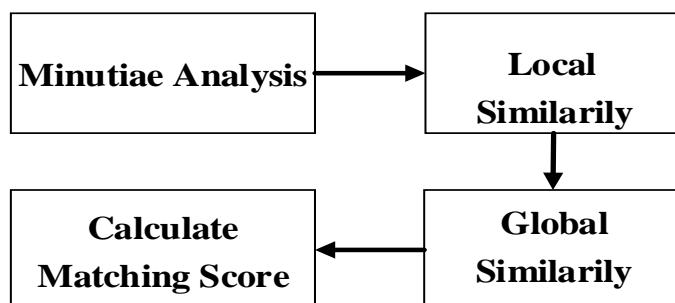
**Hình 2.2.** Sơ đồ quá trình xử lý ảnh

Mục đích của quá trình này được biểu diễn qua hình 2.2 là tăng cường ảnh vân tay, sau đó, rút trích các đặc trưng vân tay từ ảnh đã được tăng cường. Quá trình này được thực hiện qua các bước nhỏ sau:

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- **Tăng cường ảnh (Image Enhancement):** Ảnh được lấy từ thiết bị đầu đọc vân tay sẽ được làm rõ. Do các thiết bị đầu đọc vân tay không lấy ảnh tốt hay do vân tay của người dùng trong lúc lấy bị hao mòn, dơ bẩn, hay do lực ấn ngón tay trong lúc lấy vân tay. Vì vậy, bước này là một trong các bước quan trọng nhất của quá trình này để làm rõ ảnh vân tay để rút trích các đặc trưng đúng và đầy đủ.
- **Phân tích ảnh (Image Analysis):** Thông qua phân tích ảnh, ảnh sẽ được loại bỏ những thông tin làm nhiễu hay những thông tin không cần thiết.
- **Nhị phân hóa (Binarization):** Nhị phân hóa ảnh vân tay thành ảnh trắng đen. Bước này phục vụ cho bước Làm mỏng vân tay. Bước này có thể có hoặc không vì phục thuộc vào thuật toán rút trích đặc trưng.
- **Làm mỏng (Thinning):** Làm mỏng các đường vân lồi của ảnh vân tay. Bước này nhằm mục đích cho việc rút trích đặc trưng của vân tay. Bước này cũng có thể có hoặc không vì phục thuộc vào thuật toán rút trích đặc trưng.
- **Rút trích đặc trưng (Minutiae Extraction):** Rút trích những đặc trưng cần thiết cho quá trình so sánh vân tay.

### b. Quá trình so sánh vân tay



**Hình 2.3.** Sơ đồ quá trình so sánh vân tay

Mục đích của quá trình này được biểu diễn trên hình 2.3 là so sánh vân tay dựa trên các đặc trưng đã được rút trích. Quá trình này được thực hiện qua các bước nhỏ sau:

- **Phân tích đặc trưng (Minutiae Analysis):** Phân tích các đặc điểm cần thiết của các đặc trưng để phục vụ cho việc so sánh vân tay.
- **Xét độ tương tự cục bộ (Local Similarity):** Thuật toán so sánh vân tay sẽ dựa vào các thông tin cục bộ của các đặc trưng (gồm: tọa độ (x, y), hướng của đặc trưng, góc

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

tạo bởi tiếp tuyến của đường vân tại đặc trưng và trực ngang) của vân tay để tìm ra các cặp đặc trưng giống nhau giữa hai vân tay.

- **Xét độ tương tự toàn cục (Global Similarity):** Từ những khu vực tương tự nhau trên cục bộ, thuật toán sẽ tiếp tục mở rộng so sánh trên toàn cục.
- **Tính điểm so sánh (Calculate Matching Score):** Tính toán tỷ lệ độ giống nhau giữa các cặp đặc trưng. Điểm so sánh này sẽ cho biết độ giống nhau của hai ảnh vân tay là bao nhiêu.

### 2.4 GIỚI THIỆU PHẦN CỨNG

- **Thiết bị đầu vào:** bàn phím ma trận 4x4, module cảm biến vân tay R308, module Sim 900A.
- **Thiết bị đầu ra:** led đơn, màn hình LCD 16x2, động cơ Servo SG90.
- **Thiết bị điều khiển trung tâm:** Board Arduino Mega 2560.
- **Các chuẩn truyền dữ liệu UART.**

#### 2.4.1 Arduino Mega 2560

Arduino Mega 2560 sử dụng chip ATmega2560. Nó có 54 chân digital I/O (trong đó có 15 chân điều chế độ rộng xung PWM), 16 chân đầu vào tương tự (Analog Inputs), 4 USARTs (cổng nối tiếp phần cứng), một thạch anh dao động 16 MHz, kết nối USB, một jack cắm điện, một đầu ICSP và một nút reset. Nó chứa tất cả mọi thứ cần thiết để hỗ trợ các vi điều khiển, chỉ đơn giản là kết nối nó với một máy tính bằng cáp USB hoặc với một bộ chuyển đổi điện AC-DC hoặc có thể sử dụng pin.

Board có khả năng tự động reset nhờ phần mềm thay vì đòi hỏi phải ấn nút reset trước khi tải lên. Phần mềm Arduino sử dụng khả năng này để cho phép nạp code lên chỉ cần nhấn vào nút Upload trong Arduino IDE. Điều này có nghĩa rằng bộ nạp khởi động có thể có một thời gian chờ ngắn hơn. Arduino Mega 2560 có thẻ bảo vệ cổng USB của máy tính khi xảy ra hiện tượng quá dòng. Mặc dù hầu hết các máy tính cung cấp bảo vệ nội bộ, các cầu chì cung cấp thêm một lớp bảo vệ. Nếu dòng cao hơn 500mA được áp dụng cho các cổng USB, cầu chì sẽ tự động phá vỡ các kết nối cho đèn khi ngắt hoặc hiện tượng quá tải được khắc phục.

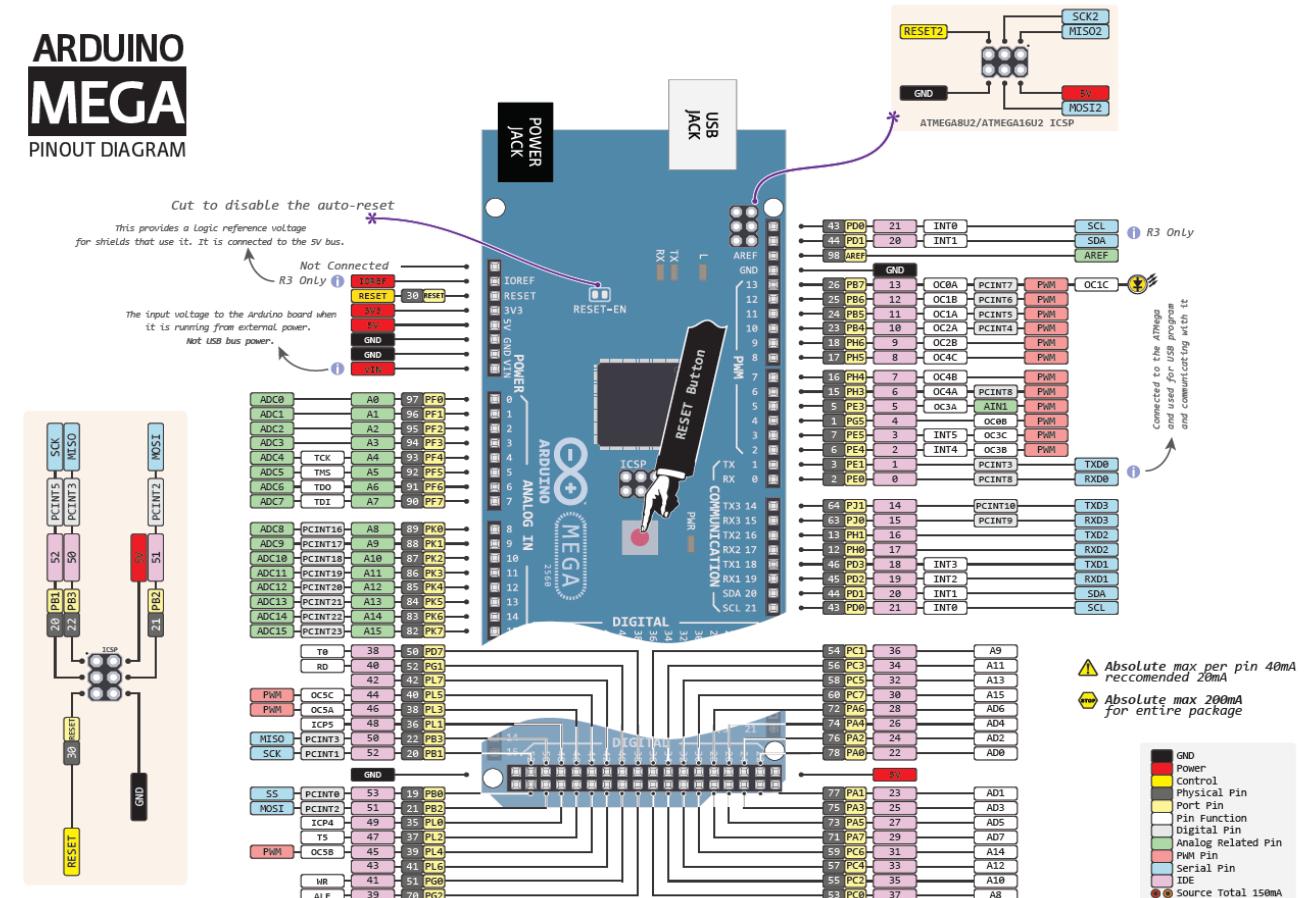
- a. **Thông số kỹ thuật Arduino Mega 2560**
  - Chip vi điều khiển: ATmega2560.

## **CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

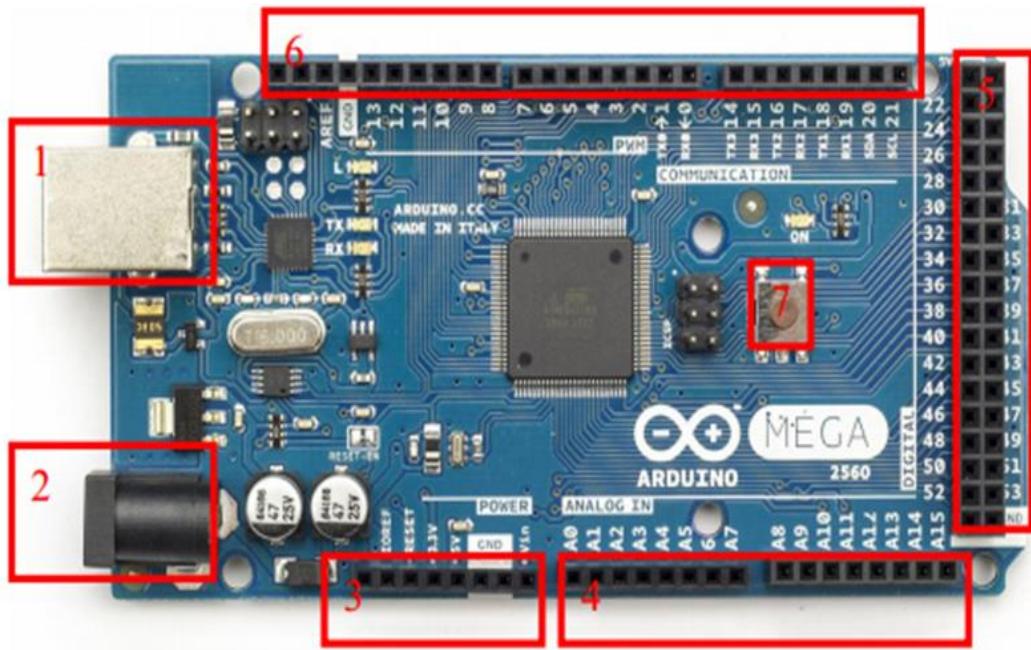
- Điện áp cấp nguồn: 5V.
  - Điện áp đầu vào (kiến nghị): 7-12V.
  - Điện áp đầu vào (giới hạn): 6-20V.
  - Số chân Digital I/O: 54 (có 15 chân điều chế độ rộng xung PWM).
  - Số chân Analog (Input ): 16.
  - Dòng DC trên chân I/O: 40 mA.
  - Dòng DC cho chân 3.3V: 50 mA.
  - Flash Memory: 256KB trong đó có 8KB được sử dụng bởi bộ nạp khởi động (bootloader).
  - SRAM: 8 KB.
  - EEPROM: 4 KB.
  - Xung nhịp: 16 MH.
  - Chiều dài: 101,52 mm.
  - Chiều rộng: 53,3 mm.
  - Cân nặng: 37 g.

b. Sơ đồ kết nối chân của Arduino Mega 2560

Trong hình 2.4 bên dưới là hình ảnh sơ đồ các chân kết nối trên Arduino Mega 2560 cùng với hình 2.5 là vị trí các chân của Arduino Mega 2560.



Hình 2.4. Sơ đồ các chân kết nối trên Arduino Mega 2560



**Hình 2.5.** Vị trí chân Arduino Mega

➤ **USB (1):**

Arduino sử dụng cáp USB để giao tiếp với máy tính. Thông qua cáp USB chúng ta có thể Upload chương trình cho Arduino hoạt động, ngoài ra USB còn là nguồn cho Arduino.

➤ **Nguồn (2 và 3):**

Khi không sử dụng USB làm nguồn thì chúng ta có thể sử dụng nguồn ngoài thông qua jack cắm 2.1mm (cực dương ở giữa) hoặc có thể sử dụng 2 chân Vin và GND để cấp nguồn cho Arduino. Bo mạch hoạt động với nguồn ngoài ở điện áp từ 5 – 20 volt. Chúng ta có thể cấp một áp lớn hơn tuy nhiên chân 5V sẽ có mức điện áp lớn hơn 5 volt. Và nếu sử dụng nguồn lớn hơn 12 Volt thì sẽ có hiện tượng nóng và làm hỏng bo mạch. Khuyến cáo nên dùng nguồn ổn định từ 6 đến dưới 12 volt. Chân 5V và chân 3.3V (Output voltage): các chân này dùng để lấy nguồn ra từ nguồn mà chúng ta đã cung cấp cho Arduino để cấp cho các thiết bị giao tiếp khác. Lưu ý: không được cấp nguồn vào các chân này vì sẽ làm hỏng Arduino.

➤ **Ngõ vào tương tự (4):**

Arduino Mega 2560 có 16 ngõ vào tương tự (các chân từ A0 đến A15), mỗi ngõ vào này đều có độ phân giải 10 bit (1024 giá trị). Mặc định đo từ 0 đến 5V, có thể thiết

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

ập giá trị bằng cách điều chỉnh chân AREF và sử dụng hàm AnalogReference() để chuyển đổi.

### ➤ Ngõ vào số (5 và 6):

Mỗi một chân trong 54 chân số của board đều có thể sử dụng như một ngõ vào hoặc ngõ ra. Mỗi chân có thể cung cấp hoặc nhận được tối đa 40 mA và có một điện trở kéo lên bên trong 20-50 kOhms. Ngoài ra, một số chân có chức năng đặc biệt:

- **Serial:** để truyền và nhận dữ liệu nối tiếp. Gồm các chân: Serial 0: 0 (RX) và 1 (TX); Serial 1: 19 (RX) và 18 (TX); Serial 2: 17 (RX) và 16 (TX); Serial 3: 15 (RX) và 14 (TX).

- **External Interrupt:** Các chân này có thể được cấu hình để kích hoạt sự kiện ngắt mức thấp, ngắt cạnh lên hoặc xuống. Gồm các chân: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), 21 (interrupt 2).

- **PWM:** Cung cấp ngõ ra PWM 8 bit. Gồm các chân từ chân 2 đến 13 và 44 đến 46.

- **SPI:** Các chân này hỗ trợ giao tiếp SPI bằng cách sử dụng thư viện SPI. Có các chân: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).

- **TWI:** Hỗ trợ giao tiếp TWI bằng việc sử dụng thư viện WIRE. Có các chân: 20 (SDA) và 21 (SCL).

### ➤ Nút Reset (7):

Mang dòng mức thấp để thiết lập lại vi điều khiển.

### 2.4.2 SIM 900A

Việc truyền dữ liệu hay giao tiếp giữa các thiết bị với nhau đã trở nên quá quen thuộc. Nhưng đôi khi việc truyền dữ liệu giữa các thiết bị với nhau lại bị gián đoạn hoặc không thể thực hiện được vì một số lỗi nào đó. Để giải quyết vấn đề này người dùng thường sử dụng các phương tiện như: Bluetooth, Wifi,... đó là phương thức đơn giản để giải quyết việc truyền dữ liệu. Sự ra đời các chuẩn Bluetooth, Wifi đã định hình thị trường, tuy nhiên Truyền tín hiệu qua mạng GSM với công nghệ tin nhắn SMS có những cách tiếp cận rất khác:

- + Truyền dữ liệu đi xa và khắc phục hạn chế về vật cản.
- + Độ ổn định cao.
- + Chi phí thấp.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

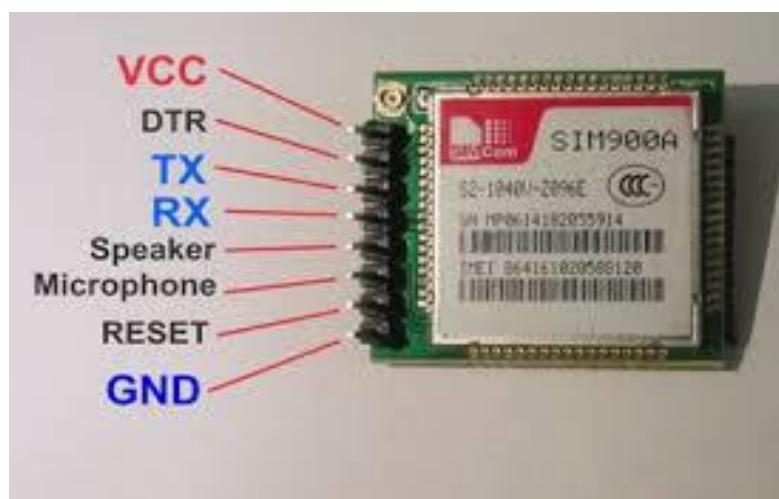
### a. Giới thiệu về module Sim 900A

Module Sim 900A là một module GSM/GPRS cực kỳ nhỏ gọn, được SIMCOM thiết kế cho thị trường toàn cầu. Sim900A hoạt động được ở 4 băng tần GSM 850MHz, EGSM 900MHz, DCS 1800MHz và PCS 1900MHz như là một loại thiết bị đầu cuối với một Chip xử lý đơn nhân đầy sức mạnh, tăng cường các tính năng quan trọng dựa trên nền vi xử lý ARM926EJ-S, cho người dùng nhiều lợi ích từ kích thước nhỏ gọn (24x24 mm), đáp ứng những yêu cầu về không gian trong các ứng dụng M2M (Mobile to Mobile), trong các SmartPhone, PDA, các thiết bị định vị cầm tay và các thiết bị di động khác hoặc các ứng dụng của AVL (Automated Vehicle Location) và cho các dịch vụ định vị khác,...

Giao tiếp vật lý trong ứng dụng điện thoại của SIM 900A là 68 chân, nó cung cấp tất cả các giao diện vật lý giữa module Sim và board mạch của khách hàng:

- Có cổng truyền dữ liệu nối tiếp và cổng gõ lỗi giúp dễ dàng hơn trong việc phát triển ứng dụng.
- Một kênh âm thanh bao gồm ngõ vào microphone và ngõ ra loa.
- Bàn phím và giao diện hiển thị SPI sẽ linh hoạt để phát triển các ứng dụng.
- Có lập trình Input/Output.
- Giao tiếp với simcard giống như điện thoại di động. SIM900A được thiết kế với công nghệ tiết kiệm năng lượng vì vậy mức tiêu thụ chỉ ở mức 1.5mA ở trong chế độ ngủ. SIM900A hỗ trợ giao thức TCP/IP, rất hữu ích cho việc truyền dữ liệu trên internet.

Vị trí các chân của sim 900A theo thứ tự như hình 2.6.



**Hình 2.6.** Vị trí chân sim 900A

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

### b. Các tính năng chính của SIM 900A

Các tính năng chính của module Sim 900A được biểu thị trong bảng 2.1 như sau:

**Bảng 2.1.** Các tính năng chính của SIM 900A

Tính năng	Mô tả thực thi
Nguồn cung cấp	- Sử dụng điện áp từ 3.4. đến 4.5V.
Nguồn tiết kiệm	- Sử dụng điển hình ở chế độ ngủ với dòng 1.5mA.
Các dải tần hoạt động	- GSM850, EGSM900, DCS1800, PCS1900 có thể tìm thấy các dải tần một cách tự động. Các dải tần có thể được thiết lập bởi câu lệnh AT. - Tương thích với pha GSM 2/2+.
Lớp GMS	- MS nhỏ.
Công suất truyền tải	- Lớp 4(2W) ở GSM850 và EGSM 900. - Lớp 1(1W) ở DCS 1800 và PCS 1900.
Kết nối GPRS	- GPRS nhiều khe mặc định trong lớp 10. - GPRS nhiều khe tùy chọn ở lớp 8. - GPRS trạm di động lớp B.
Nhiệt độ hoạt động	- Hoạt động bình thường ở -30 đến 80 độ C. - Hoạt động hạn chế ở -40 đến 30 và 80 đến 85 độ C. - Nhiệt độ lưu trữ là -45 đến 90 độ C.
Dữ liệu GPRS CSD	- Truyền dữ liệu xuống lớn nhất là 85.6 kbps. - Truyền dữ liệu lên lớn nhất 42.8 kbps. - Mã hóa chương trình CS-1, CS-2, CS-3 và CS-4. - SIM900 hỗ trợ các giao thức PAP (Giao thức xác nhận mật mã) thường được sử dụng trong các kết nối PPP. - SIM900 được tích hợp giao thức TCP/IP. - Cung cấp gói chuyển mạch kênh điều khiển quảng bá (PBCCH). - Các tốc độ truyền CSD: 2.4, 4.8, 9.6, 14.4 kbps, không trong suốt. - Hỗ trợ dịch vụ dữ liệu bù xung phi cấu trúc.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

SMS	<ul style="list-style-type: none"> <li>- MODULE, MO, CB, Text và chế độ PDU.</li> <li>- Lưu trữ SMS: thẻ SIM.</li> </ul>
Giao tiếp SIM	<ul style="list-style-type: none"> <li>- Cung cấp thẻ SIM: 1.8, 3V.</li> </ul>
Cổng nối tiếp và cổng gỡ lỗi	<ul style="list-style-type: none"> <li><b>- Cổng nối tiếp:</b> <ul style="list-style-type: none"> <li>• Giao tiếp modem 8 dây với các đường trạng thái và đường dữ liệu, không cân bằng, không đồng bộ.</li> <li>• 1.2 kbps đến 11.52 kbps.</li> <li>• Cổng nối tiếp có thể sử dụng được cho lệnh AT và luồng dữ liệu.</li> <li>• Hỗ trợ RTS/CTS phần cứng và phần mềm điều khiển luồng ON/OFF.</li> <li>• Kết hợp khả năng theo giao thức hợp kênh GSM 07.10.</li> <li>• Hỗ trợ các tốc độ baud tự động từ 1200 bps đến 115200 bps.</li> </ul> </li> <li><b>- Cổng gỡ lỗi (debug)</b> <ul style="list-style-type: none"> <li>• Giao tiếp 2 dây trống DBG_TXD và DBG_RXD.</li> <li>• Có thể sử dụng để gỡ lỗi hoặc cập nhật Firmware.</li> </ul> </li> </ul>
Chức năng định thời	<ul style="list-style-type: none"> <li>- Lập trình thông qua lệnh AT.</li> </ul>
Đặc điểm vật lý	<ul style="list-style-type: none"> <li>- Kích cỡ: 24mm x 24mm x 3mm.</li> <li>- Trọng lượng 3.4g</li> </ul>
Cập nhật Firmware	<ul style="list-style-type: none"> <li>- Cập nhật Firmware bởi cổng gỡ lỗi.</li> </ul>

### c. Tập lệnh AT cơ bản cho SIM 900A

Các tập lệnh cơ bản của Sim 900A được biểu diễn trong bảng 2.2 như sau:

**Bảng 2.2.** Một số lệnh AT cơ bản

Lệnh AT	Chức năng
AT+CMGD	Xóa tin nhắn sms.
AT+CMGF	Định dạng văn bản tin nhắn.
AT+CMGL	Danh sách tin nhắn đã lưu.
AT+CMGR	Lệnh đọc tin nhắn.
AT+CMGS	Lệnh gửi tin nhắn.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

AT+CMGW	Lưu tin nhắn vào bộ nhớ.
AT+CMSS	Gửi tin nhắn đã lưu.
AT+CMGC	Gửi sms lệnh.
AT+CNMI	MODULE gửi thông báo khi có tin nhắn mới.
AT+CPMS	Các tin nhắn riêng biệt được lưu.
AT+CRES	Cài đặt lại tin nhắn.
AT+CSAS	Lưu các cài đặt cho tin nhắn.
AT+CSCA	Địa chỉ dịch vụ tin nhắn.
AT+CSMP	Cài đặt định dạng chữ của tin nhắn.
AT+CSMS	Lựa chọn tin nhắn dịch vụ.
ATD số điện thoại;	Lệnh thực hiện cuộc gọi.
ATH	Lệnh kết thúc cuộc gọi.

### ➤ Giao tiếp với Sim 900A qua lệnh AT:

Việc điều khiển Sim 900A được thực hiện thông qua việc truyền các lệnh AT (các lệnh này thường bắt đầu bằng “AT”, dùng để điều khiển các thiết bị tương tác với mạng di động).

Ví dụ:

- Lệnh AT để gửi một tin nhắn:  
AT+CMGS="”+84906619001”>Tin nhắn cần gửi<Ctrl+Z>
- Lệnh AT để lấy danh sách tin nhắn:  
AT+CMGL="ALL"
- Lệnh AT để đọc tin nhắn vừa nhận:  
AT+CMGR=1
- Lệnh AT để xóa một tin nhắn:  
AT+CMGD=1

Tập lệnh AT cần dùng:

- Cấu hình tin nhắn:

AT+CMGF= [<MODE>]

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

MODE =1: định dạng cho tin nhắn dạng Text. MODE =0: định dạng cho tin nhắn dạng PDU

- Gửi tin nhắn:

- + Đổi với tin nhắn có định dạng Text:

- AT+CMGS=<da>[,<toda>]<CR><“Nội dung tin nhắn”><Ctrl+Z>

- <da>: Vùng chứa giá trị dạng String

- <toda>: Nếu giá trị đầu tiên của <da> là dấu “+” thì <toda> sẽ được định dạng là integer mặc định là 145 hay 129 chữ số.

- + Đổi với tin nhắn có định dạng PDU:

- AT+CMGS=<length><CR>“Nội dung tin nhắn”<Ctrl+Z>

- <length>: giá trị integer (không quá 160 bytes) để lưu chiều dài nội dung tin nhắn.

- Đọc tin nhắn vừa nhận:

- AT+CMGR=<index>[,<mode>]

- <index>: kiểu integer, vị trí của ô nhớ của bộ nhớ liên quan.

- <mode>: 0 - Bình thường

- 1 - Không thay đổi trạng thái của tin nhắn riêng biệt.

- Xóa tin nhắn:

- AT+CMGD=<index>[,<delflag>]

- <index>: kiểu integer, vị trí của ô nhớ của bộ nhớ liên quan.

- <delflag>: 0 - xóa tin nhắn theo chỉ số <index>

- 1 - xóa tất cả tin nhắn trừ các tin nhắn chưa đọc.

- 2 - xóa tất cả các tin nhắn đã đọc và gửi lại tin nhắn gốc trừ các tin nhắn chưa đọc.

- 3 - xóa tất cả các tin nhắn và không gửi lại tin nhắn gốc trừ các tin nhắn chưa đọc.

- 4 - xóa tất cả tin nhắn.

- Thực hiện cuộc gọi:

- ATD0906619001;

- Hủy cuộc gọi:

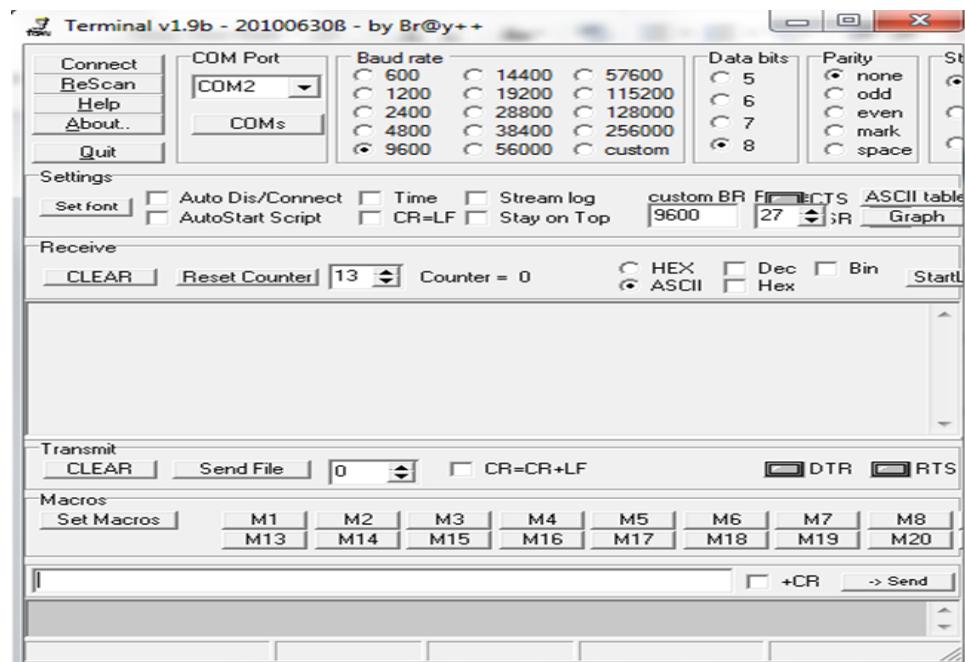
- ATH

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### ➤ Các bước cấu hình ban đầu cho Sim 900A:

- Chuẩn bị phần mềm:

Sử dụng phần mềm Terminal để giao tiếp máy tính với Sim900A trong hình 2.7.



Hình 2.7. Giao diện phần mềm Terminal

- Chuẩn bị phần cứng kết nối: Sim 900A và mạch chuyển USB UART (hình 2.8).



Hình 2.8. Mạch chuyển USB UART PL2303

- Các bước thực hiện:

#### Bước 1: Khởi động Sim 900

Sau khi khởi động Sim900 màn hình sẽ hiện:

RDY

+CFUN: 1

+CPIN: READY

Call Ready

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

### Bước 2: Kiểm tra đường truyền

Để kiểm tra đường truyền ta gởi lệnh AT cho Sim. Từ bàn phím gõ AT sau đó Enter. Màn hình sẽ hiện:

AT

OK

### Bước 3: Kiểm tra nhận tin nhắn

Câu hình tin nhắn của Sim 900 với định dạng là kiểu Text.

Cấu trúc lệnh : AT+CMGF=1 . Sau đó nhấn Enter.

Gửi tin nhắn đến Sim 900 từ một điện thoại khác. Nếu Sim hoạt động tốt màn hình sẽ hiện:

+CM: "+8401655162564", "", "16/20/07,09:10:30+28"

Test module sim900

### 2.4.3 Module cảm biến vân tay R308 (Fingerprint R308)

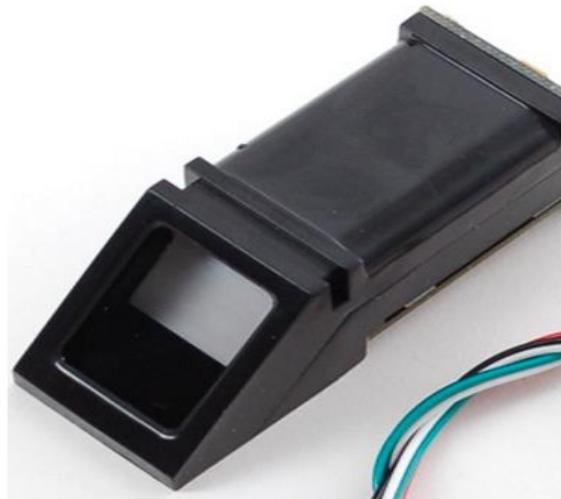
Hiện tại trên thị trường có rất nhiều loại cảm biến để nhận dạng và phát hiện người dùng như: cảm biến nhận dạng khuôn mặt, cảm biến hồng ngoại, cảm biến giọng nói, RFID,... nhưng với tính bảo mật và tiện lợi của cảm biến vân tay rất phù hợp cho việc quản lý khóa và mở cửa.

Đây là module nhận dạng vân tay giao tiếp trực tiếp qua giao thức UART có thể kết nối trực tiếp đến vi điều khiển hoặc qua PC adapter Max232/USB-Serial. Người sử dụng có thể lưu trữ dữ liệu vân tay trực tiếp vào module. Module có thể dễ dàng giao tiếp với các loại vi điều khiển chuẩn 3.3V hoặc 5V. Có một con Led xanh được bật sáng nằm sẵn trong ống kính trong suốt quá trình chụp vân tay. Cảm biến với độ chính xác cao và có thể được nhúng vào các thiết bị như: điều khiển truy cập, két sắt, khóa cửa nhà, khóa cửa xe,...

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Trong hình 2.9 là hình ảnh thực tế của module cảm biến vân tay R308.



**Hình 2.9.** Module cảm biến vân tay R308

### a. 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.

### b. 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...

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

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.

### c. Thông số kỹ 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 °C ~ + 40 °C

+ Độ ẩm tương đối: 10% - 85%

Môi trường bảo quản:

+ Nhiệt độ: -40 °C ~ + 85 °C

+ Độ ẩm tương đối: <85%

### d. Giao tiếp phần cứng

Giao tiếp phần cứng của module R308 được thể hiện qua bảng 2.3:

**Bảng 2.3.** Các chân kết nối của module R308

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).

### e. Giao thức truyền thông nối tiếp không đồng bộ UART

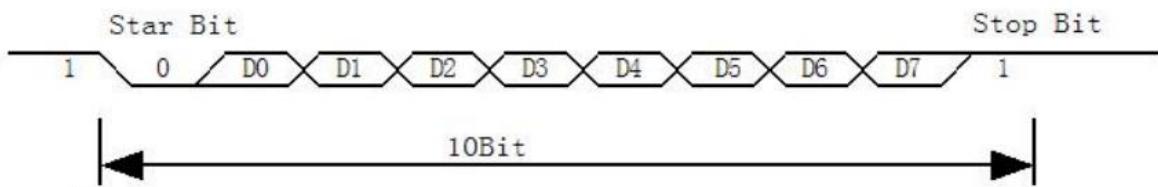
Được truyền theo chế độ nối tiếp bán song công bất đồng bộ. Tốc độ baud truyền mặc định là 57600 bps và có thể cài đặt tốc độ này trong dải từ 9600 – 115200. Tại thời điểm bật nguồn, nó sẽ tốn 300ms cho việc thiết lập.

Khung truyền với định dạng 10bit: với 1 bit bắt đầu (start bit) ở mức logic ‘0’, 8 bit dữ liệu với bit đầu LBS và 1 bit kết thúc (stop bit). Không có bit kiểm tra (check bit).

Dữ liệu được truyền đi trên chân TX gồm 1 start bit (mức ‘0’), data và 1 stop bit (mức ‘1’). Tốc độ truyền: đơn vị bit per second (bps) còn gọi là Baud (số lần thay đổi tín hiệu trong 1 giây – thường sử dụng cho modem). UART là phương thức truyền nhận bất đồng bộ, nghĩa là bên nhận và bên phát không cần phải có chung tốc độ xung clock (ví dụ: xung clock của vi điều khiển khác xung clock của máy tính). Khi đó bên truyền muốn truyền dữ liệu sẽ gửi start bit (bit ‘0’) để báo cho bên thu biết để bắt đầu nhận dữ liệu và khi truyền xong dữ liệu thì stop bit (bit ‘1’) sẽ được gửi để báo cho bên thu biết kết thúc quá trình truyền.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Khi có start bit thì cả hai bên sẽ dùng chung 1 xung clock (có thể sai khác một ít) với độ rộng 1 tín hiệu (0 hoặc 1) được quy định bởi baud rate, ví dụ baud rate = 9600 bps nghĩa là độ rộng của tín hiệu 0 (hoặc 1) là  $1/9600 = 104$  ms và khi phát thì bên phát sẽ dùng baud rate chính xác (ví dụ 9600 bps) còn bên thu có thể dùng baud rate sai lệch 1 ít (9800bps chẳng hạn). Truyền bắt đồng bộ sẽ truyền theo từng frame và mỗi frame có cấu trúc như trong hình 2.10 sau đây:



**Hình 2.10.** Giao thức truyền thông của R308

Ngoài ra trong frame truyền có thể có thêm bit odd parity (bit lẻ) hoặc even parity (bit chẵn) để kiểm tra lỗi trong quá trình truyền. Bit parity này có đặc điểm nếu sử dụng odd parity thì số các bit ‘1’ + odd parity bit sẽ ra một số lẻ còn nếu sử dụng even parity thì số các bit ‘1’ + even parity bit sẽ ra một số chẵn.

Module sẽ kết nối với MCU theo kết nối sau: TXD (chân 3 của module) kết nối với RXD (chân nhận của MCU), RXD (chân 4 của module) kết nối với TXD (chân truyền của MCU).

### f. Tài nguyên hệ thống

#### ➤ Bộ đếm:

Có một bộ đếm hình ảnh và hai 512 byte tệp kí tự đệm bên trong không gian bộ nhớ RAM của module. Người dùng có thể đọc và viết bất kỳ của bộ đếm bằng cách hướng dẫn.

**Lưu ý:** Nội dung của bộ đếm trên sẽ bị mất khi tắt nguồn.

#### - Bộ đếm hình ảnh:

Bộ đếm hình ảnh phục vụ cho việc lưu trữ hình ảnh và các định dạng hình ảnh là 256 \* 288 pixel. Khi truyền qua UART, để đẩy nhanh tốc độ, chỉ có 4 bit cao của các điểm ảnh được truyền (có nghĩa là 16 độ xám). Và hai điểm ảnh lân cận của cùng hàng sẽ hình thành một byte trước khi truyền. Khi tải lên máy tính, hình ảnh 16-xám-độ sẽ được mở rộng sang định dạng 256 mức xám. Đó là định dạng BMP 8-bit. Khi chuyển qua USB, hình ảnh 8 bit pixel, đó là 256 mức xám.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

### - Bộ đệm tệp kí tự:

Bộ đệm kí tự CharBuffer1, CharBuffer2, có thể được sử dụng để lưu trữ cả tệp kí tự và tệp mẫu.

### ➤ Thư viện vân tay:

Hệ thống đặt ra một không gian nhất định trong Flash cho mẫu dấu vân tay lưu trữ, đó là thư viện vân tay. Nội dung của thư viện vẫn còn khi tắt nguồn.

Dung lượng của thư viện thay đổi dung lượng của Flash, hệ thống sẽ nhận biết sau khi tự động. Lưu trữ dấu vân tay mẫu trong Flash là theo tuần tự.

### ➤ Cấu hình các thông số của hệ thống:

#### - Kiểm soát tốc độ baud(Thông số thứ: 6):

Các thông số điều khiển UART tốc độ truyền thông của Module. Giá trị của nó là một số nguyên N, N = [1, 12]. Tỷ lệ tương ứng là 9600 baud \* N bps.

#### - Mức độ bảo mật(Thông số thứ: 5):

Các thông số kiểm soát các giá trị ngưỡng phù hợp với tìm kiếm của dấu vân tay và đối chiếu. Mức độ bảo mật được chia thành 5 lớp và giá trị tương ứng là 1, 2, 3, 4, 5. Ở cấp độ 1, FAR là cao nhất và FRR là thấp nhất. Tuy nhiên ở cấp độ 5, FAR là thấp nhất và FRR là cao nhất.

#### - Độ dài gói dữ liệu(Thông số thứ: 7):

Các thông số quyết định độ dài tối đa của các gói dữ liệu chuyển giao khi giao tiếp với máy tính trên. Giá trị của nó là 0, 1, 2, 3, tương ứng với 32 bytes, 64 byte, 128 byte, 256 byte tương ứng.

### ➤ Thanh ghi trạng thái hệ thống:

Ghi trạng thái hệ thống cho biết tình trạng hoạt động hiện tại của Module. Chiều dài của nó là 1 word, và có thể được đọc qua hướng dẫn ReadSysPara. Định nghĩa của thanh ghi được biểu hiện qua hình 2.11 như sau:

Bit Num	15	4	3	2	1	0
Description	Reserved		ImgBufStat	PWD	Pass	Busy

**Hình 2.11.** Định nghĩa của thanh ghi

#### Chú ý:

- Busy: 1 bit. 1: Hệ thống là lệnh thực thi; 0: hệ thống thì rãnh;
- Pass: 1 bit. 1: tìm thấy ngón tay phù hợp; 0: sai ngón tay;

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

- PWD: 1 bit. 1: xác minh mật khẩu bắt tay của thiết bị.
- ImgBufStat: 1 bit. 1: bộ đệm hình ảnh chứa hình ảnh hợp lệ.

### ➤ Mật khẩu module:

Khi mở nguồn lại, hệ thống kiểm tra đầu tiên cho dù mật khẩu bắt tay đã được sửa đổi. Nếu không, hệ thống máy tính trên xét thấy không có yêu cầu xác minh mật khẩu và sẽ bước vào chế độ hoạt động bình thường. Đó là, khi mật khẩu vẫn là mặc định, quá trình xác minh có thể được bỏ qua. Chiều dài mật khẩu là 4 byte, và giá trị nhà máy mặc định của nó là 0xFF, 0xFF, 0xFF, 0xFF.

Nên thay đổi mật khẩu, tham khảo hướng dẫn SetPwd, sau đó Module (hoặc thiết bị) bắt tay mật khẩu phải được xác nhận trước khi hệ thống đi vào chế độ hoạt động bình thường. Hoặc nếu không, hệ thống sẽ từ chối thực hiện và lệnh.

### ➤ Địa chỉ module:

Mỗi module có một địa chỉ xác định. Khi giao tiếp với máy tính trên, mỗi lệnh / dữ liệu được chuyển giao theo hình thức gói dữ liệu, trong đó có các mục địa chỉ. Module hệ thống chỉ phản ứng với các gói dữ liệu có mục địa chỉ giá trị là giống với địa chỉ xác định của nó. Chiều dài địa chỉ là 4 byte, và giá trị mặc định của nó là 0xFFFFFFFF. Người dùng có thể thay đổi địa chỉ qua SetAdder. Địa chỉ đổi mới vẫn còn lưu lại khi tắt nguồn.

### ➤ Bộ tạo số ngẫu nhiên:

Module tích hợp một phần cứng 32-bit Bộ tạo số ngẫu nhiên (RNG). Qua GetRandomCode, hệ thống sẽ tạo ra một số ngẫu nhiên và tải nó lên.

### g. 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 hình 2.12:

Header	Adder	Package identifier	Package length	Package content (instruction/data/Parameter)	Checksum
--------	-------	--------------------	----------------	--	----------

Hình 2.12. Định dạng gói dữ liệu

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

Trong đó, các thông số chi tiết hơn được biểu diễn qua bảng 2.4 sau:

**Bảng 2.4.** 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 identifier	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.

### h. 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.

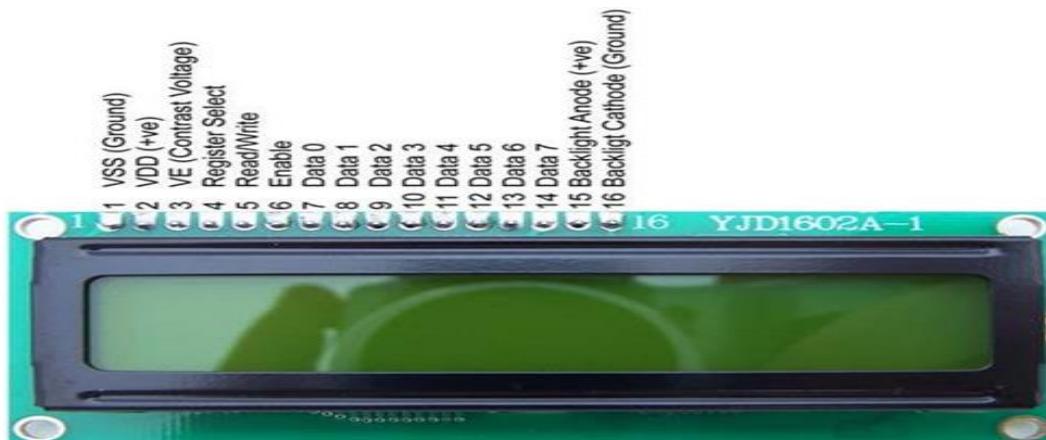
Định nghĩa bytes xác nhận:

- **0x00h:** thực thi hoàn tất.
- **0x01h:** lỗi nhận dữ liệu.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

- **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 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.

### 2.4.4 Màn hình LCD



Hình 2.13. Màn hình LCD 16x2

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

Hình ảnh thực tế màng hình LCD 16x2 và chi tiết các chân của LCD được biểu hiện lần lượt qua hình 2.13 và bảng 2.5 như sau:

**Bảng 2.5.** Các chân của LCD

Thứ tự	Tên kí hiệu	I/O	Mô tả
1	Vss	Power	GND
2	Vdd	Power	+5V
3	V <sub>0</sub>	Analog	Điều khiển ánh sáng nền
4	RS	Input	Register Select
5	R/W	Input	Read/Write
6	E	Input	Enable (Storage)
7	D0	I/O	Data LSB
8	D1	I/O	Data
9	D2	I/O	Data
10	D3	I/O	Data
11	D4	I/O	Data
12	D5	I/O	Data
13	D6	I/O	Data
14	D7	I/O	Data MSB
15	A	I	Nguồn dương 5V
16	K	I	GND

- **Chân cấp nguồn:** Vss nối mass (0V), VDD (nối nguồn +5V), V<sub>0</sub> (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.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

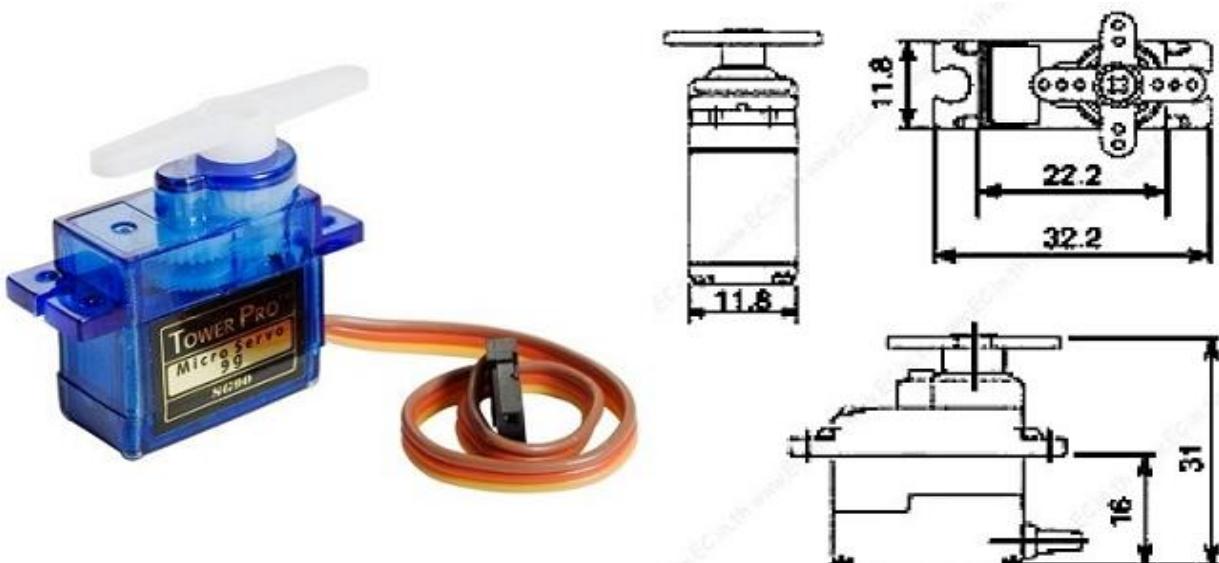
+ Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào (chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E.

+ 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.

### 2.4.5 Giới thiệu động cơ Servo SG90 9g Micro

Hình 2.14 là ảnh thực tế và kích thước chi tiết của động cơ Servo SG90 9g Micro:



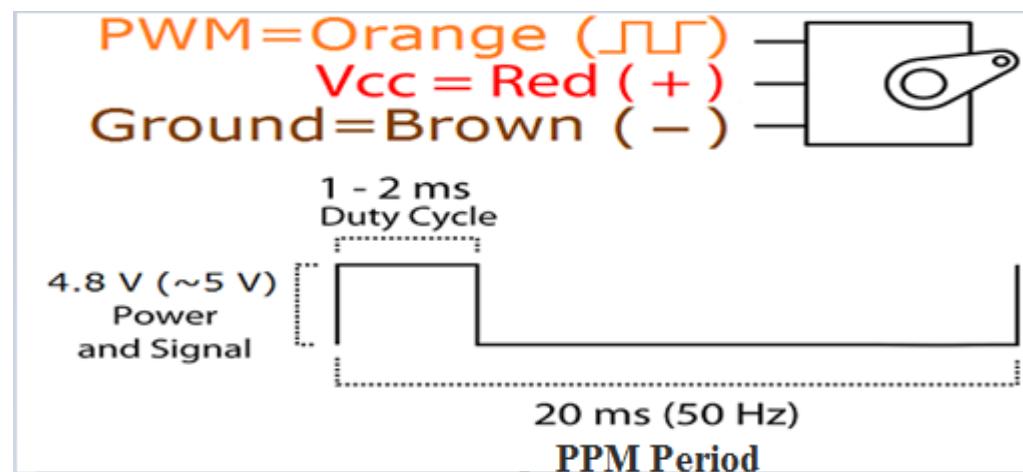
Hình 2.14. Động cơ Servo SG90 9g Micro

#### a. Thông số kỹ thuật của Servo SG90 9g Micro

- Khối lượng : 9g
- Kích thước: 22.2x11.8.32 mm
- Momen xoắn: 1.8kg/cm
- Tốc độ hoạt động: 60 độ trong 0.1 giây
- Điện áp hoạt động: 4.8V(~5V)
- Nhiệt độ hoạt động: 0 °C – 55 °C

#### b. Điều khiển và kết nối

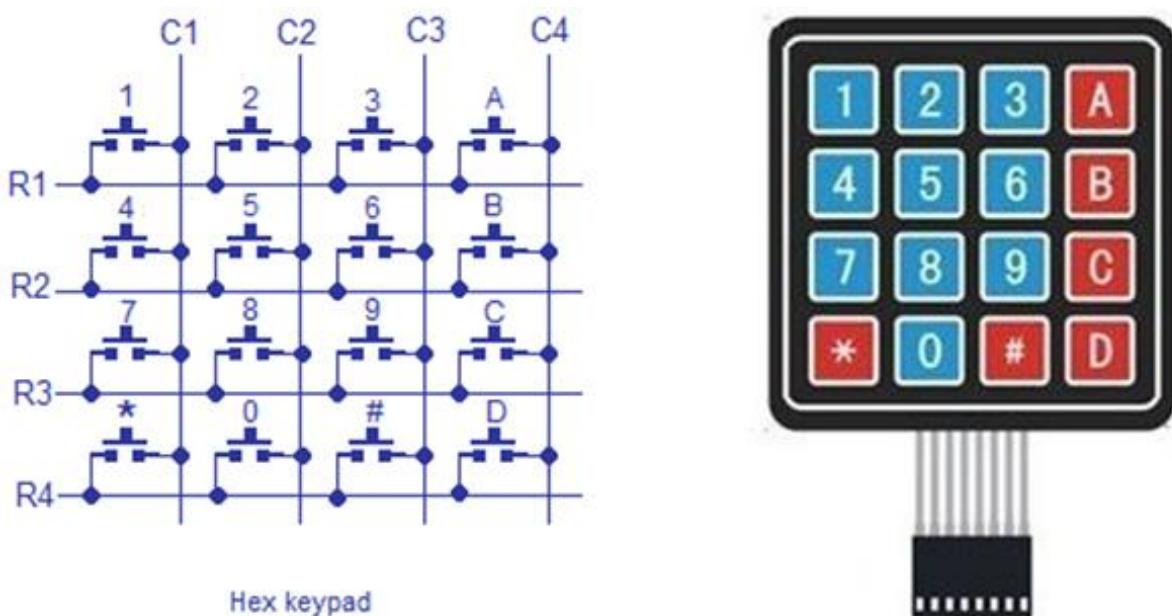
Kết nối dây màu đỏ với 5V, dây màu nâu với mass, dây màu cam với chân phát xung của vi điều khiển. Ở chân xung cấp một xung từ 1ms - 2ms theo để điều khiển góc quay theo ý muốn. Cách kết nối và điều khiển của động cơ Servo SG90 9g Micro được thể hiện qua hình 2.15:



Hình 2.15. Điều khiển và kết nối của động cơ servo SG90 9g Micro

#### 2.4.6 Giới thiệu bàn phím ma trận 4x4 (Keypad 4x4)

Sơ đồ nối dây và hình ảnh thực tế của bàn phím ma trận 4x4 được giới thiệu qua hình 2.16 bên dưới.



Hình 2.16. Sơ đồ nối dây và hình ảnh thực tế của bàn phím ma trận 4x4

##### a. Khái niệm

Keypad là một "thiết bị nhập" chứa các nút nhấn cho phép người dùng nhập các chữ số, chữ cái hoặc ký hiệu vào bộ điều khiển. Keypad không chứa tất cả bảng mã ASCII như keyboard và vì thế keypad thường được tìm thấy trong các thiết bị chuyên dụng. Các nút nhấn trên các máy tính điện tử cầm tay là một ví dụ về keypad. Số lượng nút nhấn của một keypad thay đổi phụ thuộc vào yêu cầu ứng dụng. Gọi là keypad 4x4 vì keypad này có 16 nút nhấn được bố trí dạng ma trận 4 hàng và 4 cột. Cách bố trí ma

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

---

trận hàng và cột là cách chung mà các keypad sử dụng. Cũng giống như các ma trận LED, các nút nhấn cùng hàng và cùng cột được nối với nhau, vì thế với keypad 4x4 sẽ có tổng cộng 8 ngõ ra (4 hàng và 4 cột).

### b. Hoạt động của Keypad 4x4

Theo hình trên, giả sử nút ‘2’ được nhấn, khi đó đường R1 và C2 được nối với nhau. Giả sử đường C2 được nối với GND (mass, 0V) thì R1 cũng sẽ là GND. Tuy nhiên, bằng cách kiểm tra trạng thái đường R1 chúng ta sẽ không kết luận nút ‘2’ được nhấn. Giả sử tất cả các đường C1, C2, C3, C4 đều nối với GND, nếu R1= GND thì rõ ràng ta không thể kết luận nút ‘1’ hay nút ‘2’ hay nút ‘3’ hay nút ‘A’ được nhấn. Kỹ thuật để khắc phục vấn đề này chính là kỹ thuật “quét” keypad. Có 2 cách quét phím là quét theo cột hoặc quét theo hàng. Sau đây là ví dụ về quét theo hàng, quét cột cũng hoàn toàn tương tự:

- Ta lần lượt xuất tín hiệu mức 0 ra các hàng (khi một hàng là mức ‘0’ thì tất cả các hàng khác phải là mức 1).
- Sau đó kiểm tra các cột nếu cột nào có mức logic 0 thì phím có tọa độ hàng và cột đó được án.

### c. Thông số kỹ thuật và Keypad 4x4

- Module bàn phím ma trận 4x4 loại phím mềm.
- Dòng / áp hoạt động tối đa: 30 mA, 24 VDC.
- Tuổi thọ hoạt động: khoảng 1.000.000 lần nhấn phím.
- Thời gian phản hồi:  $\leq 5\text{ms}$ .
- Độ dài cáp: 88mm.
- Nhiệt độ hoạt động  $0 \sim 70^\circ\text{C}$
- Đầu nối ra 8 chân.
- Kích thước bàn phím 77 x 69 mm.

### d. Ứng dụng của Keypad 4x4

Ý tưởng ứng dụng:

- Hệ thống an ninh bảo vệ bằng mật khẩu.
- Nhập liệu lựa chọn menu, điều khiển thiết bị.
- Nhập dữ liệu cho các hệ thống nhúng.

## **Chương 3. TÍNH TOÁN VÀ THIẾT KẾ**

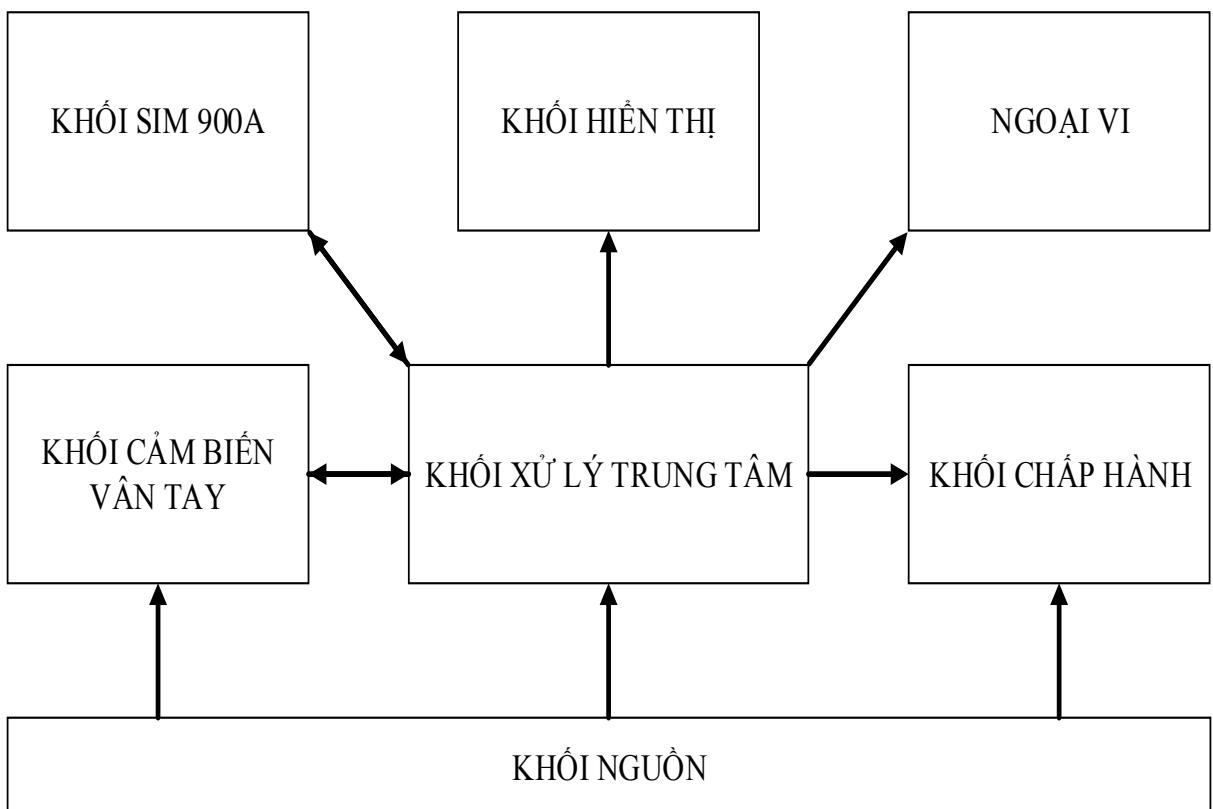
### **3.1 GIỚI THIỆU**

Đề tài thiết kế hệ thống đóng mở cửa thông minh được bảo mật qua vân tay và điện thoại. Kiểm tra trạng thái cửa và báo về cho người dùng khi có yêu cầu.

### **3.2 TÍNH TOÁN VÀ THIẾT KẾ HỆ THỐNG**

#### **3.2.1 Thiết kế sơ đồ khối hệ thống**

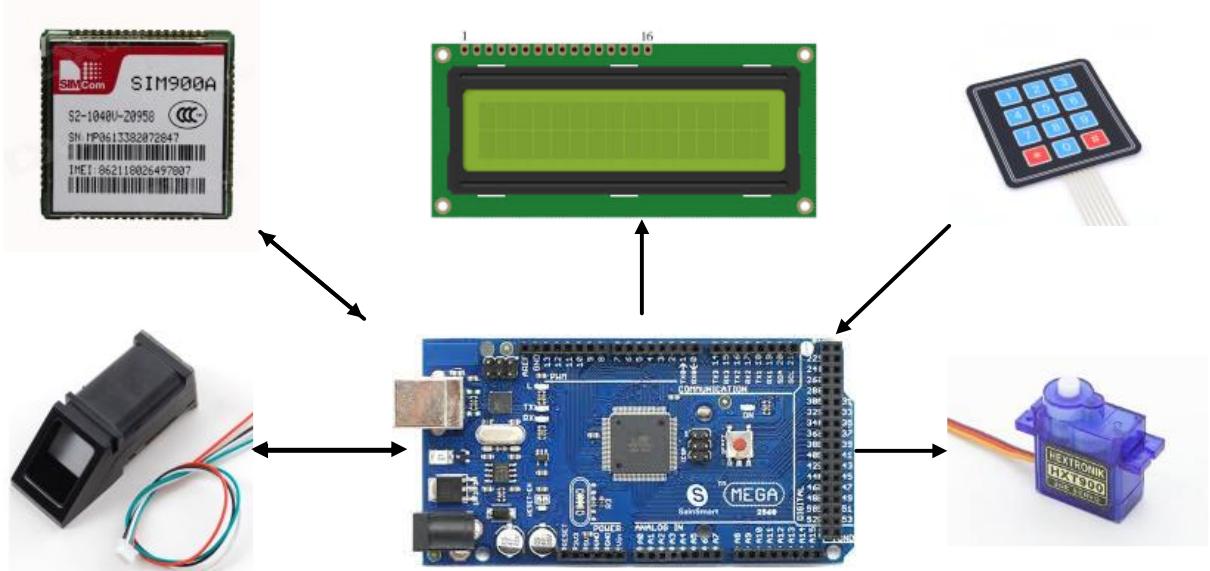
Hệ thống gồm 7 khối ghép lại với nhau theo nhiều hướng tạo nên một hệ thống đóng mở cửa hoàn chỉnh hoạt động ổn định được trình bày trong sơ đồ khối hình 3.1 như sau:



**Hình 3.1** Sơ đồ khối hệ thống

### CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Đối với hình 3.2 là hình ảnh thực tế của các khối trong sơ đồ hệ thống đóng mở cửa kết nối với nhau.



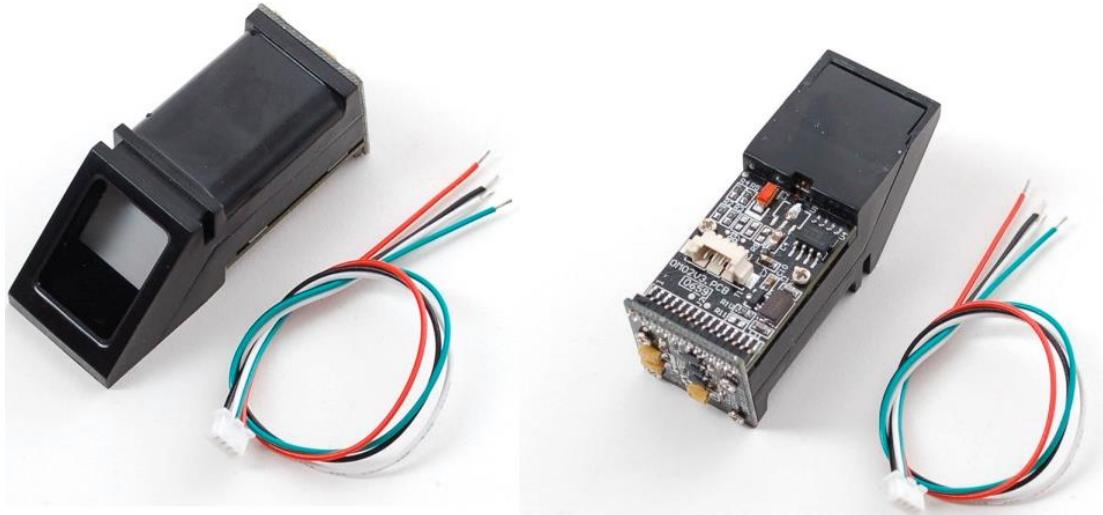
**Hình 3.2** Sơ đồ khối hệ thống thiết bị thực tế

#### Chức năng từng khối:

- Khối nguồn:** cung cấp nguồn 5V DC cho Sim 900A, khối hiển thị, cảm biến vân tay và khối chấp hành, nguồn 9V DC cấp nguồn cho khối xử lý trung tâm board Arduino Mega 2560. Sử dụng hai bộ adapter nguồn thiết kế sẵn.
- 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 R308 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 mega 2560 thực hiện.
- Khối chấp hành:** sử dụng để đóng mở chốt cửa do động cơ servo 9g thực hiện.
- Khối Sim 900A:** dùng để điều khiển đóng mở cửa qua hai cách là dùng cuộc gọi và tin nhắn. Kiểm tra trạng thái cửa đang đóng hay mở.
- Khối hiển thị:** hiển thị thông tin chế độ hoạt động trên LCD và các led báo trạng thái chế độ.
- Khối ngoại vi:** bao gồm nút nhấn hành trình báo trạng thái cửa và bàn phím ma trận cho phép nhập mật khẩu và ID người dùng.

### 3.2.2 Tính toán và thiết kế mạch

#### a. Khối cảm biến vân tay



**Hình 3.3** Cảm biến vân tay R308

Cảm biến vân tay R308 trên hình 3.3 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, và hiệu suất tuyệt vời.

- + 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.

**Bảng 3.1.** Chức năng chân cảm biến vân tay R308

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)

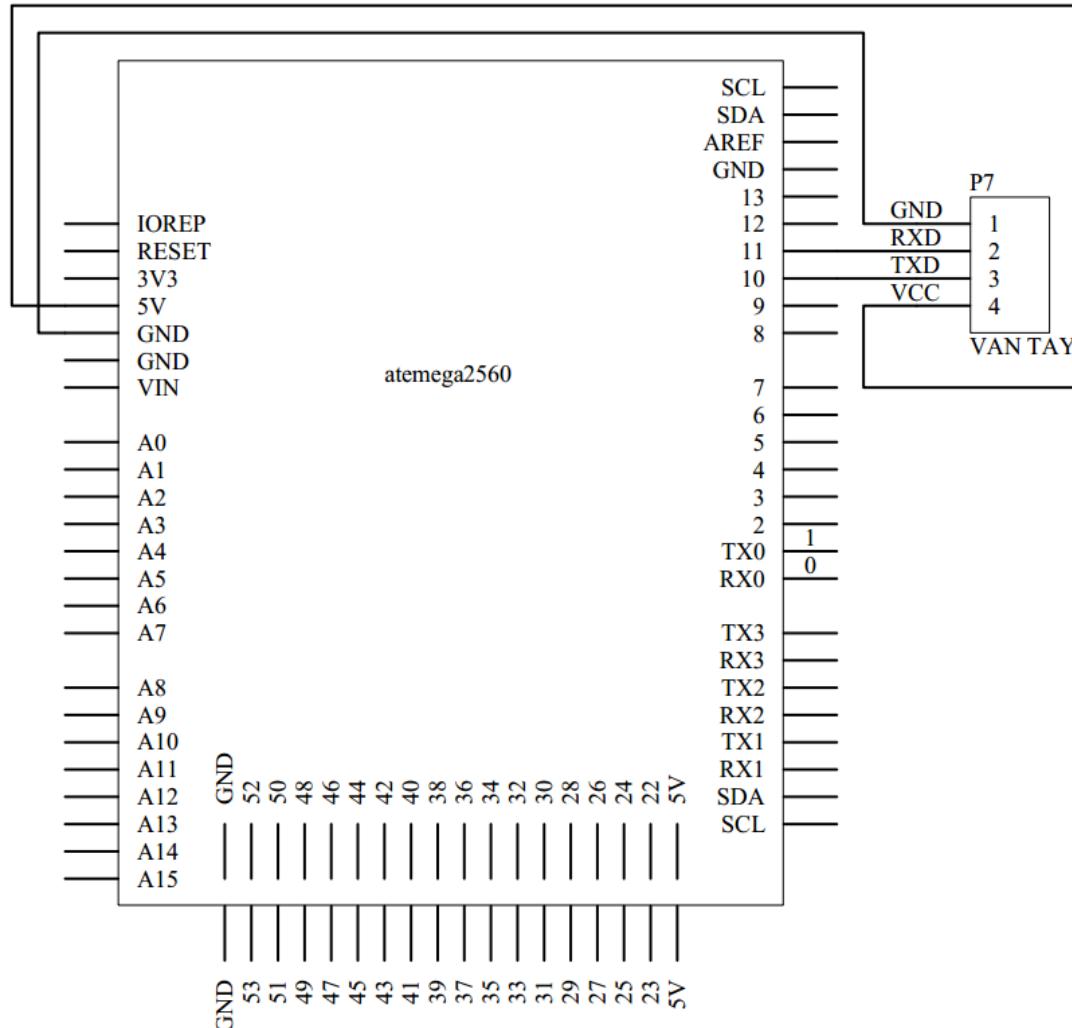
## CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Cách nối dây cho cảm biến vân tay R308 vào Arduino Mega 2560 theo thứ tự chân ở bảng 3.1 ở trên như sau:

- Dây số 1 và số 6 không cần sử dụng.
- Dây số 2 nối vào nguồn 5V và dây số 5 nối vào chân GND của arduino.
- Dây số số 3 nối vào chân số 10 và dây số 4 nối chân 11 của arduino mega.

Vì ở đây ta sử dụng thư viện Software Serial có chân 10 11 hỗ trợ ngắt sẽ làm một cổng Serial ảo để trao đổi dữ liệu.

Còn đây là cách nối dây vào arduino với cảm biến vân tay thể hiện trong hình 3.4

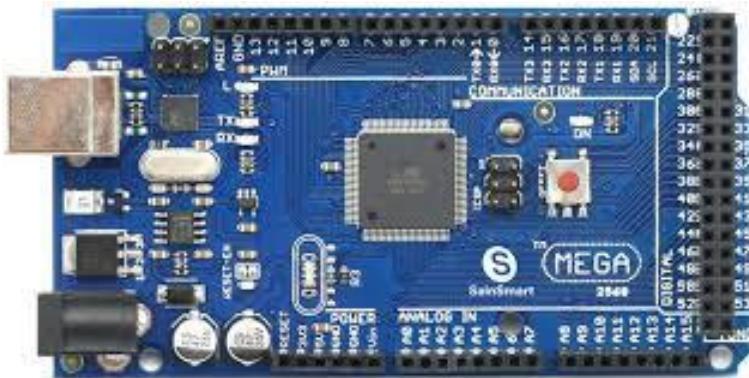


Hình 3.4 Sơ đồ nguyên lý cảm biến vân tay kết nối Arduino Mega

### b. Khối xử lý trung tâm

Vì Arduino Mega 2560 sử dụng chip ATmega2560. Nó có 54 chân digital I/O (trong đó có 15 chân điều chế độ rộng xung PWM), 16 chân đầu vào tương tự (Analog Inputs), 4 UARTs (cổng nối tiếp phần cứng), một thạch anh dao động 16 MHz, kết nối USB, một jack cảm điện, một đầu ICSP và một nút reset như trong hình 3.5. Nó chứa tất cả mọi thứ cần thiết để tạo thành khối xử lý trung tâm với đầy đủ các port .

### CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ



**Hình 3.5** Khối xử lý trung tâm sử dụng board arduino mega 2560

Thông số kỹ thuật của Arduino Mega 2560 được thể hiện trong bảng 3.2 dưới đây.

**Bảng 3.2** Thông số kỹ thuật Arduino Mega 2560

Chip vi điều khiển	ATmega2560
Điện áp cấp nguồn	5V
Điện áp đầu vào (kiến nghị)	7-12V
Điện áp đầu vào (giới hạn)	6-20V
Số chân Analog (Input )	16
Dòng DC trên chân I/O	40 mA
Dòng DC cho chân 3.3V	50 mA

Bộ nhớ sử dụng hết 2313 bytes vào khoảng 23% bộ nhớ

Tổng số chân I/O sử dụng là 25 chân:

$$\text{dòng tiêu thụ} = 25 \times 40 = 1\text{A} \quad (3.1)$$

#### c. Khối chấp hành.

Hình ảnh thực tế khối chấp hành ở hình 3.6.



**Hình 3.6** Động cơ servo 9g

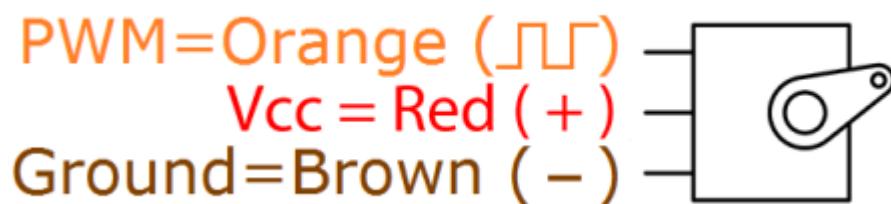
## CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

### ➤ Thông số kỹ thuật của Servo SG90 9g Micro:

- Khối lượng : 9g
- Kích thước: 22.2x11.8.32 mm
- Momen xoắn: 1.8kg/cm
- Tốc độ hoạt động: 60 độ trong 0.1 giây
- Điện áp hoạt động: 4.8V(~5V)

### ➤ Điều khiển và kết nối

Sơ đồ chân kết nối với động cơ Servo 9g biểu diễn qua hình 3.7 bên dưới.



Hình 3.7 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 khoảng  $0^\circ - 180^\circ$

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 động 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.

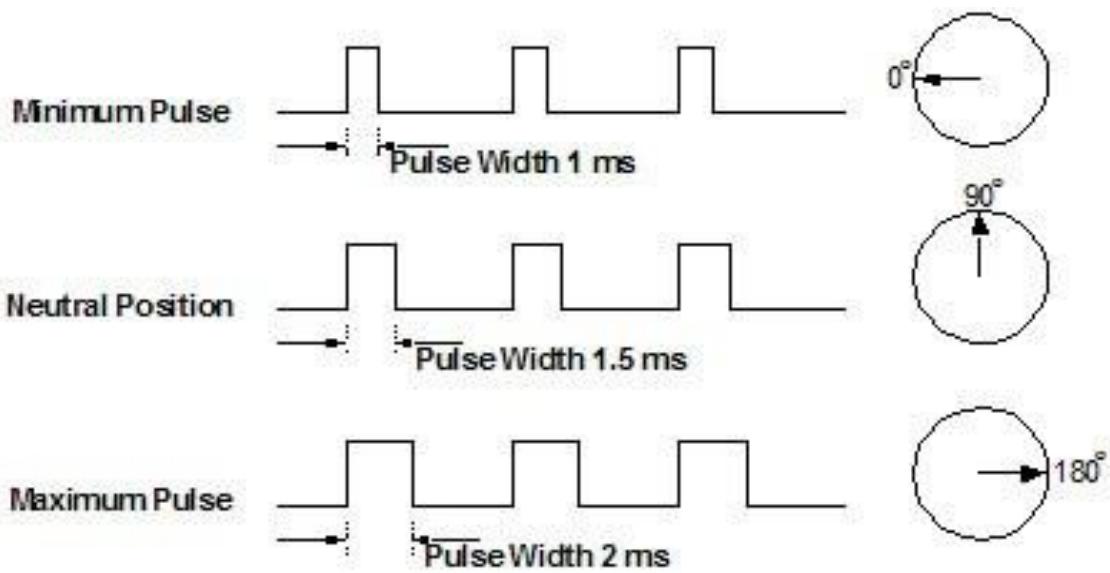
Kết nối dây màu đỏ với 5V, dây màu nâu với 0V, dây màu cam với chân phát xung (PPM) của vi điều khiển như hình 3.7. Tùy vào góc quay mà ta cấp thời gian xung ở mức cao tương ứng. Phương pháp điều chế PPM (Pulse Position Modulation) là phương pháp điều chỉnh điện áp ra tải hay nói cách khác là vị trí 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.

### CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Để dễ hiểu hơn ta có hình 3.8 sau:



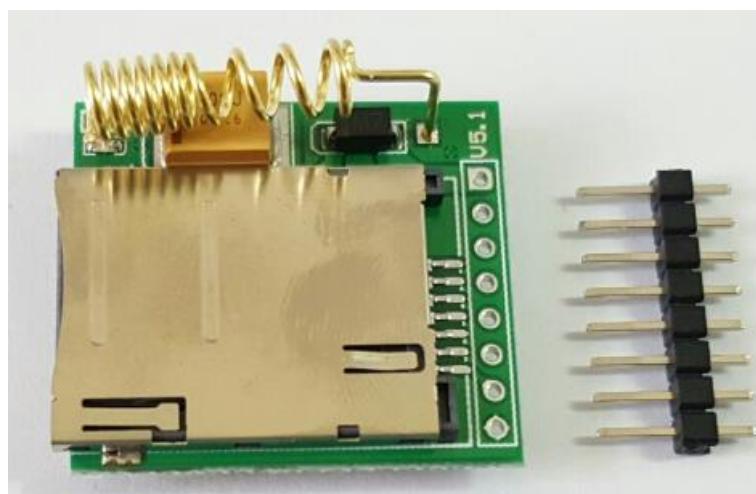
**Hình 3.8** Điều chỉnh độ rộng xung PPM

Với hình trên, thời gian 1ms mức cao, góc quay của servo là 0°; 1.5ms góc quay 90° và 2ms góc quay là 180°. Các góc khác từ 0-180° được xác định trong khoảng thời gian 1-2ms.

**Lưu ý:** Có thể ghép nhiều xung trong cùng một thời gian là 20ms để xác định vị trí góc của nhiều Servo cùng một lúc. Tối đa là 10 động cơ Servo.

#### d. Khối Sim 900A.

Module Sim900A hình 3.9 là một module GSM/GPRS cực kỳ nhỏ gọn, được SIMCOM thiết kế cho thị trường toàn cầu



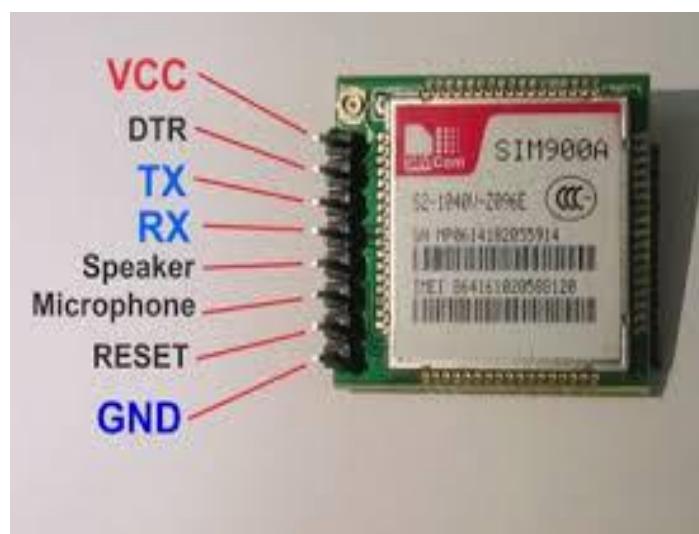
**Hình 3.9** Sim 900A thực tế

## CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

### ➤ Thông số kỹ thuật:

Phiên bản sim V5.1 ( 8 chân)

- Nguồn cấp: 4.5-5V, có thể sử dụng với nguồn dòng thấp từ 500mAh trở lên (như cổng USB, nguồn từ Board Arduino). Khuyên dùng nguồn 2A để đảm bảo hiệu suất hoạt động của SIM
- Dòng khi ở chế độ chờ: 10 mA
- Dòng khi hoạt động: 100 mA đến 2A.
- Kích thước: 2.5 cm x 3.1 cm.
- Có 8 chân VCC, DTR, TXD, RXD, Speaker, Micro Phone, Reset, GND theo hình 3.10 sau đây:



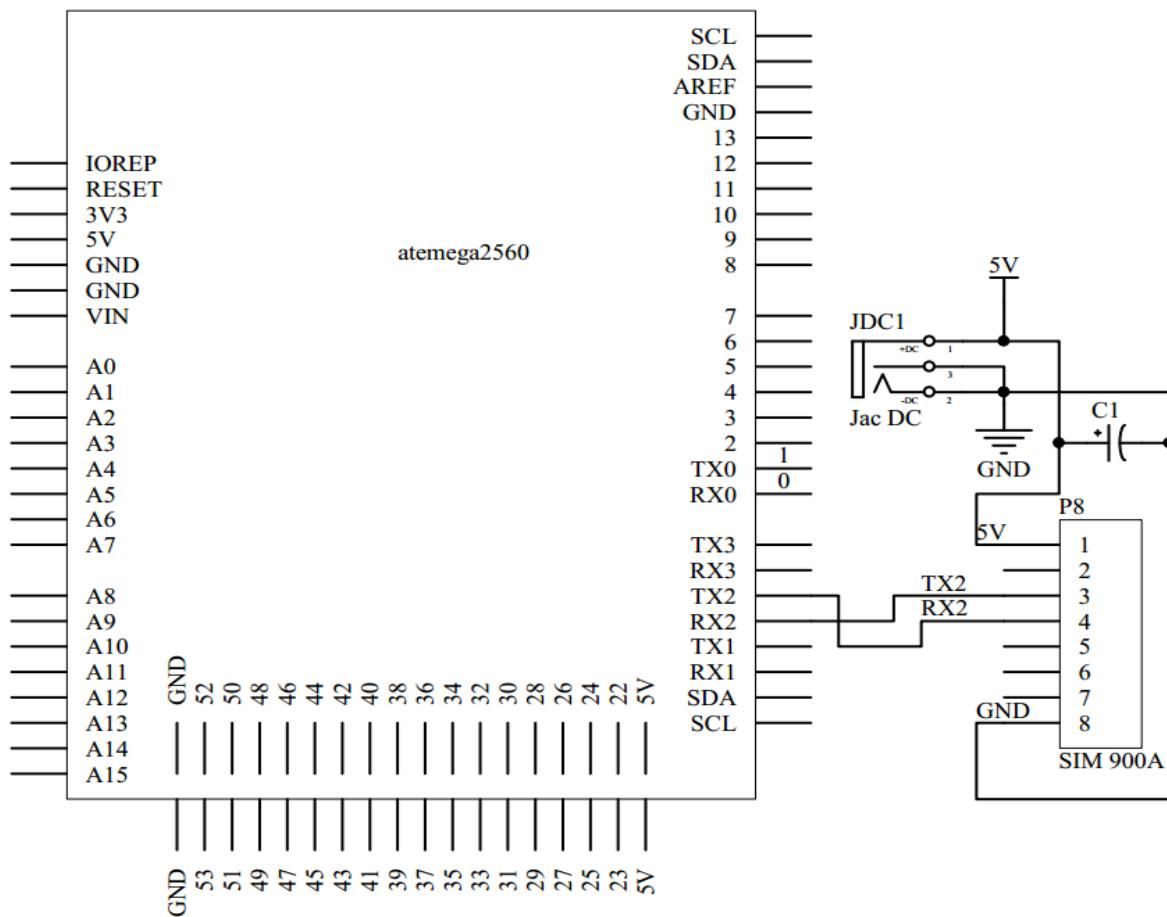
**Hình 3.10** Sơ đồ chân Sim 900A thực tế

Vì chỉ cần điều khiển đóng mở cửa và báo trạng thái nên ta chỉ dùng 4 chân là VCC, GND, TXD, RXD được kết nối với Arduino Mega như sau:

- Hai chân nguồn VCC và GND được kết nối với Adapter 5V 3A để dòng điện cho sim hoạt động ổn định lâu dài.
- TXD được nối vào chân RX2 và RXD nối vào TX2 của arduino mega để truyền nhận dữ liệu theo chuẩn UART.

Hình 3.11 thể hiện sơ đồ nguyên lý kết nối sim vào Arduino Mega và sử dụng nguồn riêng 5V 3A.

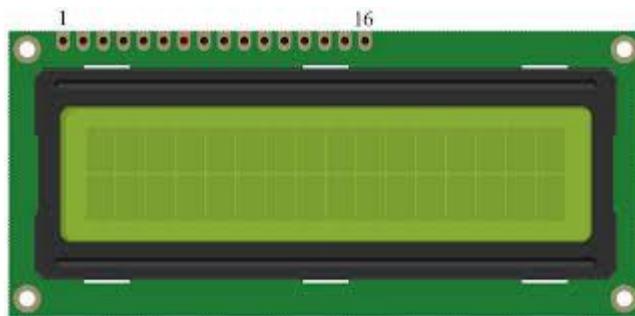
### CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ



Hình 3.11 Sơ đồ nguyên lý kết nối Sim 900A vào Arduino Mega

#### e. Khởi hiển thị

Hiển thị trạng thái làm việc của hệ thống lên màn hình LCD 16X2 như hình 3.12



Hình 3.12 LCD

- **Chân cấp nguồn:** Vss nối mass (0V), VDD (nối nguồn +5V), V<sub>0</sub> (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.
  - **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.

### **CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ**

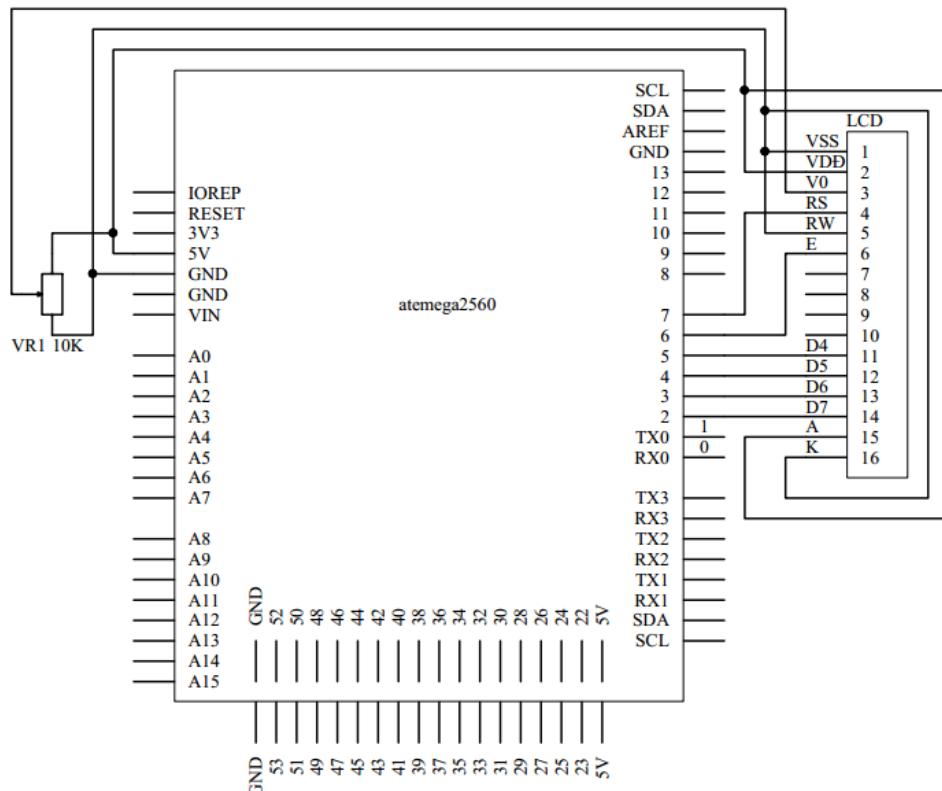
- **E:** Chân cho phép chót xung kí tự (Enable). Các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E.
  - **D0-D7:** Chân dữ liệu dùng để trao đổi dữ liệu giữa thiết bị điều khiển và LCD.
  - **A, K:** Chân điều khiển đèn nền.

Vì ta sử dụng truyền dữ liệu 4 bit nên chỉ dùng 4 chân D4 đến D7 để tiết kiệm số chân nhưng tốc độ truyền sẽ giảm lại. Thứ tự nối chân LCD đến Arduino Mega như trong bảng 3.3 sau đây:

**Bảng 3.3** kết nối chân LCD với arduino

LCD	Arduino
RS	7
E	6
D7	5
D6	4
D5	3
D4	2

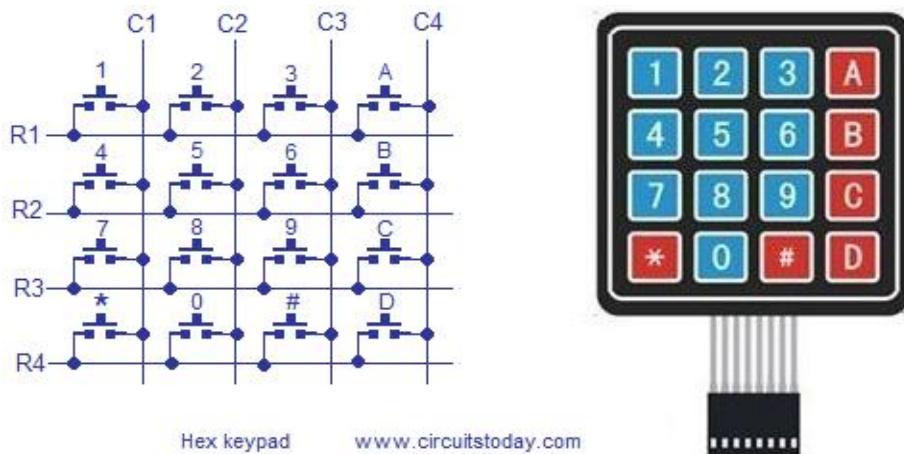
Còn hình 3.13 thể hiện sơ đồ gắp LCD và hệ thống để hiển thị thông tin hệ thống.



**Hình 3.13** Sơ đồ nguyên lý kết nối Arduino Mega với LCD

### f. Khối ngoại vi

Trong khói này có phím ma trận 4x4. Sơ đồ nối dây và hình ảnh thực tế của bàn phím ma trận 4x4 được biểu hiện trong hình 3.14 sau:



**Hình 3.14** Sơ đồ nối dây và hình ảnh thực tế của bàn phím ma trận 4x4

Keypad là một "thiết bị nhập" chứa các nút nhấn cho phép người dùng nhập các chữ số, chữ cái hoặc ký hiệu vào bộ điều khiển. Keypad không chứa tất cả bảng mã ASCII như keyboard và vì thế keypad thường được tìm thấy trong các thiết bị chuyên dụng. Các nút nhấn trên các máy tính điện tử cầm tay là một ví dụ về keypad. Số lượng nút nhấn của một keypad thay đổi phụ thuộc vào yêu cầu ứng dụng.

Gọi là keypad 4x4 vì keypad này có 16 nút nhấn được bố trí dạng ma trận 4 hàng và 4 cột. Cách bố trí ma trận hàng và cột là cách chung mà các keypad sử dụng. Cũng giống như các ma trận LED, các nút nhấn cùng hàng và cùng cột được nối với nhau, vì thế với keypad 4x4 sẽ có tổng cộng 8 ngõ ra (4 hàng và 4 cột).

- Dòng / áp hoạt động tối đa: 30 mA, 24 VDC.

### g. Khối nguồn

Board Arduino Mega 2560 sử dụng hết 25 chân (N). Dòng DC trên mỗi chân I/O ( $I_{1C}$ ) là 40mA.

$$I = I_{1C} \times N \quad (3.2)$$

$$= 40 \times 25 = 1000 \text{ mA}$$

Dòng LCD ( $I_{LCD}$ ) tiêu thụ tối đa là 2.5 mA.

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

Tổng dòng tiêu thụ cho Adapter 9V ( $I_A$ ) là:

### CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

$$\begin{aligned} I_A &= I + I_{LCD} + I_{CB} \\ &= 1000 + 2.5 + 150 = 1152.5 \text{ mA} \end{aligned} \quad (3.3)$$

Theo tính toán thì sẽ cần dòng tối thiểu là 1125.5 mA. Thực tế thì chọn Adapter 9VDC 2A như trong hình 3.15 để đảm bảo mạch hoạt động ổn định.

Cấp nguồn cho sim 900A và động cơ servo sử dụng Adapter 5VDC. Theo lý thuyết khi sim hoạt động có thể sử dụng đến 2A. Thực tế thì chọn Adapter 5VDC 3A như hình 3.16 để đảm bảo mạch hoạt động ổn định.

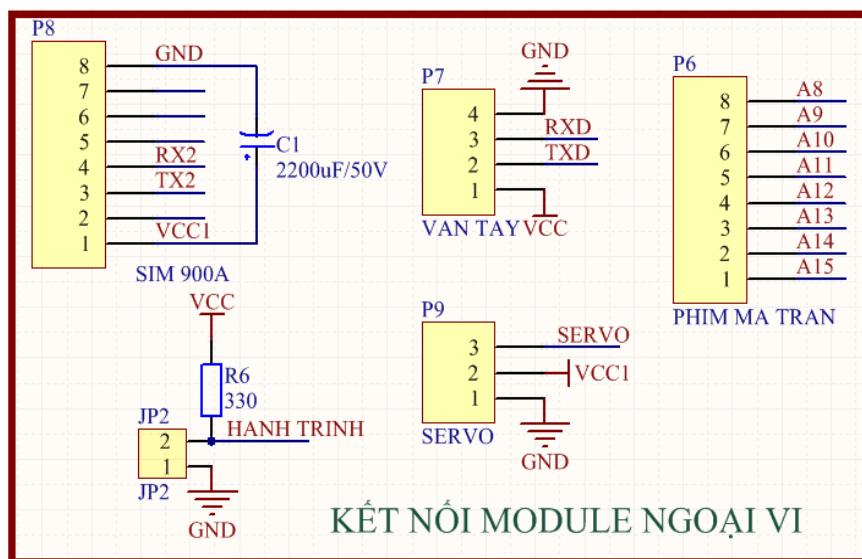


Hình 3.15 adapter 9VDC 2A



Hình 3.16 adapter 5VDC 3A

#### 3.2.3 Sơ đồ nguyên lý của toàn mạch

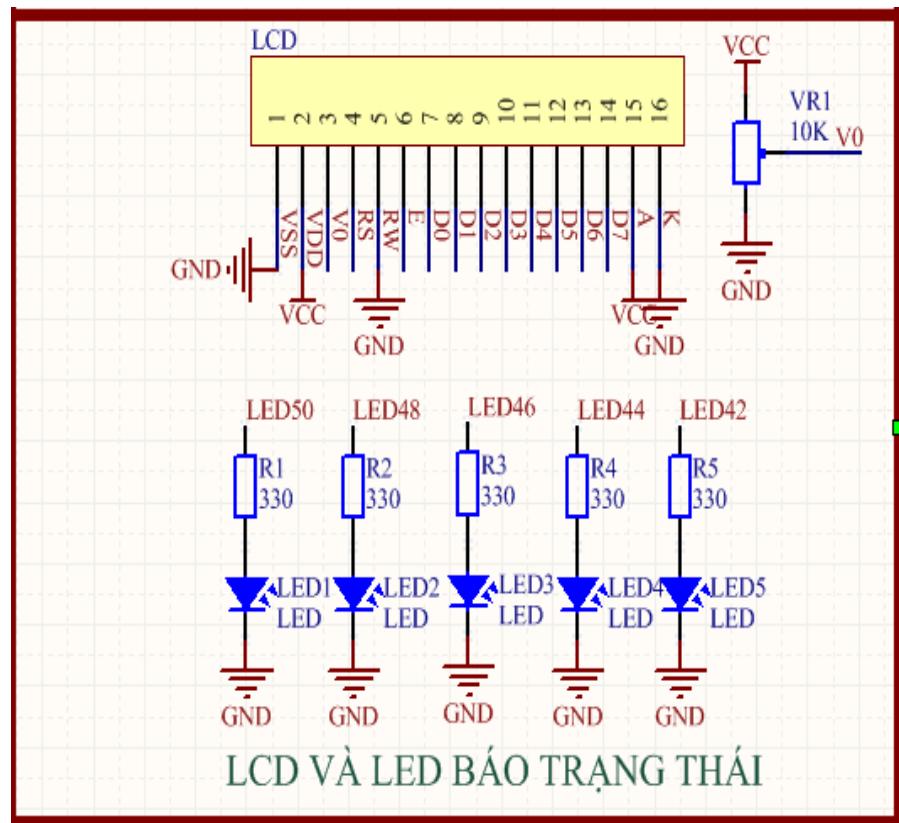


Hình 3.17 Kết nối module ngoại vi

Kết nối ngoại vi như trên hình 3.17 bao gồm việc kết nối sim 900A, cảm biến vân tay, bàn phím ma trận, động cơ servo và nút hành trình vào port thiết kế đi dây cảm vào board Arduino Mega 2560.

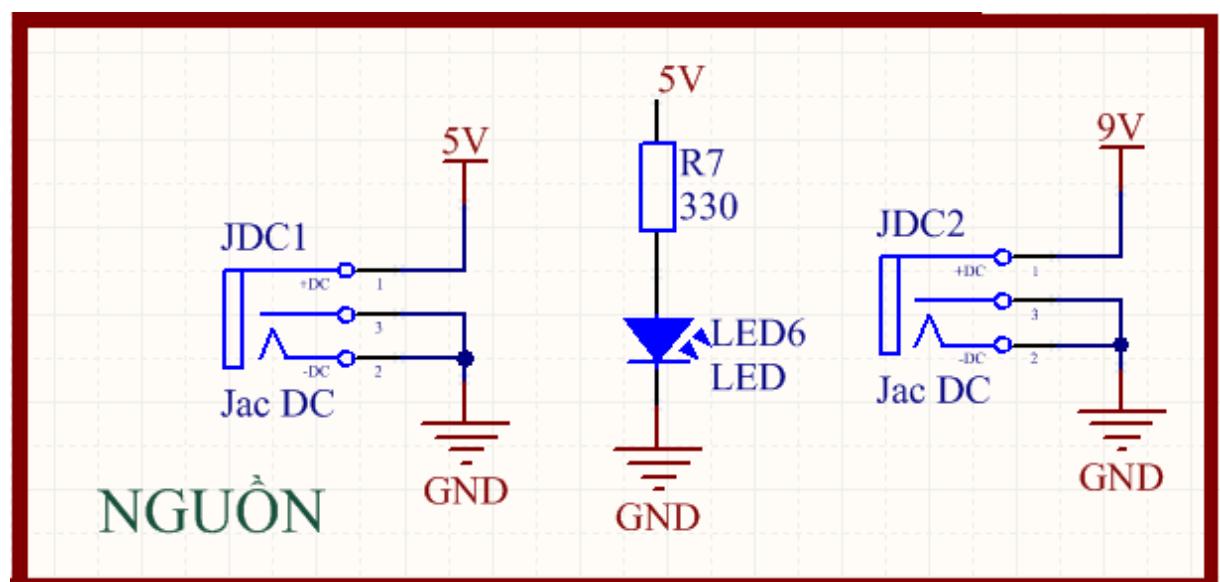
### CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Kết nối màng hình LCD để hiển thị trạng thái và các LED báo trạng thái làm việc của hệ thống được biểu diễn qua hình 3.18 sau:



Hình 3.18 LCD và LED báo trạng thái

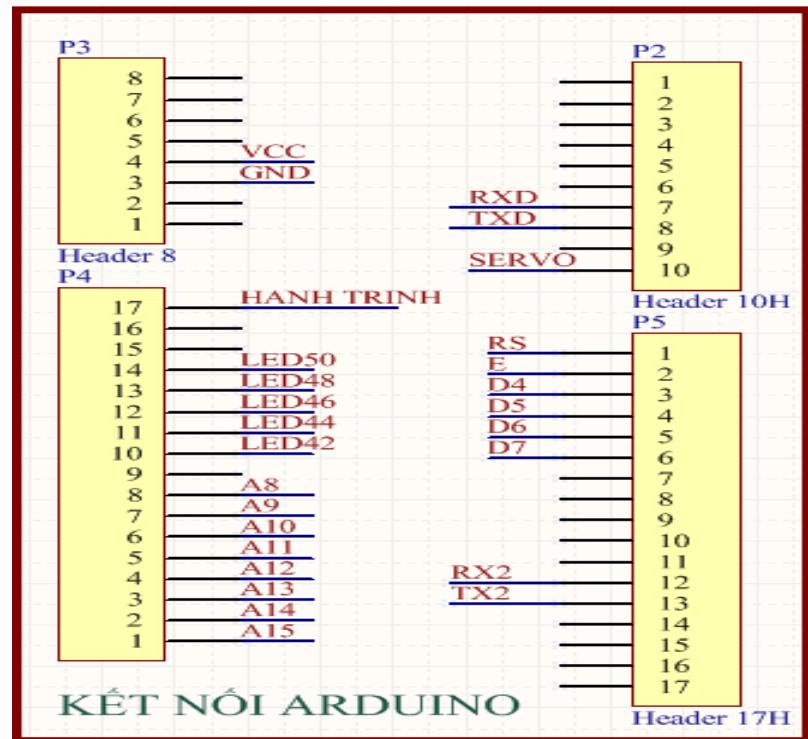
Khối nguồn trong hình 3.19 cấp nguồn 5V và 9V từ 2 Adapter cho toàn bộ hệ thống hoạt động.



Hình 3.19 Khối nguồn

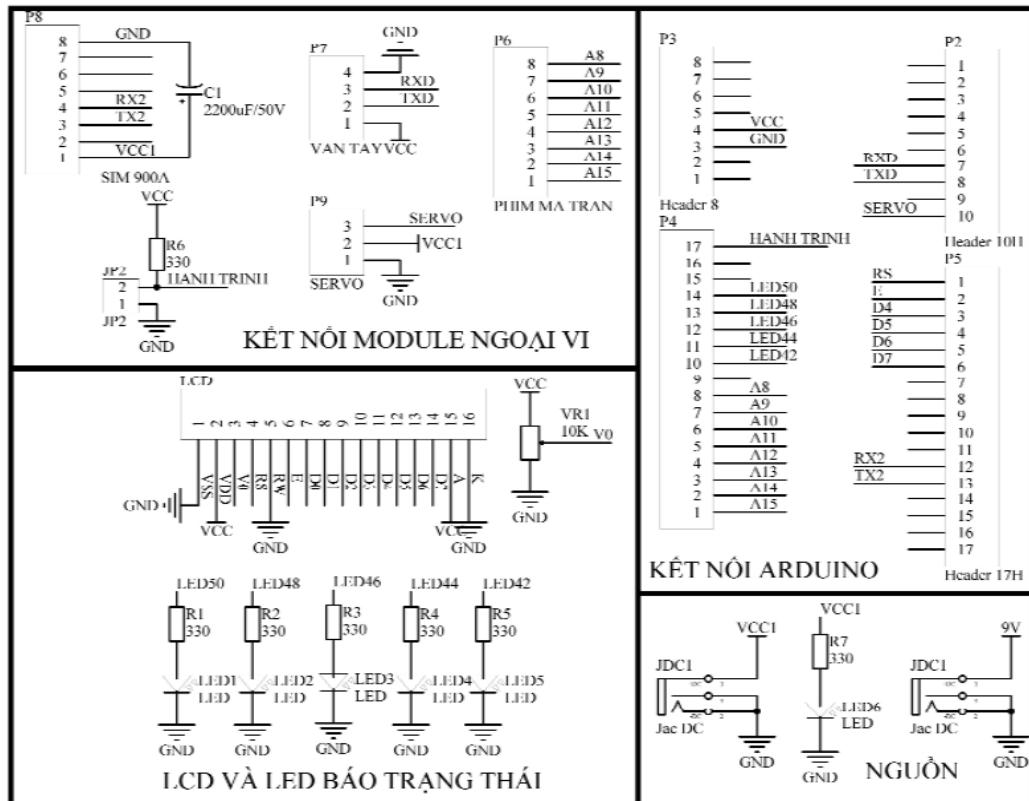
### CHƯƠNG 3. TÍNH TOÁN VÀ THIẾT KẾ

Mạch nguyên lý dùng để cắm board Arduino vào kết nối thiết bị ngoại vi ở hình 3.20



Hình 3.20 Shield kết nối Arduino Mega 2560

Còn đây là hình 3.21 là sơ đồ nguyên lý toàn mạch thể hiện tất cả các khối và kết nối các thiết bị lại với nhau rồi cắm vào Arduino Mega.



Hình 3.21 Sơ đồ nguyên lý toàn mạch

## **Chương 4. THI CÔNG HỆ THỐNG**

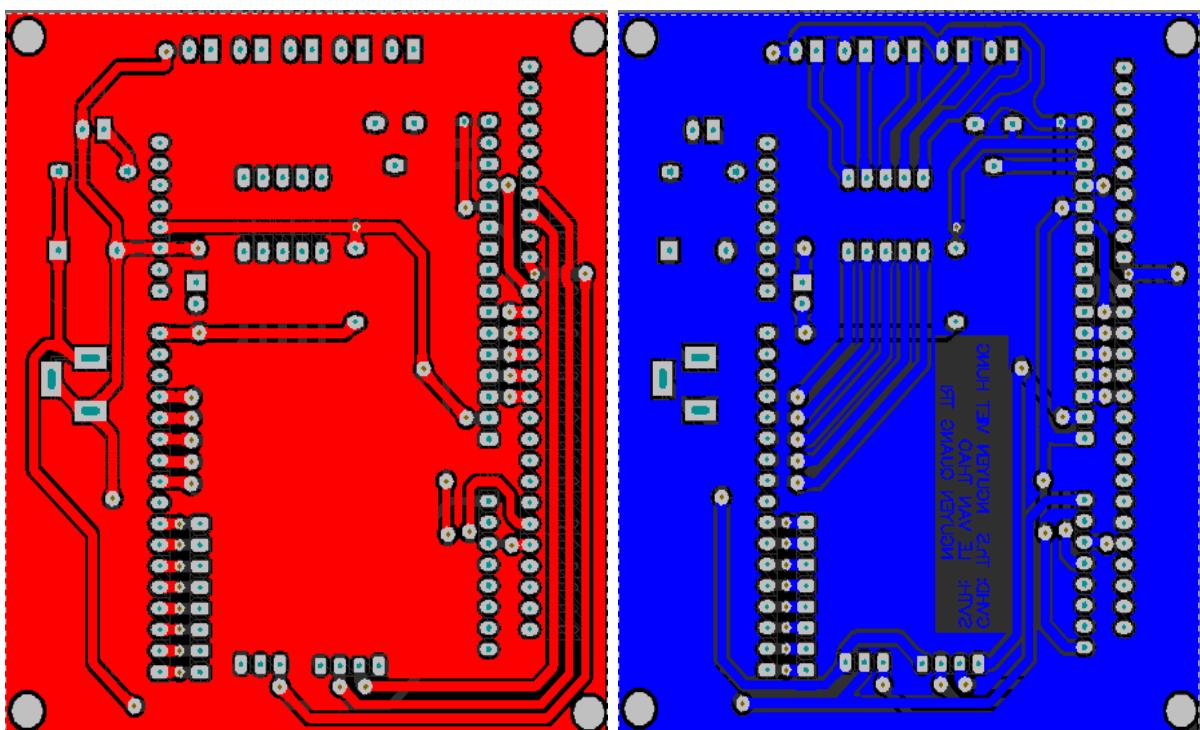
### **4.1 GIỚI THIỆU**

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**

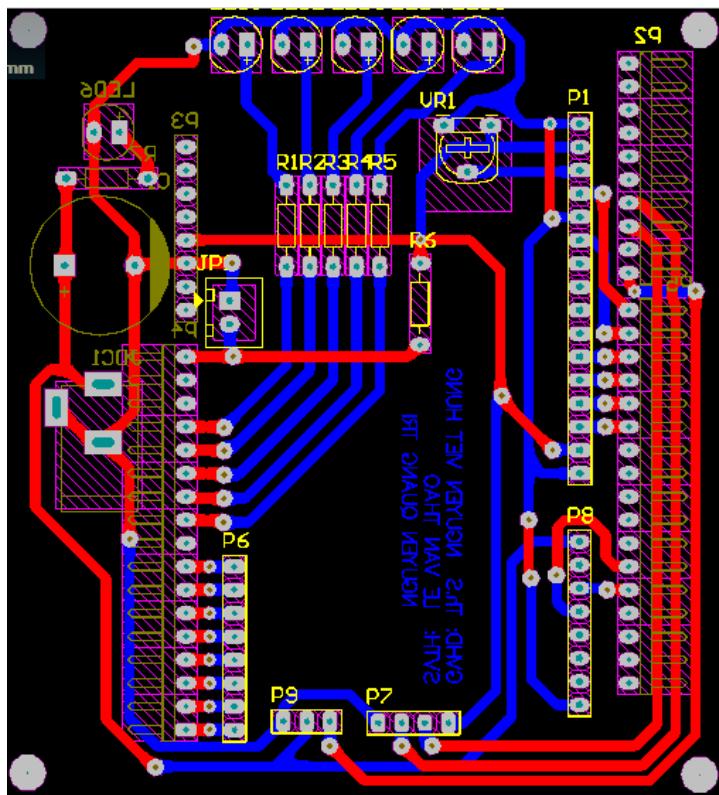
#### **4.2.1 Thi công bo mạch**

Sau khi thiết kế xong sơ đồ nguyên lý và tiến hành vẽ mạch PCB 2 lớp thủ công. Với kích thước board là 78 x 86 mm. Với lớp bottom để kết nối nguồn 5V và gắn Arduino Mega vào. Còn lớp top dùng để nối cảm biến vân tay, LCD, sim 900A, bàn phím ma trận, nút nhấn hành trình và các led đơn báo trạng thái. Hình 4.1 thể hiện lớp dây top khi đã phủ đồng, hình 4.2 thể hiện lớp dây lớp bottom, hình 4.3 thể hiện đường mạch lớp top với bottom. Mô phỏng hình dạng 3D lớp top trong hình 4.4 còn lớp bottom ở hình 4.5.

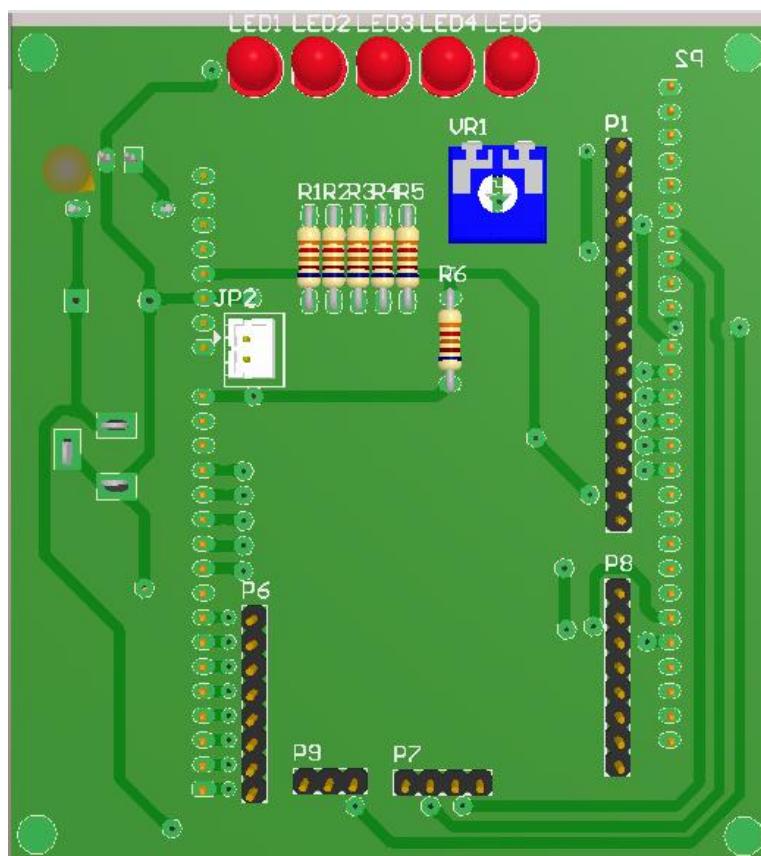


**Hình 4.1** Sơ đồ đi dây lớp top

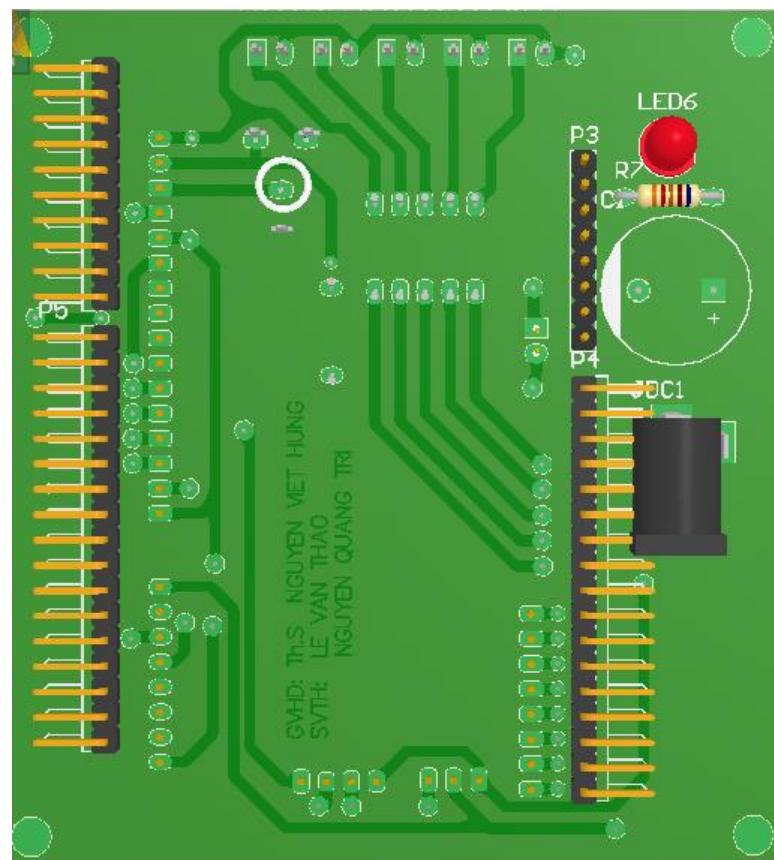
**Hình 4.2** Sơ đồ đi dây lớp bottom.



Hình 4.3 Sơ đồ đi dây lớp bottom và top chưa phủ đồng.



Hình 4.4 Hình dạng 3D lớp top.



**Hình 4.5** Hình dạng 3D lớp bottom.

Danh sách các linh kiện sử dụng của mạch được liệt kê trong bảng 4.1

**Bảng 4.1.** Danh sách các linh kiện.

STT	Tên linh kiện	Giá trị	Loại	Chú thích
1	Led	5v	5 mm	5 màu
2	Điện trở	330Ω	1/4W	
3	Tụ điện	2200uF	50V	
4	Hàng rào đức	3 hàng	Chân dài	
5	Hàng rào cáp	2 hàng	Lỗ vuông	
6	Hàng rào đức	1 hàng	vuông góc	
7	Nút nhấn hành trình			
8	Điện trở	1KΩ	1/4W	
9	Jack cáp nguồn			
10	Biến trở	10KΩ	Nút áo	
11	Board đồng	1	2 lớp	

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

### 4.2.2 Lắp ráp và kiểm tra

Quy trình lắp ráp – kiểm tra mạch :

**Bước 1:** Rửa board đồng sạch sẽ bằng thuốc rửa mạch sau khi ủi mạch và tiến hành khoan lỗ.

**Bước 2:** 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 3:** Tiến hành hàn lỗ via giữa 2 lớp top và bottom lại với nhau và tiếp tục kiểm tra có bị ngắn mạch giữa 2 chân nguồn không.

**Bước 4:** 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.

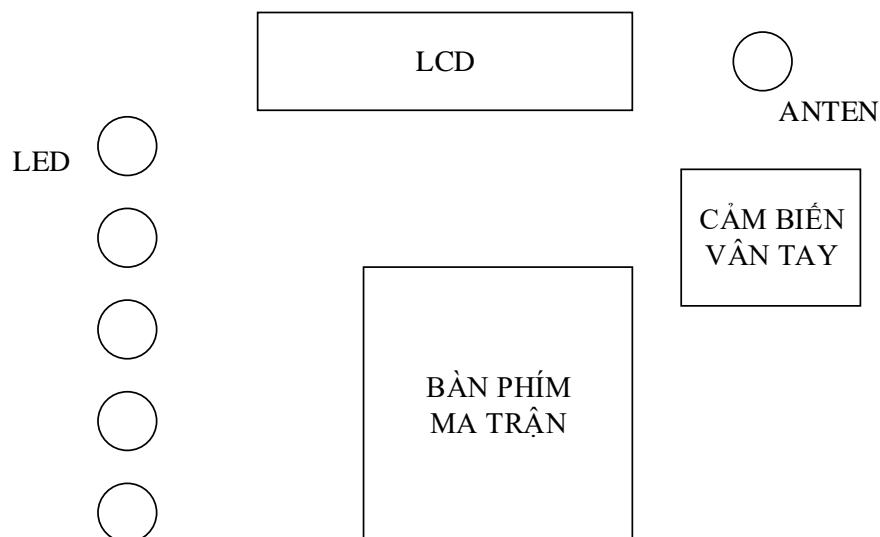
**Bước 6:** Gắn board arduino vào mạch vừa hàn xong. Đo kiểm tra từng chân từ arduino ra port đã kết nối hết chưa.

**Bước 7:** Cuối cùng cắm điện vào arduino và nguồn 5V trên mạch. Dùng đồng hồ đo áp ở ngõ vào và ngõ ra gắn với sim 900A, LCD, cảm biến vân tay. Tiếp tục chỉnh biến trở để tinh chỉnh độ tương phản LCD.

### 4.3 ĐÓNG GÓI VÀ THI CÔNG MÔ HÌNH

#### 4.3.1 Đóng gói bộ điều khiển

Sau khi kiểm tra mạch hoạt động tốt ta tiến hành đóng hộp thành mô hình cửa. Bộ điều khiển được trong 1 hình hộp chữ nhật bằng meca đen dày 5 mm với kích thước 30x30x40 cm. Sau đây hình 4.6 là sơ đồ bố trí linh kiện hiển thị ra bên ngoài mặt trước.

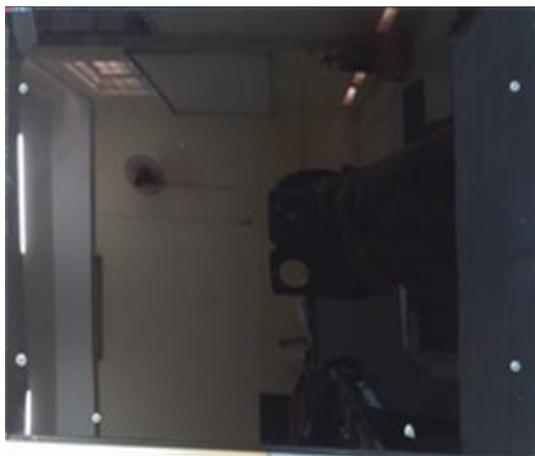


Hình 4.6 Sơ đồ bố trí linh kiện mặt trước mô hình

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

### 4.3.2 Thi công mô hình

- Khung cửa bằng meca kết nối bằng linh kiện cơ khí.

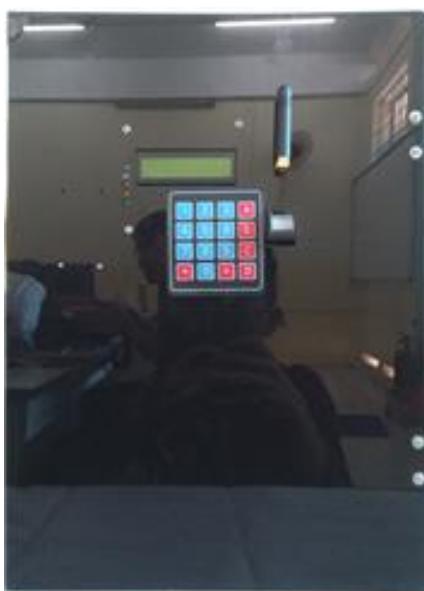


Mặt bên mô hình: 30x40 cm



Mặt trên mô hình: 30x30 cm

**Hình 4.7** Hình dạng khung cửa thực tế.



Mặt trước mô hình: 30x40 cm



**Hình 4.9** Chi tiết cơ khí ráp khung.

Khung cửa được làm từ meca kết nối bởi các chi tiết cơ khí như hình 4.9 bao gồm 3 mặt bên và 2 đáy như hình 4.7 với kích thước 30x40cm và 30x30cm và mặt trước cánh cửa hình 4.8 kích thước 30x40cm để tạo thành một hình hộp chữ nhật có kích thước 30x30x40 cm.

Thể tích mô hình tính theo công thức số (4.1).

$$V = A \times B \times C = 30 \times 30 \times 40 = 0.036 \text{ m}^3 \quad (4.1)$$

Với A, B C lần lượt là chiều dài, rộng, cao của mô hình cửa

### 4.4 LẬP TRÌNH HỆ THỐNG

#### 4.4.1 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 hoặc từ điện thoại thông qua tin nhắn hay cuộc gọi, cho phép nạp thêm vân tay, xóa bớt số lượng vân tay.

Khi cấp điện vào hệ thống, khởi động arduino, sim 900a, cảm biến vân tay và LCD. Sau khi khởi động xong mặc định sẽ vào chế độ đóng mở cửa, muốn đổi chế độ thì ta nhấn phím B tương ứng với chức năng nạp vân tay, phím C chức năng xóa vân tay, phím A chức năng đóng mở cửa.

#### ➤ LUU ĐO CHÍNH

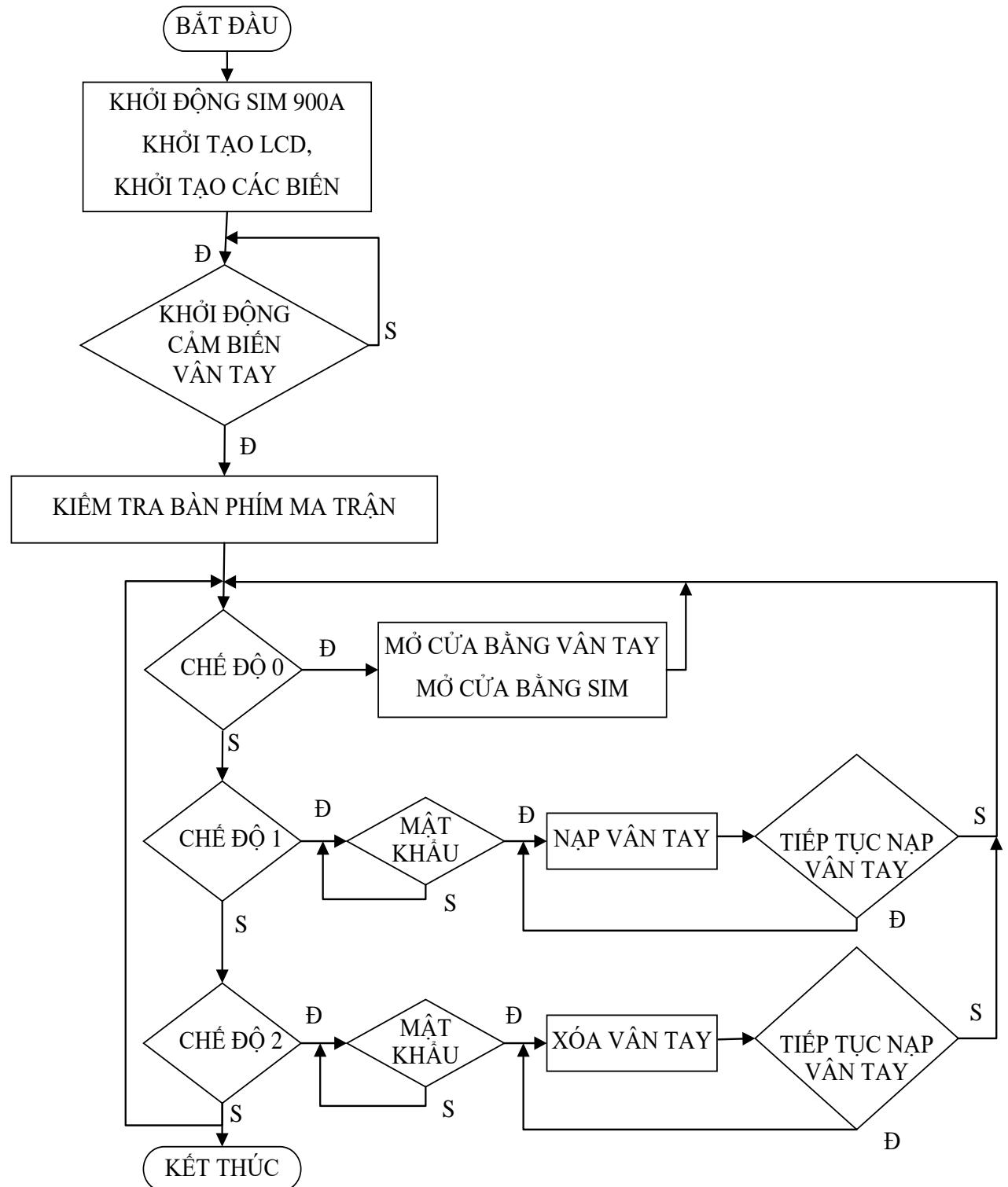
Quy trình làm việc của hệ thống như thể hiện trong sơ đồ hình 4.10. Chương trình bắt đầu vào khởi động sim 900A, LCD, các biến. Sau đó kiểm tra có cảm biến vân tay kết nối chưa. Nếu chưa kết nối thì chương trình sẽ dừng đợi đến khi nào có kết nối. Khi phát hiện có kết nối vân tay thì sẽ khởi động chương trình ngắn 100ms kiểm tra bàn phím ma trận 1 lần, để xem chương trình đang hoạt động ở chế độ nào. Mặc định chương trình vào chế độ 0. Nếu phím A được nhấn thì sẽ vào chế độ 0, nếu phím B được nhấn thì sẽ vào chế độ 1, nếu phím C được nhấn thì sẽ vào chế độ 2.

Ở chế độ 0 thì cho phép đóng mở cửa bằng dấu vân tay hay từ điện thoại thông qua tin nhắn hoặc cuộc gọi. Cho phép kiểm tra trạng thái cửa hoặc tài khoản thông báo đến người dùng khi có yêu cầu.

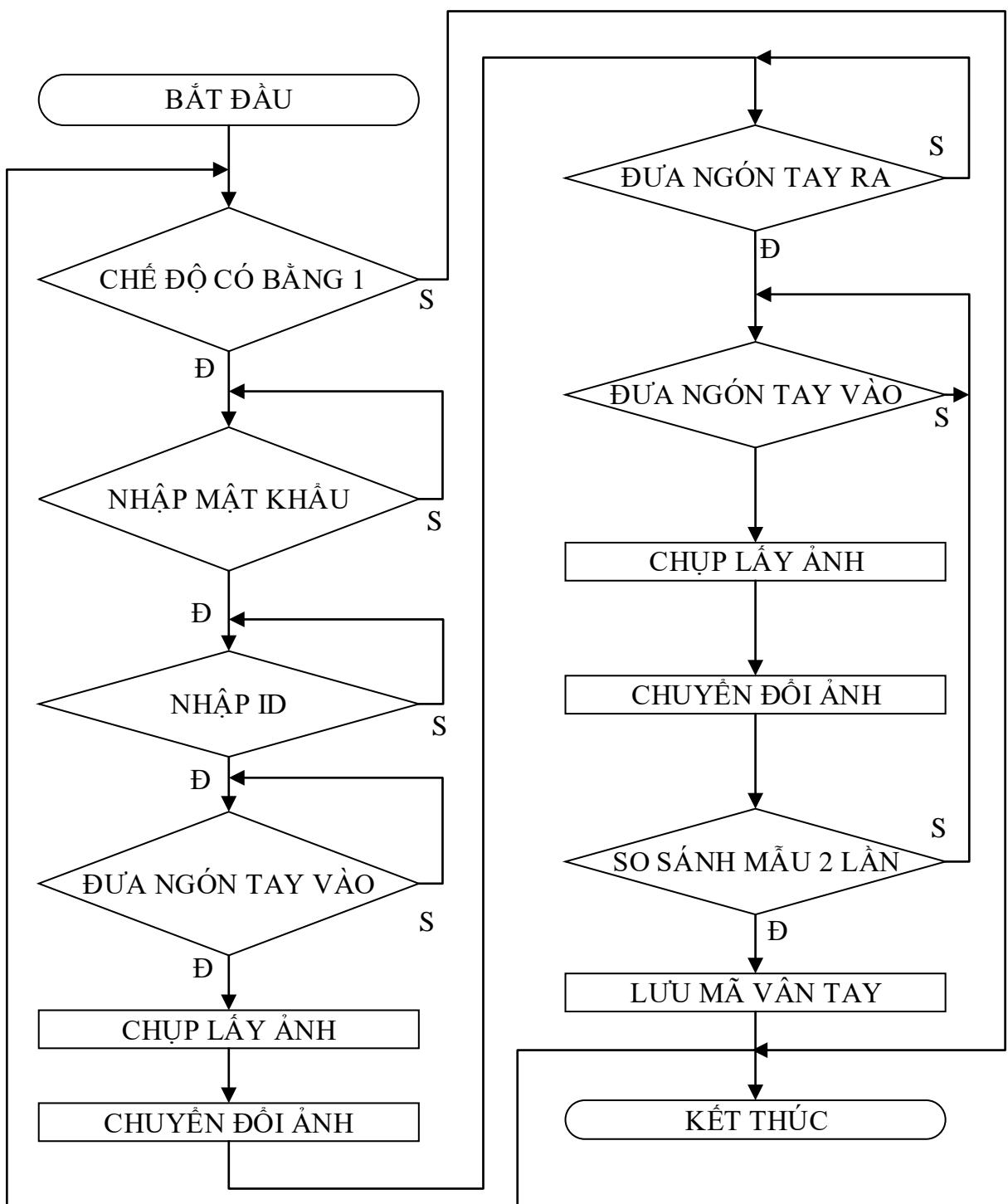
Ở chế độ 1 thì cho phép nạp thêm dấu vân tay cho người dùng mới. Để nạp thêm người dùng thì cần phải nhập mật khẩu chính xác trước để đảm bảo chủ thiết bị mới có thể cấp thêm quyền sử dụng cho người khác. Sau đó chúng ta tiến hành nạp vân tay lần lượt các ID muốn nạp. Khi nào xong ta chọn phím A hoặc phím B để chuyển chế độ.

Ở chế độ 2 thì cho phép xóa dấu vân tay người dùng cũ. Để xóa người dùng thì cần phải nhập mật khẩu chính xác trước để đảm bảo chủ thiết bị mới có quyền xóa dấu vân tay cũ. Sau đó chúng ta tiến hành xóa lần lượt các ID muốn xóa. Khi nào xong ta chọn phím A hoặc phím B để chuyển chế độ.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.10 Lưu đồ chính làm việc của hệ thống.



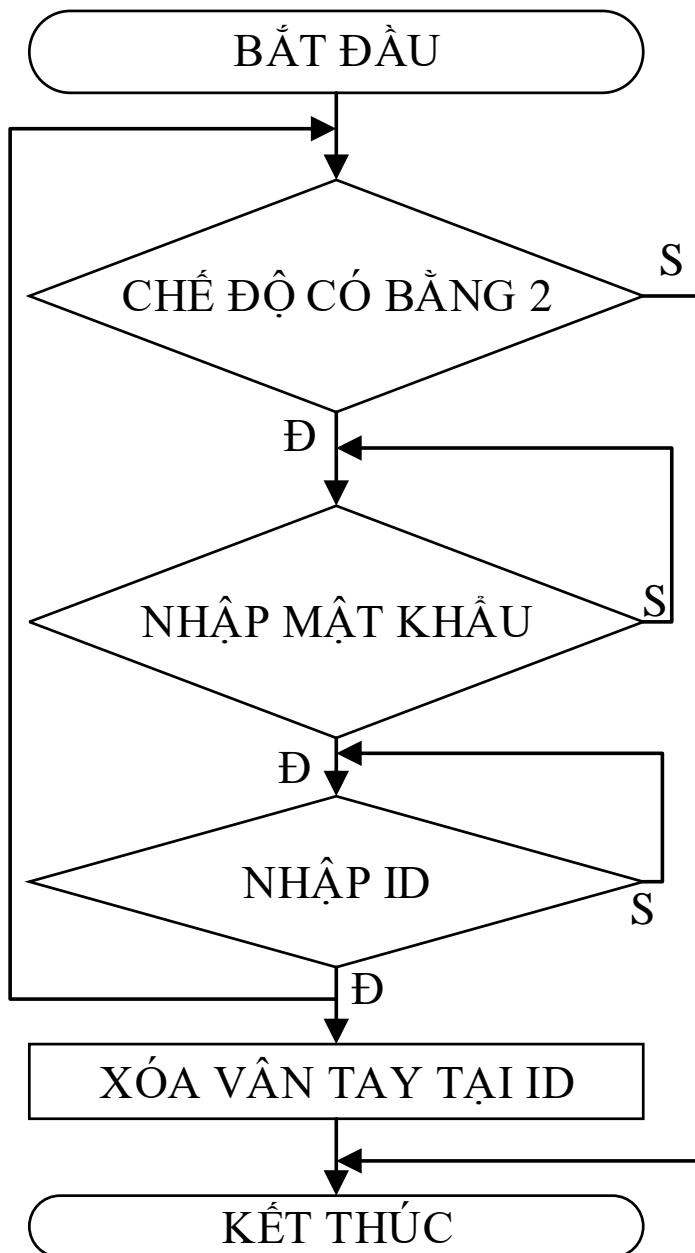
**Hình 4.11** Lưu đồ chế độ nạp vân tay

➤ **CHẾ ĐỘ NẠP VÂN TAY**

Quy trình lấy mẫu vân tay thêm người dùng thực hiện theo lưu đồ hình 4.11. Chương trình ngắt kiểm tra nếu đang ở chế độ 1 thì nhập mật khẩu cho đến khi nào đúng để vào được chương trình nạp dấu vân tay. Nhập ID vị trí lưu dấu vân tay sau đó ta đưa

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

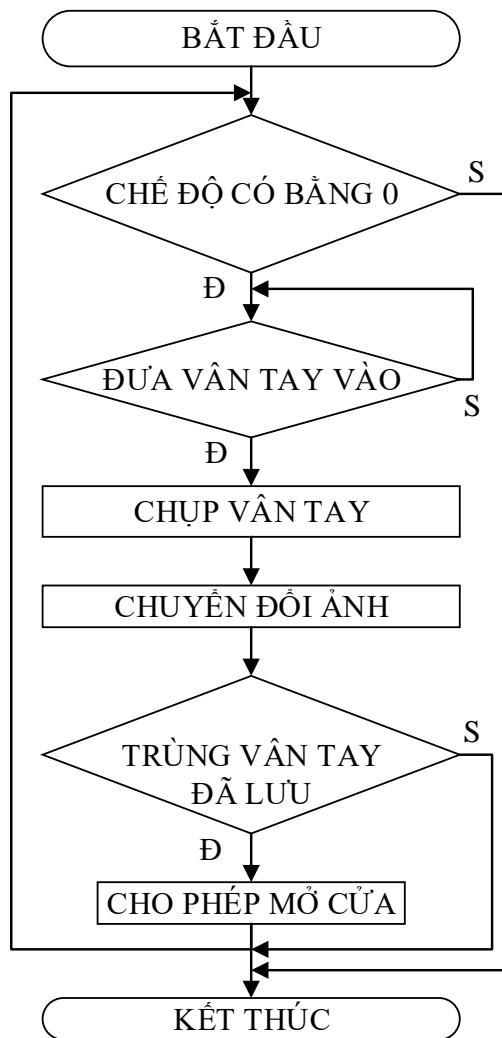
vân tay cần lưu vào cảm biến để tiến hành lấy mẫu lần một và tương tự lấy mẫu lần hai. Nếu hai lần mà không giống nhau thì ta sẽ lấy mẫu lại lần hai cho đến khi giống thì ta lưu dấu vân tay lại. Tiếp tục lưu vân tay với những mẫu khác tương tự như trên cho đến khi kết thúc việc lấy mẫu thì ta thoát khỏi chế độ 1 bằng việc ấn phím A hoặc phím C.



Hình 4.12 Lưu đồ chế độ xóa vân tay

### ➤ XÓA VÂN TAY

Quy trình xóa vân tay người dùng như lưu đồ hình 4.12. Chương trình ngắt kiểm tra nếu đang ở chế độ 2 thì nhập mật khẩu cho đến khi nào đúng để vào được chương trình xóa dấu vân tay. Nhập ID cần xoá thực hiện cho đến khi kết thúc quá trình xóa thì ta chuyển chế độ sang 0 hoặc 1 để thoát chương trình.

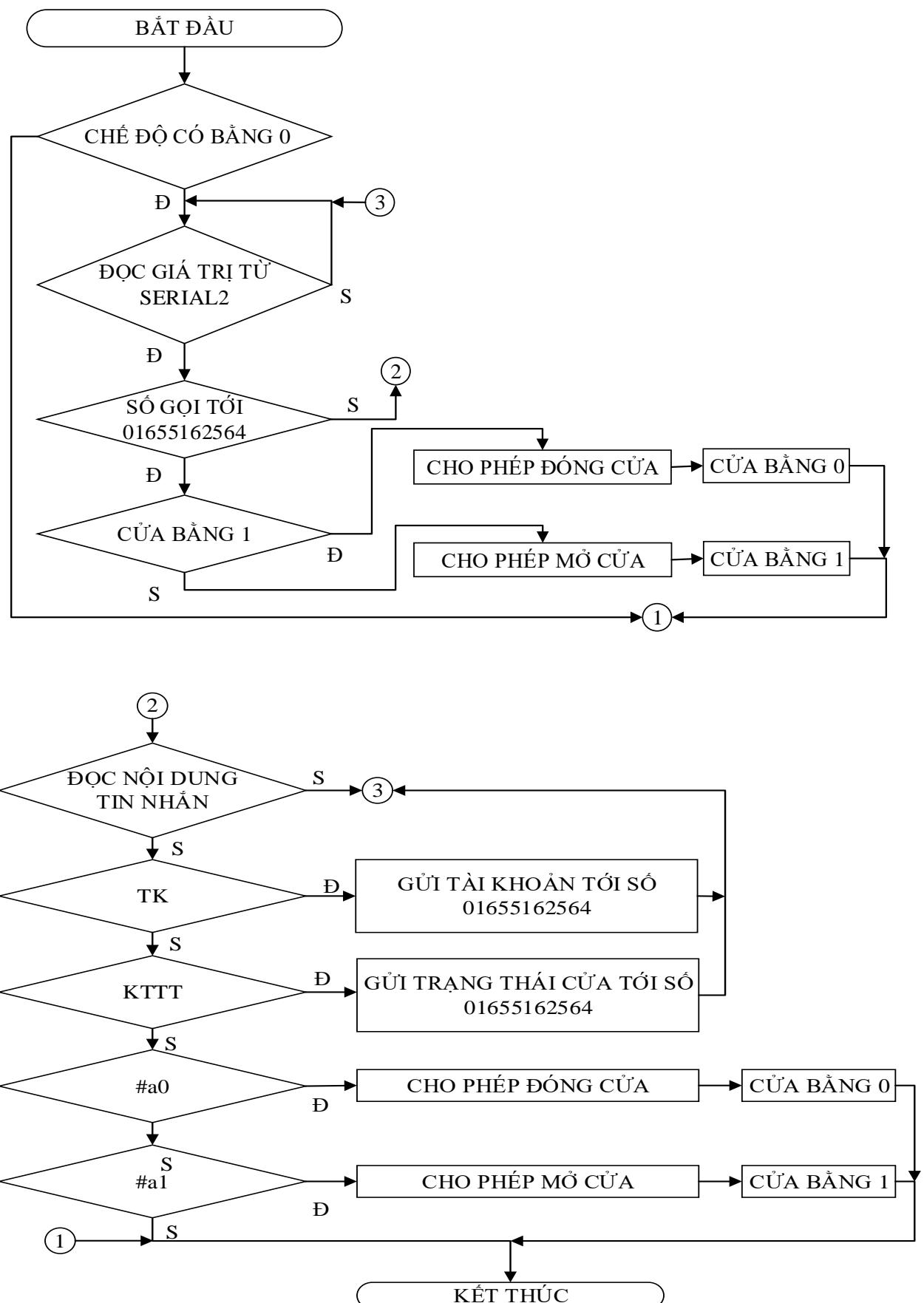


**Hình 4.13** Lưu đồ chế độ mở cửa bằng vân tay

### ➤ MỞ CỬA BẰNG VÂN TAY

Quá trình mở cửa bằng dấu vân tay thực hiện theo lưu đồ hình 4.13 phía trên. Chương trình ngắt kiểm tra nếu đang ở chế độ 0 thì ta đưa vân tay vào. Nếu vân tay được quét so sánh trùng với vân tay đã được lưu trước đó thì cho phép mở cửa, nếu không trùng thì không cho phép mở cửa.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

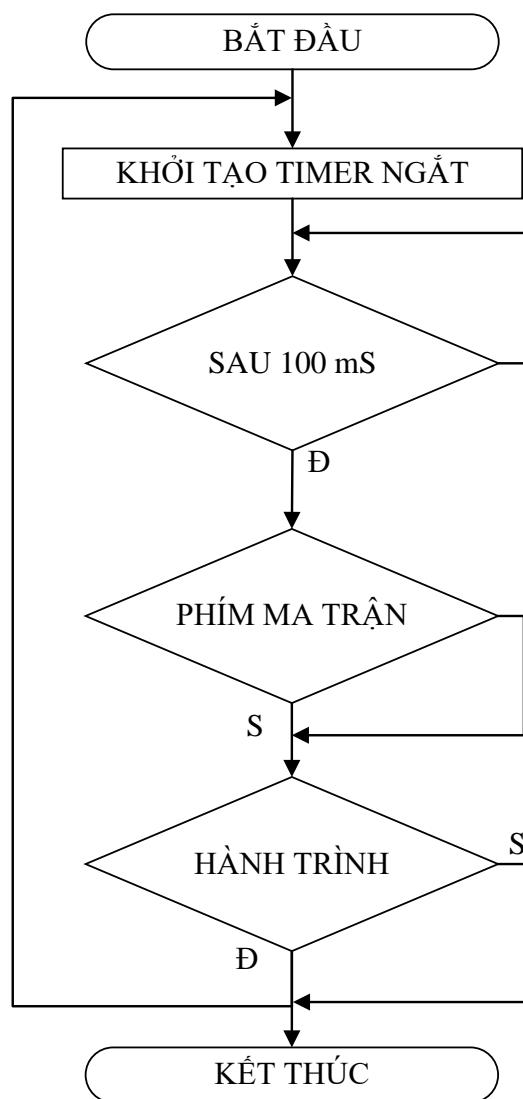


Hình 4.14 Lưu đồ chế độ mở cửa bằng điện thoại

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

### ➤ ĐIỆN THOẠI

Quá trình điều khiển đóng mở cửa và kiểm soát cửa như sơ đồ hình 4.14 trên hoạt động như sau: Chương trình ngắt kiểm tra nếu đang ở chế độ 0 thì cho cổng serial2 đọc dữ liệu. Nếu dữ liệu thu được trùng với cho điện thoại 01655162564 thì cho phép đóng mở cửa khi cửa đang đóng. Nếu dữ liệu thu được là #a0 thì đóng cửa, #a1 thì mở cửa, kttt thì sim sẽ gửi báo trạng thái cửa lại cho số 01655162564 và tk thì sim gửi tài khoản trong sim đến số 01655162564 được cài mặc định trong chương trình.



Hình 4.15 Lưu đồ chương trình ngắt

### ➤ NGẮT

Chương trình ngắt hoạt động theo lưu đồ hình 4.15 sử dụng bộ chia 64. Thời gian ngắt là 100ms kiểm tra hàm quét phím ma trận và nút nhấn hành trình một lần thực hiện liên tục song song cùng với vòng lặp chính của chương trình.

### 4.4.2 Phần mềm lập trình cho vi điều khiển

#### Giới thiệu phần mềm lập trình Arduino IDE

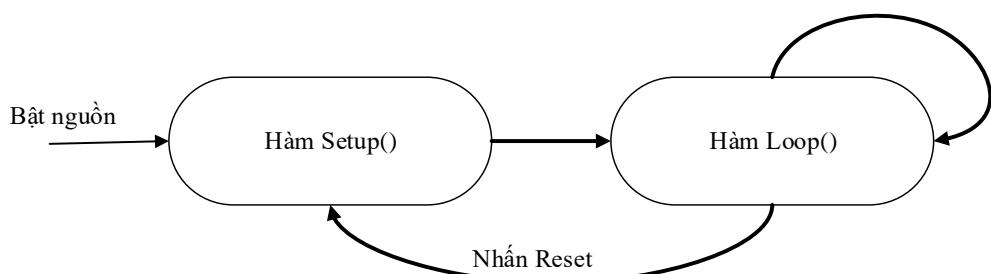
Môi trường phát triển tích hợp Arduino IDE là một ứng dụng đa nền tảng được viết bằng Java, và được dẫn xuất từ IDE cho ngôn ngữ lập trình xử lý và các dự án lắp ráp. Do có tính chất mã nguồn mở nên môi trường lập trình này hoàn toàn miễn phí và có thể mở rộng thêm bởi người dùng có kinh nghiệm.

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.

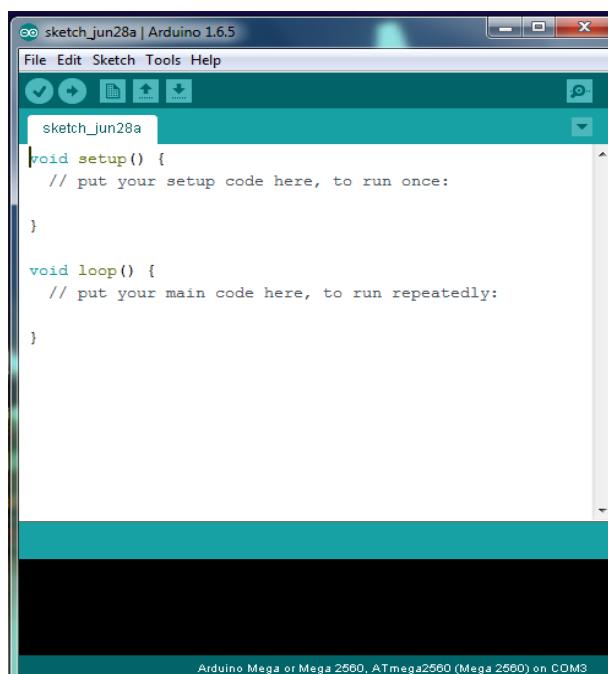
**loop()**: hàm được gọi lặp lại liên tục cho đến khi bo mạch được tắt.

Chu trình đó có thể mô tả trong hình 4.16 dưới đây:



**Hình 4.16** Quy trình làm việc của arduino

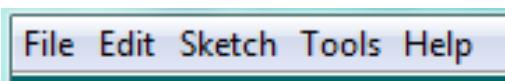
Arduino IDE hình 4.17 là nơi để soạn thảo code, kiểm ra lỗi và upload code.



**Hình 4.17** Giao diện lập trình arduino.

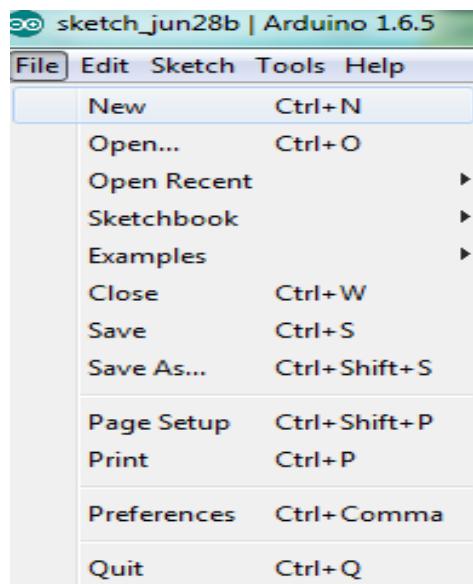
## CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Arduino IDE Menu:



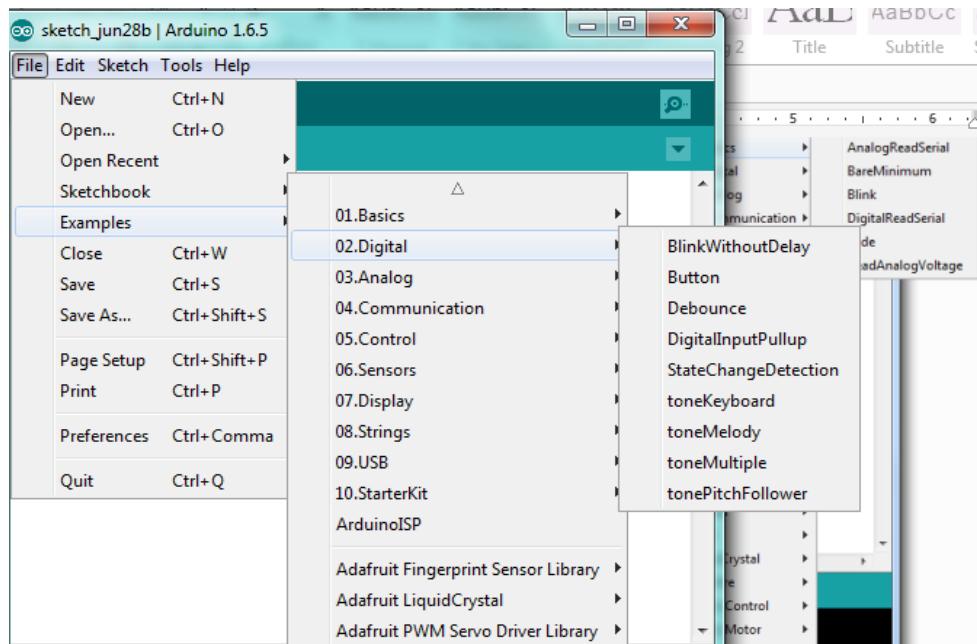
Hình 4.18 Giao diện menu arduino IDE

File menu:



Hình 4.19 Giao diện file menu arduino IDE

Trong file menu cần quan tâm tới mục Examples, đây là nơi chứa các chương trình mẫu đơn giản như: cách sử dụng các chân digital, analog, sensor,...



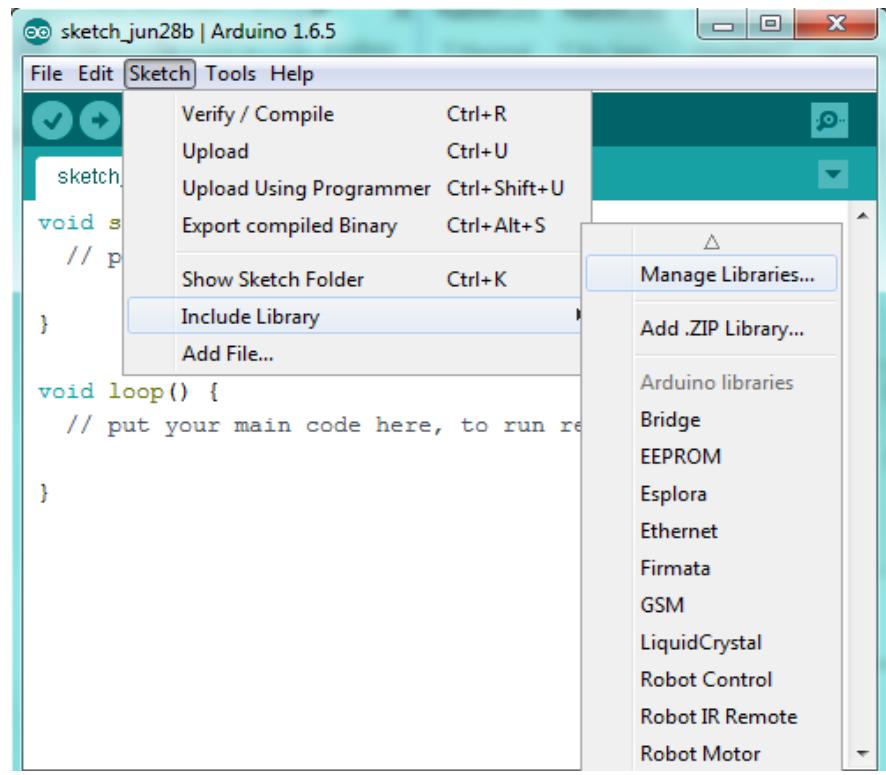
Hình 4.20 Giao diện Examples menu arduino IDE

Sketch menu:

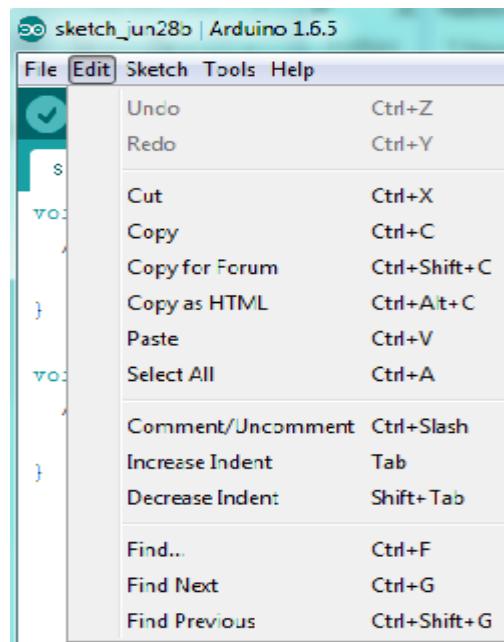
- Verify/ Compile: chức năng kiểm tra lỗi code.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

- Show Sketch Folder: hiển thị nơi code được lưu.
- Add File: thêm vào một Tap code mới.
- Import Library: thêm thư viện từ bên ngoài cho IDE.



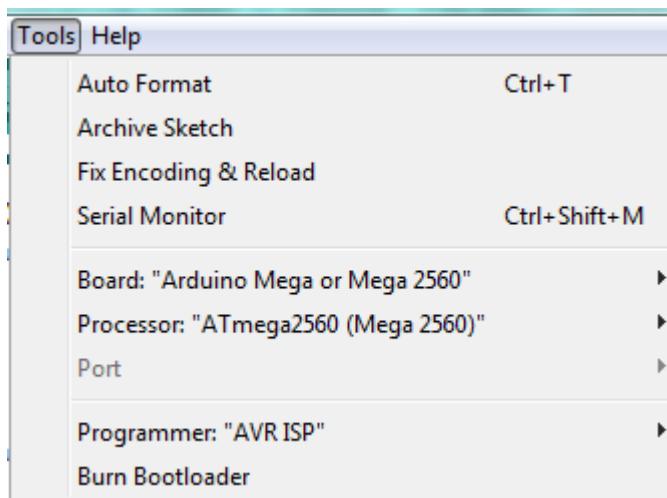
Hình 4.21 Giao diện Sketch Menu Arduino IDE



Hình 4.22 Giao diện edit menu arduino IDE

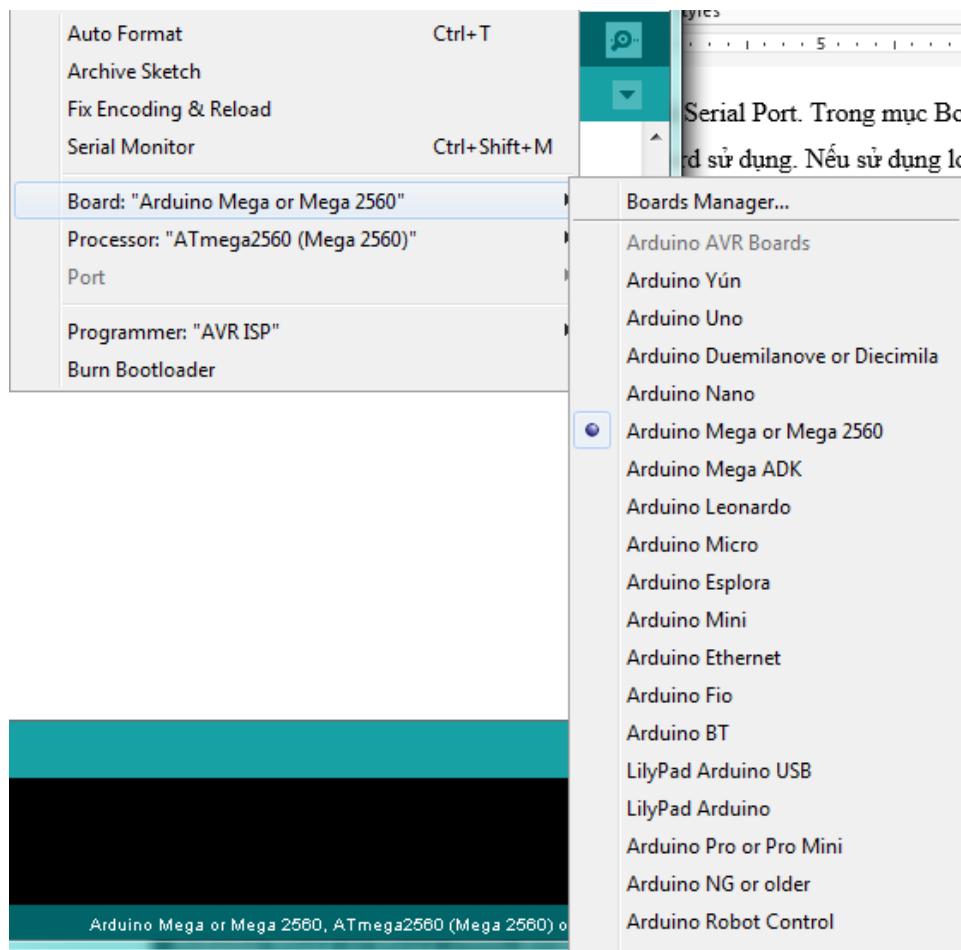
## CHƯƠNG 4. THI CÔNG HỆ THỐNG

### Tool menu:



Hình 4.23 Giao diện Tool Menu Arduino IDE

Trong Tool menu ta quan tâm các mục Board và Serial Port. Trong mục Board, cần phải lựa chọn board mạch cho phù hợp với loại board sử dụng. Nếu sử dụng loại board khác thì phải chọn đúng loại board mà mình đang có, nếu sai thì khi Upload chương trình vào chip sẽ bị lỗi. Nếu là Arduino Mega 2560 thì phải chọn như hình 4.24:



Hình 4.24 Board Arduino sử dụng

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

Serial Port: đây là nơi lựa chọn cổng Com của Arduino. Khi chúng ta cài đặt driver thì máy tính sẽ hiện thông báo tên cổng Com của Arduino là bao nhiêu, ta chỉ việc vào Serial Port chọn đúng cổng Com để nạp code, nếu chọn sai thì không thể nạp code cho Arduino được.



Hình 4.25 Hiển thị Board và Serial Port đã kết nối

Arduino Toolbar có một số button và chức năng của chúng như sau:



Hình 4.26 Arduino Toolbar

- Verify (1): kiểm tra code có lỗi hay không.
- Upload (2): nạp code đang soạn thảo vào Arduino.
- New, Open, Save (3): Tạo mới, mở và lưu sketch.
- Serial Monitor (4): Đây là màn hình hiển thị dữ liệu từ Arduino gửi lên máy tính.

Những hình ảnh từ 4.17 đến 4.26 mô tả những tác vụ trên phần mềm lập trình cho arduino cho người dùng hiểu được các chức năng cũng như thông tin về arduino.

### 4.5 KẾT QUẢ MÔ PHỎNG

Dạng xung PPM điều khiển Servo để đóng mở cửa như sau:



Hình 4.27 Xung PPM mở cửa

## CHƯƠNG 4. THI CÔNG HỆ THỐNG



Hình 4.28 Xung PPM đóng cửa

Chu kỳ tạo xung PPM trong arduino là 20 ms (50HZ). Theo lý thuyết thì với thời gian tồn tại mức cao (T) 1ms thì góc quay của servo ( $\theta$ ) là  $0^\circ$ , 1.5 ms thì góc quay là  $90^\circ$ , còn 2 ms là  $180^\circ$ . Vậy ta có được 0.1 ms tương ứng với  $18^\circ$ .

Từ hình 4.27 ta thấy thời gian tồn tại mức 1 (T) khoảng 1.3 ms, còn hình 4.28 là 1.8 ms. Từ đây ta tính được góc quay của servo như công thức (4.2) và (4.3).

$$\text{PPM (mở cửa)} = T \times \theta = 3 \times 18 = 54^\circ \quad (4.2)$$

$$\text{PPM (đóng cửa)} = T \times \theta = 8 \times 18 = 144^\circ \quad (4.3)$$

### 4.6 VIẾT TÀI LIỆU HƯỚNG DẪN SỬ DỤNG, THAO TÁC

#### 4.6.1 Viết tài liệu hướng dẫn sử dụng

**Bước 1:** Cáp nguồn cho hệ thống, hệ thống sử dụng nguồn là 220V AC, đã có 2 adapter 5V và 9V cắm sẵn vào ổ điện.

**Bước 2:** Đợi hệ thống khởi động xong sẽ sáng đèn số 1 và 3. Khi đèn 3 sáng là ở chế độ đóng mở cửa. Khách hàng đưa vân tay vào để mở cửa hoặc sử dụng điện thoại để mở cửa hoặc kiểm tra trạng thái cửa.

**Bước 3:** Nếu muốn nạp thêm người dùng thì nhấn phím B sau đó tiến hành nhập mật khẩu là 1234 sau đó nhấn phím \* để đăng nhập để vào chế độ nạp thêm người dùng. Nhập ID cần lưu từ bàn phím sau đó nhấn phím \*, đưa vân tay vào cảm biến khi nào đèn xanh trong cảm biến tắt ta rút vân tay ra để đèn xanh sáng lại ta thực hiện đưa vân tay vào lại để xác nhận.

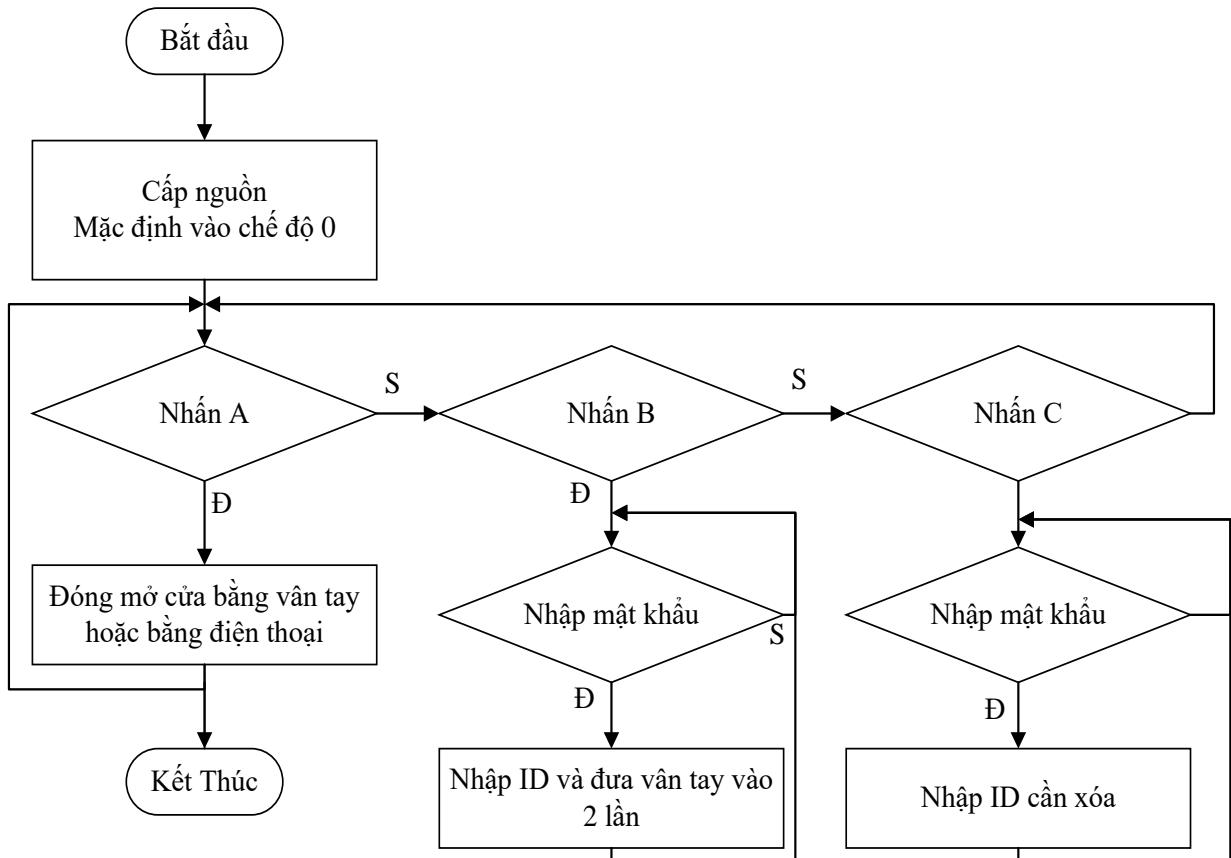
**Bước 4:** Nếu muốn xóa người dùng thì nhấn phím C sau đó tiến hành nhập mật khẩu là 1234 sau đó nhấn phím \* để đăng nhập để vào chế độ xóa người dùng. Nhập ID cần lưu từ bàn phím sau đó nhấn phím \* để xóa ID đó.

## CHƯƠNG 4. THI CÔNG HỆ THỐNG

**Bước 5:** Nếu muốn vào chế độ đóng mở thì ta nhấn phím A.

**Lưu ý** bước 3, 4, 5 có thể thay đổi vị trí từng bước cho nhau vì 3 chế độ này độc lập với nhau không cần phải theo quy trình.

### 4.6.2 Quy trình thao tác



**Hình 4.29** Quy trình thao tác

Hệ thống sau khi thiết kế thi công hoàn thiện được sử dụng dễ dàng đối với tất cả các đối tượng vì việc vận hành sử dụng rất dễ dàng theo quy trình hình 4.29 với 3 chế độ. Nhưng người sử dụng thường chỉ dùng để thao tác đóng mở cửa là thường xuyên, còn 2 chế độ nạp và xóa người dùng chỉ dùng cho lập trình viên hay chủ thiết bị. Thiết bị tuy đơn giản vẫn đảm bảo tính bảo mật cao.

## **Chương 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

### **5.1 KẾT QUẢ**

Sau thời gian tiến hành tìm hiểu, nghiên cứu các tài liệu chuyên ngành tiếng Việt cũng như tiếng Anh, tìm hiểu thêm qua mạng Internet, tổng hợp lại các kiến thức đã được học trong suốt 4 năm cũng như được sự hướng dẫn của thầy GVHD ThS. Nguyễn Việt Hùng. Nhóm chúng em cũng đã hoàn thành được đồ án tốt nghiệp với đề tài “THIẾT KẾ VÀ THI CÔNG HỆ THỐNG MỞ CỦA BẰNG CẢM BIẾN VÂN TAY” qua gần 15 tuần thực hiện.

Sau đề tài đồ án này, nhóm em cũng đã nghiên cứu và tích lũy được thêm nhiều hiểu biết, kiến thức mới như:

- Hiểu biết sâu hơn về sử dụng và các tính năng của Arduino như giao tiếp giữa Arduino với các module mở rộng như: module cảm biến vân tay R308, module Sim 900A, động cơ Servo, phím ma trận, màn hình LCD.
- Nghiên cứu và biết cách kết nối giữa Arduino với các module mở rộng: module cảm biến vân tay R308, module Sim 900A, động cơ Servo, phím ma trận, màn hình LCD và lắp vào mô hình để thành sản phẩm hoàn chỉnh.
- Nghiên cứu biết được cách sử dụng của cảm biến vân tay R308, nguyên lý hoạt động của cảm biến, các thông số kỹ thuật, các tính năng của cảm biến. Tìm hiểu sâu hơn và biết được qui trình lấy mẫu dấu vân tay, lịch sử phát triển của ngành sinh trắc học vân tay.
- Nghiên cứu biết được cách sử dụng module Sim 900A, nguyên lý hoạt động, các thông số kỹ thuật, tính năng của Sim 900A. Biết được cách thiết lập cho module sim, sử dụng tính năng gọi điện và nhắn tin SMS trên module sim để điều khiển thiết bị.
- Biết cách sử dụng thêm phần mềm vẽ mạch Altium (trước đó chỉ sử dụng Proteus) để thiết kế mạch in 2 lớp, làm mạch kết nối giữa kit Arduino với các module mở rộng để giảm sử dụng các dây cắm và các linh kiện gắn rời nhằm tăng tính nhỏ gọn cho mạch điều khiển.

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

---

Sau quá trình nghiên cứu, thi công thì đồ án “Thiết kế và thi công hệ thống mở cửa bằng cảm biến vân tay” của nhóm 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 R308. 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. Nếu sai thì không thực hiện.



**Hình 5.1** Chế độ đóng mở bằng vân tay

- Điều khiển đóng mở khóa cửa thành công qua tin nhắn SMS. Với tính năng này, khi người dùng tiến hành nhắn tin tới module Sim 900A, lúc này hệ thống sẽ tiến hành kiểm tra xem có đúng cú pháp tin nhắn và mật khẩu đã cài đặt hay không. Nếu đúng cú pháp tin nhắn và mật khẩu thì cho phép điều khiển động cơ Servo để mở chốt cửa. Nếu sai thì không thực hiện.
- Điều khiển đóng mở khóa cửa thành công bằng cách gọi điện thoại tới SIM 900A. Người dùng khi gọi tới module sim, hệ thống sẽ tiến hành kiểm tra xem số điện thoại gọi tới có đúng với số điện thoại đã được cho phép không. Nếu đúng thì cho phép mở khóa cửa. Nếu khác số điện thoại thì không thực thi.

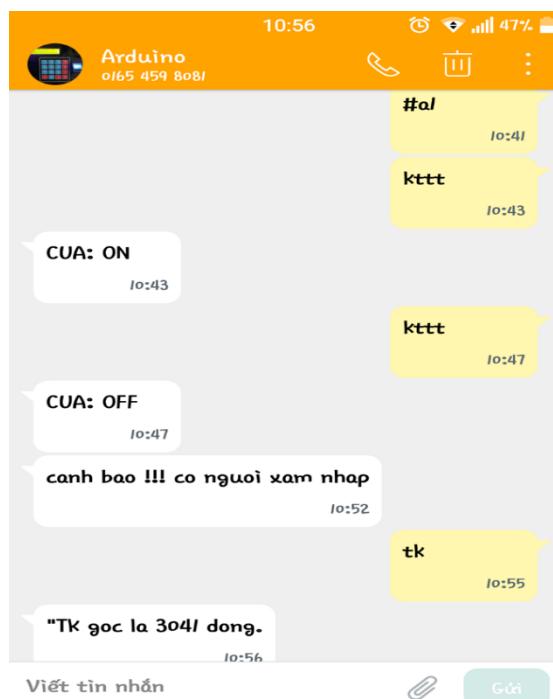
## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

---



**Hình 5.2** Chế độ đóng mở bằng cách gọi điện thoại

- Kiểm tra được trạng thái khóa hoặc mở khóa cửa qua tin nhắn SMS. Khi ở xa, ta muốn kiểm tra trạng thái cửa đã được khóa hay chưa khóa, thì tiến hành nhắn tin kiểm tra tới module sim. Hệ thống sẽ gửi tin nhắn tới số điện thoại người dùng trạng thái của cửa hiện tại là khóa hay mở.



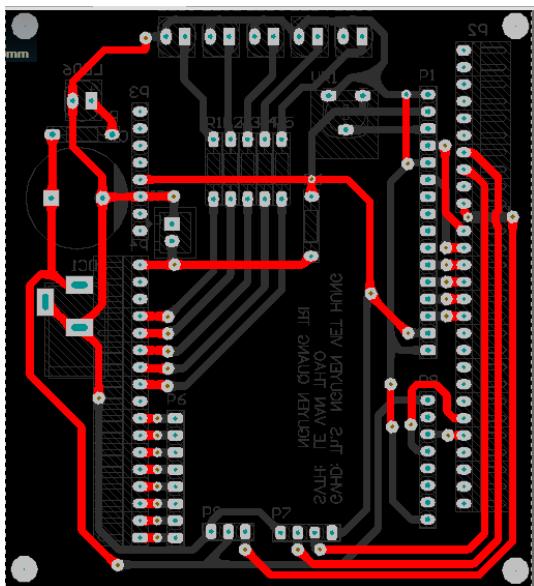
**Hình 5.3** Kiểm tra trạng thái cửa

- Hiển thị được trạng thái khóa hoặc mở khóa cửa trên màn hình LCD gắn bên ngoài cửa.

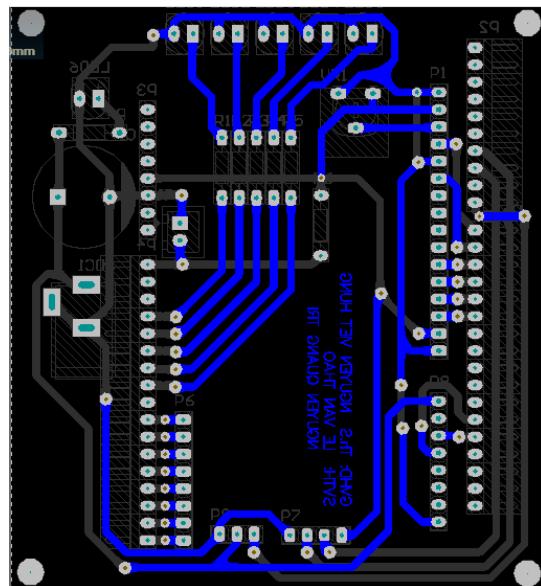
## CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ

- Bàn phím ma trận để chọn chế độ thêm hoặc thay đổi dấu vân tay truy cập. Có thể thay đổi được hoặc thêm mật khẩu vân tay để cấp quyền truy cập khóa, mở cửa cho người khác bằng cách nhập mật khẩu qua ma trận phím, sau đó tiến hành quét vân tay để thay đổi hoặc thêm dấu vân tay mới.

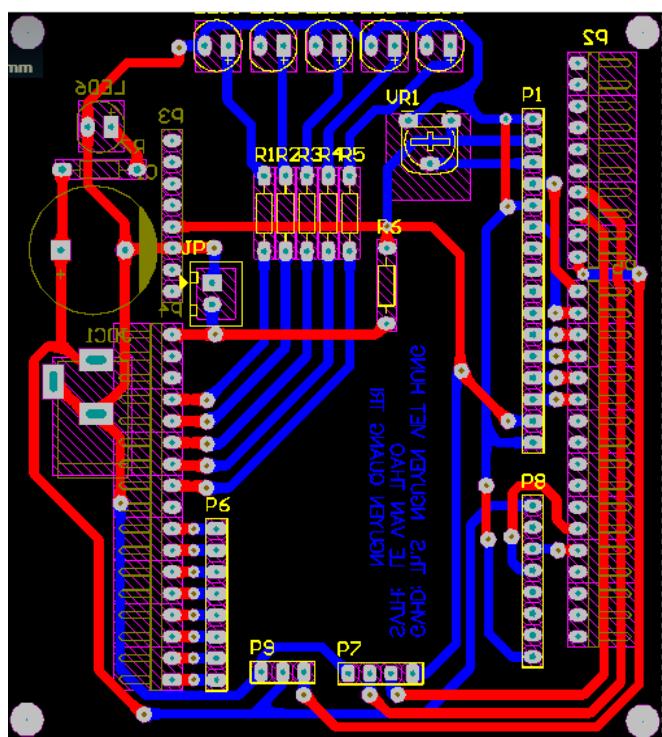
Sau đây là hình ảnh của mạch in 2 lớp để kết nối giữa Arduino với các module mở rộng được vẽ bằng phần mềm Altium ở hình 5.4 là lớp top, hình 5.5 là lớp bottom, còn hình 5.6 là dây lớp top và bottom nằm trên một mặt cho ta dễ quan sát hơn.



Hình 5.4 Mạch in mặt trên



Hình 5.5 Mạch in mặt dưới

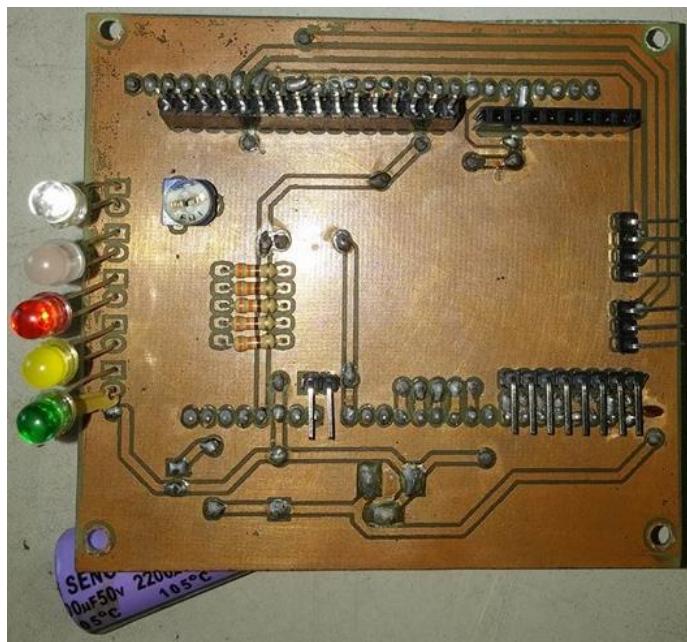


Hình 5.6 Sơ đồ mạch in

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

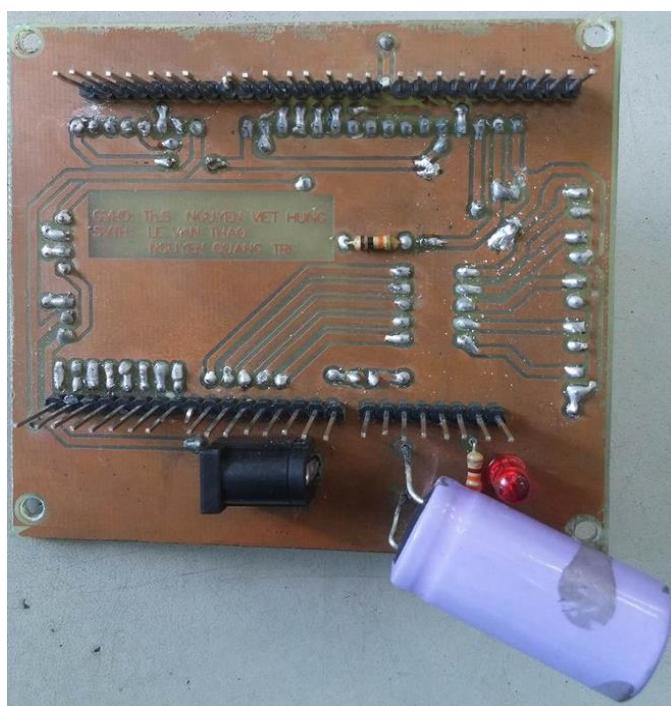
---

Hình ảnh thực tế:



**Hình 5.7** Hình ảnh thực tế gắn linh kiện cho mặt trên

Trong hình 5.7 hình ảnh thực tế lớp trên có các hàng rào dùng để kết nối với các thiết bị và module mở rộng như: cảm biến vân tay R308, Sim 900A, bàn phím ma trận, màn hình LCD, động cơ Servo; các led dùng để báo hiệu trạng thái hoạt động của hệ thống; biến trở dùng để tinh chỉnh cho LCD.

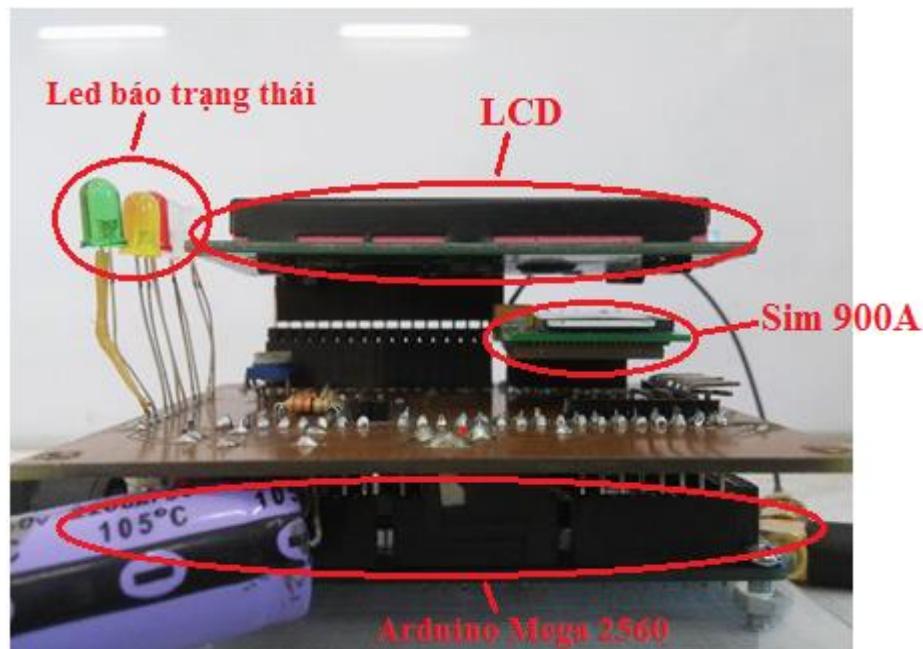


**Hình 5.8** Hình ảnh thực tế gắn linh kiện cho mặt dưới

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

Trong hình 5.8 hình ảnh thực tế lớp dưới các hàng rào được dùng để kết nối với kit Arduino, có giắc cắm để cấp nguồn cho các module mở rộng, tụ điện dùng để ổn định dòng và áp cho Sim 900A hoạt động ổn định, led báo hiệu có nguồn cấp.

Hình 5.9 kết nối thành mạch hệ thống hoàn chỉnh:



**Hình 5.9** Hình ảnh hệ thống sau khi kết nối

Khi lắp đặt hệ thống vào mô hình cửa bên trong như hình 5.10 còn bên ngoài như hình 5.11.

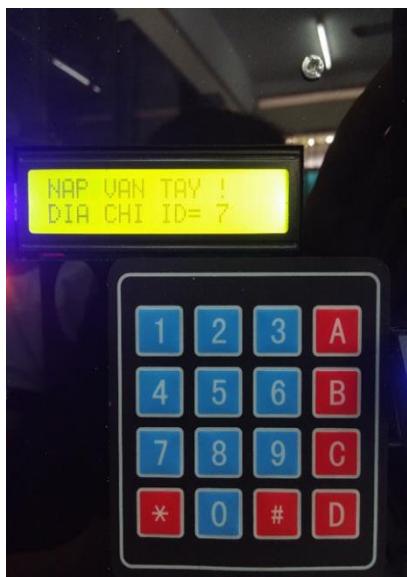


**Hình 5.10** Mặt bên trong mô hình cửa



**Hình 5.11** Mặt ngoài mô hình cửa

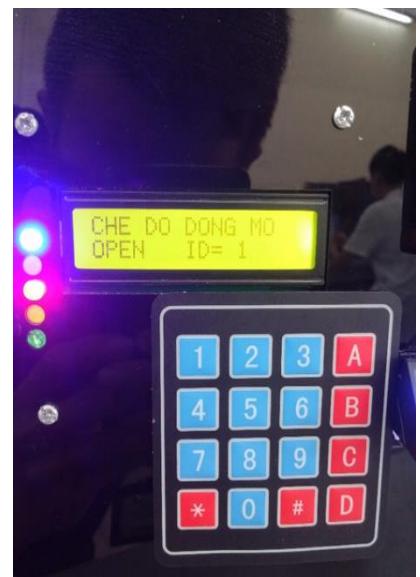
Mô hình cửa có các chế độ hoạt động cơ bản như: Nạp thêm vân tay mới vào để cấp thêm quyền truy cập (hình 5.12); Xóa dấu vân tay cũ đang được lưu trữ (hình 5.13); Chế độ đóng mở bằng cách sử dụng dấu vân tay hoặc dùng điện thoại để gọi và nhắn tin (hình 5.14)



**Hình 5.12** Nạp vân tay



**Hình 5.13** Xóa vân tay



**Hình 5.14** Đóng mở cửa

## 5.2 NHẬN XÉT – ĐÁNH GIÁ:

Sau thời gian nghiên cứu, thi công thì đồ án tốt nghiệp của nhóm 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.

## **CHƯƠNG 5. KẾT QUẢ\_NHẬN XÉT\_ĐÁNH GIÁ**

---

Hệ thống sử dụng nguồn cấp nhỏ từ 9V trở xuống nên an toàn cho người sử dụng trước nguy cơ điện giật

Do sử dụng công nghệ nhận dạng vân tay nên tính bảo mật của mô hình khá cao. Nếu sử dụng chức năng gọi điện thì chỉ khi đúng số điện thoại được cho phép thì hệ thống mới thực thi, cũng như khi nhắn tin tới thì đúng mật khẩu và cú pháp tin nhắn đã cài đặt thì mới điều khiển được cửa. Do đó tính bảo mật an toàn của mô hình cửa đạt được yêu cầu.

Thời gian đáp ứng từ khi quét vân tay đến khi mở và khóa cửa trong khoảng 1-2 giây. Thời gian đáp ứng khi sử dụng chức năng gọi điện và nhắn tin thì lâu hơn một chút, khoảng từ 3-4 giây. Thời gian đáp ứng như vậy cũng khá nhanh so với mô hình cửa thông thường, vì bình thường phải lấy chìa khóa ra và chọn chìa khóa thích hợp để mở cửa rất tốn thời gian.

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:

- Hạn chế lớn nhất là chưa có được nguồn điện dự trữ để cung cấp cho hệ thống hoạt động khi bị mất nguồn chính.
- Mô hình hình hoạt động chưa thực sự trọn 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

Sau thời gian tìm hiểu, nghiên cứu nhóm đã thiết kế thành công mạch điều khiển hệ thống cửa thông qua module cảm biến vân tay R308 và module SIM 900A với các tính năng đáp ứng được các nội dung và mục tiêu ban đầu đã đề ra:

- Giao tiếp và truyền dữ liệu thành công giữa Arduino Mega 2560 với các module cảm biến vân tay R308, module SIM 900A, màn hình LCD, phím ma trận, động cơ Servo SG90 9g.
- Điều khiển đóng mở khóa cửa thành công qua nhận dạng dấu vân tay.
- Điều khiển đóng mở khóa cửa thành công qua tin nhắn SMS.
- Điều khiển đóng mở khóa cửa thành công bằng cách gọi điện thoại tới SIM 900A.
- Kiểm tra được trạng thái khóa hoặc mở khóa cửa qua tin nhắn SMS.
- Hiển thị được trạng thái khóa hoặc mở khóa cửa trên màn hình LCD.
- Có thể thay đổi được hoặc thêm mật khẩu vân tay khi cần thiết.
- Ngoài ra, mạch điều khiển được thiết kế sẵn sàng cho việc tích hợp thêm các ngoại vi trong tương lai.

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 nhóm chỉ thực hiện được mô hình tương đối hoàn chỉnh.
- Hạn chế lớn nhất là chưa có được bộ nguồn dự trữ cấp cho hệ thống hoạt động khi bị mất nguồn chính.
- Mô hình hình hoạt động chưa thực sự trọn tru, việc tiến hành quét mã vân tay thực hiện hơi chậm.
- SIM 900A hoạt động chưa thực sự ổn định, đôi khi bị mất tín hiệu.

## **CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

---

Nhưng nhận xét tổng quan về hệ thống thì:

- 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.
- Hệ thống nhận, trả lời yêu cầu từ người dùng và gửi tín hiệu điều khiển của SIM 900A hoạt động khá ổn định, chính xác.
- Module cảm biến vân tay R308 thực hiện quét nhận diện dấu vân tay hoạt động ổn định, độ chính xác rất cao.

### **6.2 HƯỚNG PHÁT TRIỂN**

Hệ thống điều khiển cửa có thể mở rộng khả năng điều khiển bằng cách gắn thêm module Bluetooth để điều khiển đóng mở qua bluetooth, hoặc mở rộng khả năng đóng mở cửa thông qua bảo mật 2-3 lớp (như sau khi quét dấu vân tay thì tiến hành nhập mật khẩu để đóng mở khóa).

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

### Sách tham khảo

- [1] Lakhmi C. Jain, Ugur Halici, Isao Hayashi, SB Lee, Shigeyoshi Tsutsui, “Intelligent Biometric Techniques in Fingerprint and Face Recognition”, CRC Press, June 29 1999.
- [2] John Boxal, “ Arduino Workshop, Willinam Pollock”, May 2013.
- [3] Phạm Đức Huy Và Nguyễn Đình Toàn, “Hệ Thống Hiển Thị Thông Tin Và Chống Trộm Xe Máy Dùng Cảm Biến Vân Tay”, Đồ Án Tốt Nghiệp ĐH, Trường ĐH Sư Phạm Kỹ Thuật TP.HCM, 2016.
- [4] Đỗ Quang Vịnh, “Điều Khiển Thiệt Bị Qua SMS”, Đồ Án Tốt Nghiệp ĐH, Trường ĐH Sư Phạm Kỹ Thuật TP.HCM, 2013.
- [5] Adith Jagadish Boloor, “Arduino by Example”, Packt Publishing Ltd., September 2015.
- [6] “Arduino Door Locking System”, <http://www.instructables.com>
- [7] “Arduino Fingerprint Lock”, <http://www.instructables.com>

### DATASHEET

- [8] LCD 16x2, <http://www.picaxe.com/docs/led008.pdf>
- [9] Cảm biến vân tay,  
[http://iranmicro.com/images/pdfupload/r308\\_fingerprint\\_module\\_user\\_m.pdf](http://iranmicro.com/images/pdfupload/r308_fingerprint_module_user_m.pdf)
- [10] LCD 16x2, <https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf>
- [11] Tập lệnh AT, [http://www.espruino.com/datasheets/SIM900\\_AT.pdf](http://www.espruino.com/datasheets/SIM900_AT.pdf)
- [12] Sim 900A, <http://www.propox.com/download/docs/SIM900.pdf>
- [13] Động cơ Servo, <http://www.micropik.com/PDF/SG90Servo.pdf>
- [14] Arduino mega, [https://www.arduino.cc/en/uploads/Main/arduino-mega2560\\_R3-schematic.pdf](https://www.arduino.cc/en/uploads/Main/arduino-mega2560_R3-schematic.pdf)

## PHỤ LỤC

### ĐOẠN CODE CHƯƠNG TRÌNH CON

```
/*KIEM TRA NUT NHAN*/
void kiemtra_nutnhan()
{
    customKey = customKeypad.getKey();
    if ( customKey == 10 )
    {
        CHEDO = 0; // mo cua
        digitalWrite(A3, LOW);
        digitalWrite(A4, LOW);
        digitalWrite(A5, HIGH);
        tam = 0;
    }
    else if (customKey == 11)
    {
        CHEDO = 1; // nap van tay
        digitalWrite(A3, LOW);
        digitalWrite(A4, HIGH);
        digitalWrite(A5, LOW);
        tam = 0;
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("NAP VAN TAY ");
        lcd.setCursor(0,1);
        lcd.print("pass:");
        delay(1000);
    }
    else if ( customKey == 12 )
    {
        CHEDO = 2; // xoa van tay
        digitalWrite(A3, HIGH);
        digitalWrite(A4, LOW);
        digitalWrite(A5, LOW);
        tam = 0;
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("XOA VAN TAY!");
        lcd.setCursor(0,1);
        lcd.print("pass:");
        delay(1000);
    }
}
//////////
```

```
/* HÀM NẤP ĐẦU VÂN TAY */
void nap_vantay()
{
    Serial.println("Ready to enroll a fingerprint! Please Type in the ID # you want to save
this finger as... ");
    if ( CHEDO == 1)
    {
        lcd.setCursor(0,1);
        lcd.print("          ");
        lcd.setCursor(0,1);
        lcd.print("DIA CHI ID= ");
        while( id == 0 && CHEDO == 1)
        {
            kiemtra_nutnhan();
            id = nhapid1(bien);
        }
    }
    else
    {
        Serial.print("Enrolling ID #"); // vị trí lưu vân tay
        Serial.println(id);
    }
    while (! getFingerprintEnroll() );
}
uint8_t getFingerprintEnroll() {

int p = -1;

if ( CHEDO == 1)
{
    Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id); //
    lcd.setCursor(12,1);
    lcd.print(id);
    lcd.setCursor(0,1);
    lcd.print("          ");
    lcd.setCursor(0,1);
    lcd.print("Starts loading");
    //Cho ngón tay vào ghi danh
}
while (p != FINGERPRINT_OK && (CHEDO == 1) ) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK: // l?nh th?c hiên hoàn thành
            Serial.println("Image taken"); //h?nh ?nh ch?p
            break;
        case FINGERPRINT_NOFINGER: // không có ngón tay trên c?m bi?n
            Serial.println(".");
    }
}
```

```
break;
case FINGERPRINT_PACKETRECEIVEERR: // lỗi khi nhận gói dữ liệu;
    Serial.println("Communication error");//lỗi giao tiếp
    break;
case FINGERPRINT_IMAGEFAIL: // không ghi danh các ngón tay
    Serial.println("Imaging error"); // lỗi hình ảnh
    break;
default:
    Serial.println("Unknown error"); // Lỗi không rõ
    break;
}
}

// OK success!

if ( CHEDO == 1)
{
p = finger.image2Tz(1);
switch (p)
{
case FINGERPRINT_OK:
    Serial.println("Image converted"); // hình ảnh chuyển đổi
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy"); // hình ảnh quá lỏng xốp
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error"); // lỗi giao tiếp
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE: // không tạo ra hình ảnh
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}
Serial.println("Remove finger");// di chuyển ngón tay
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) // khi không có vân tay trên cảm biến
{
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
```

```
p = -1;
lcd.setCursor(0,1);
lcd.print("          ");
lcd.setCursor(0,1);
lcd.print("AGAIN");
Serial.println("Place same finger again"); // d?t ngón tay 1 l?i 1 l?n n?a
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken"); // h?nh ?nh ch?p
            break;
        case FINGERPRINT_NOFINGER:
            Serial.print(".");
            break;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Communication error");//l?i giao ti?p
            break;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Imaging error"); // l?i h?nh ?nh
            break;
        default:
            Serial.println("Unknown error"); // l?i kh?ng r?
            break;
    }
}
// OK success!
```

```
p = finger.image2Tz(2);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted"); // h?nh ?nh chuy?n đ?i
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy"); // h?nh ?nh quá l?n x?n
        return p;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error"); // l?i giao ti?p
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
```

```
Serial.println("Unknown error"); // lỗi không rõ
    return p;
}
// OK converted!
Serial.print("Creating model for #"); Serial.println(id); // lấy mẫu
p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error"); // lỗi giao tiếp
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match"); // Dấu vân tay không phù hợp
    return p;
}
else
{
    Serial.println("Unknown error"); // lỗi không rõ
    lcd.setCursor(0,1);
    lcd.print("          ");
    lcd.setCursor(0,1);
    lcd.print("Unknown error!");
    return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    Serial.println("Stored!"); // lưu trữ
    lcd.setCursor(0,1);
    lcd.print("          ");
    lcd.setCursor(0,1);
    lcd.print("Stored!");
    lcd.setCursor(8,1);
    lcd.print("ID");
    lcd.setCursor(12,1);
    lcd.print(id);
    delay (2000);
    id = 0;
    lcd.setCursor(0,1);
    lcd.print("          ");
    lcd.setCursor(0,1);
    lcd.print("DIA CHI ID= ");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error"); // lỗi giao tiếp
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
```

```
Serial.println("Could not store in that location"); // không thực hiện lưu vị trí này
return p;
} else if (p == FINGERPRINT_FLASHERR) {
Serial.println("Error writing to flash"); // lỗi ghi vào
return p;
} else {
Serial.println("Unknown error"); // lỗi không rõ
lcd.setCursor(0,1);
lcd.print("          ");
lcd.setCursor(0,1);
lcd.print("Unknown error!");
return p;
}
}
}

uint8_t readnumber(void) {
uint8_t num = 0;
boolean validnum = false;
while ( (CHEDO == 1)) {
while (! Serial.available() && (CHEDO == 1));
char c = Serial.read();
if (isdigit(c)) {
num *= 10;
num += c - '0';
validnum = true;
} else if (validnum) {
return num;
}
}
}
///////////////////////////////
/* XÓA VÂN TAY*/
void xoa_vantay()
{
while (!Serial );
delay(500);
Serial.println("Type in the ID # you want delete..."); // Địa chỉ muốn xóa
uint8_t id = 0;
while (true)
{
giatri = nhapid(bien);
if(giatri != -256 && giatri != 0)
{
id = giatri;
giatri = 0;
}
if( id != 0 ) break;
```

```
if( CHEDO != 2 ) break;
}
if (id != 0) {
Serial.print("deleting ID #");
Serial.println(id);
lcd.setCursor(0,1);
lcd.print("DIA CHI ID= ");
delay(1000);
deleteFingerprint(id);
}
}

uint8_t deleteFingerprint(uint8_t id) {
uint8_t p = -1;
p = finger.deleteModel(id);
if (p == FINGERPRINT_OK) {
Serial.println("Deleted!"); // đã xóa
lcd.setCursor(0,1);
lcd.print("          ");
lcd.setCursor(0,1);
lcd.print("Deleted");
lcd.setCursor(8,1);
lcd.print("ID");
lcd.setCursor(11,1);
lcd.print(id);
delay (2000);
id = 0;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("XOA VAN TAY ");
lcd.setCursor(0,1);
lcd.print("DIA CHI ID= ");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
Serial.println("Communication error"); // lỗi giao tiếp
return p;
} else if (p == FINGERPRINT_BADLOCATION) {
Serial.println("Could not delete in that location"); // không thể xóa vị trí này
return p;
} else if (p == FINGERPRINT_FLASHERR) {
Serial.println("Error writing to flash");
return p;
} else {
Serial.print("Unknown error: 0x"); Serial.println(p, HEX);
return p;
}
}

///////////////////////////////
/*MO CUA*/
```

## PHỤ LỤC

---

```
void mo_cua()
{
    getFingerprintIDez();
    digitalWrite(A7,HIGH);
    delay(20);
}

uint8_t getFingerprintID() {
    uint8_t p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image taken"); // hình ảnh chụp
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println("No finger detected"); // không phát hiện dấu vân tay
            return p;
        case FINGERPRINT_PACKETRECIEVEERR:
            Serial.println("Communication error"); // lỗi giao tiếp
            return p;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Imaging error");//lỗi hình ảnh
            return p;
        default:
            Serial.println("Unknown error"); // lỗi không biết
            return p;
    }
    // OK success!
    p = finger.image2Tz();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image converted"); //hình ảnh chuyển đổi
            break;
        case FINGERPRINT_IMAGEMESS:
            Serial.println("Image too messy"); // hình ảnh quá lộn xộn
            return p;
        case FINGERPRINT_PACKETRECIEVEERR:
            Serial.println("Communication error"); // lỗi giao tiếp
            return p;
        case FINGERPRINT_FEATUREFAIL:
            Serial.println("Could not find fingerprint features");
            return p;
        case FINGERPRINT_INVALIDIMAGE:
            Serial.println("Could not find fingerprint features");
            return p;
        default:
            Serial.println("Unknown error"); // lỗi không rõ
            return p;
    }
}
```

```
// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!"); // tìm thấy dấu vân tay
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error"); // lỗi giao tiếp
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match"); // không tìm thấy dấu vân tay
    return p;
} else {
    Serial.println("Unknown error"); // lỗi không rõ
    return p;
}
// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID); // tìm thấy dấu vân tay.
Serial.print(" with confidence of "); Serial.println(finger.confidence); //chỉ số tin cậy
}
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return -1;

    // found a match!
    digitalWrite(A7,LOW);
    digitalWrite(A6,HIGH);
    delay (500);
    digitalWrite(A6,LOW);
    delay (500);
    cua = cua + 1;
    gServo.attach(SERVO_PIN);
    if(cua >= 2)
    {
        cua = 0;
    }
    lcd.setCursor(7,1);
    lcd.print("ID= ");
    lcd.setCursor(11,1);
    lcd.print(finger.fingerID);
    Serial.print("Found ID #"); Serial.print(finger.fingerID);
```

```
Serial.print(" with confidence of "); Serial.println(finger.confidence);
return finger.fingerID;
}
///////////////////////////////
/* PHIM MA TRAN*/
void keypadEvent(KeypadEvent eKey){
switch (keypad.getState()){
case PRESSED:
Serial.print("Pressed: ");
Serial.println(eKey);

lcd.setCursor(0,1);
lcd.print("pass:");
lcd.setCursor( i=i+1,1);
lcd.print("*");
switch (eKey){
case '*':
if(CHEDO == 1)
{
i = 5;
checkPassword_napvantay(); break;
}
else if (CHEDO == 2)
{
i = 5;
checkPassword_xoavantay(); break;
}
case '#': password.reset(); break;
default: password.append(eKey);
}
}
}
/////////////////////////////
/* NHAP MAT KHAU DE NAP VAN TAY*/
void checkPassword_napvantay()
{
if (password.evaluate())
{
Serial.println("Success");
password.reset();
tam = 1;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("NAP VAN TAY !");
while( tam == 1)
{
nap_vantay();
}
```

```
        }
    }
else
{
    Serial.println("Wrong");
    tam = 0;
    lcd.setCursor(0,1);
    lcd.print("          ");
    lcd.setCursor(0,1);
    lcd.print("wrong!");
    delay(2000);
    i = 5;
    password.reset();
    lcd.setCursor(0,1);
    lcd.print("          ");
    lcd.setCursor(0,1);
    lcd.print("pass:");
}
}

//////////////////////////////////////////////////////////////////
/*NHAP MAT KHAU XOA VAN TAY*/
void checkPassword_xoavantay()
{
if (password.evaluate())
{
    Serial.println("Success");
    tam = 2;
    lcd.setCursor(0,1);
    lcd.print("          ");
    lcd.setCursor(0,1);
    lcd.print("Success");
    password.reset();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("XOA VAN TAY !");
    lcd.setCursor (0,1);
    lcd.print("DIA CHI ID=");
    while ( tam == 2)
    {
        xoavantay();
    }
}
else
{
    Serial.println("Wrong");
    tam = 0;
    lcd.setCursor(0,1);
```

```
lcd.print("      ");
lcd.setCursor(0,1);
lcd.print("wrong!");
delay(2000);
i = 5;
password.reset();
lcd.setCursor(0,1);
lcd.print("      ");
lcd.setCursor(0,1);
lcd.print("pass:");
}
}
///////////////////////////////
/* nhap id xoa van tay*/
int nhapid(int bien)
{
while( CHEDO == 2)
{
kiemtra_nutnhan();
idkey = custom.getKey();
if(CHEDO != 2) break;
if (idkey != NO_KEY)
{
int id = idkey - '0';
so[z] = id;
if (idkey != '*' )
{
lcd.setCursor(12+z,1);
lcd.print(id);
}
z++;
if (idkey == '*' )
{
for(int k = 0 ; k < z-1; k++)
{
ID = ID * 10 + so[k];
}
for(int k = 0 ; k < z-1; k++)
{
so[k] = ' ';
}
bien = ID;
Serial.println(ID);
ID = 0;
z = 0;
return(bien);
}
}
```

```
    }
    return(bien);
}
}
///////////
/* NHAP ID DE NAP VAN TAY */
int nhapid1(int bien)
{
    while( CHEDO == 1)
    {
        kiemtra_nutnhan();
        idkey = custom.getKey();
        if(CHEDO != 1) break;
        if (idkey != NO_KEY )
        {
            int id = idkey - '0';
            so[z] = id;
            if (idkey != '*' )
            {
                lcd.setCursor(12+z,1);
                lcd.print(id);
            }
            z++;
            if (idkey == '*' )
            {
                for(int k = 0 ; k < z-1; k++)
                {
                    ID = ID * 10 + so[k];
                }
                for(int k = 0 ; k < z-1; k++)
                {
                    so[k] = ' ';
                }
                bien = ID;
                Serial.println(ID);
                ID = 0;
                z = 0;
                return(bien);
            }
        }
        return(bien);
    }
}
///////////
/*NHẮN TIN */
void nhantin()
{
```

## PHỤ LỤC

---

```
if(Serial2.available() >0)
{
    inchar=Serial2.read();
    //delay(10);
    if (inchar == 'k')
    {
        inchar=Serial2.read();
        delay(10);
        if (inchar == 't')
        {
            inchar=Serial2.read();
            delay(10);
            if (inchar == 't')
            {
                inchar=Serial2.read();
                delay(10);
                if (inchar == 't')
                {
                    sendSMS(SMS_Text,phoneNumber);
                    Serial2.println("AT+CMGD=1,4"); // delete all SMS;
                    Serial2.println(SMS_Text);
                    Serial.println("AT+CMGD=1,4"); // delete all SMS;
                    Serial.println(SMS_Text);
                }
            }
        }
    }
}

if (inchar == 't')
{
    inchar=Serial2.read();
    delay(10);
    if (inchar == 'k')
    {
        k = true;
        kiemtrataikhoan();
        sendSMS(taikhoan,phoneNumber);
        Serial2.println("AT+CMGD=1,4"); // delete all SMS;
        Serial2.println(taikhoan);
        Serial.println("AT+CMGD=1,4"); // delete all SMS;
        Serial.println(taikhoan);
        taikhoan = "";
    }
}

if (inchar=='#')
{
```

```
delay(10);
inchar=Serial2.read();
if (inchar=='a')
{
    delay(10);
    inchar=Serial2.read();
    if (inchar=='1')
    {
        delay(10);
        cua = 1;
        gServo.attach(SERVO_PIN);
        Serial.println(cua);
    }
    else if (inchar=='0')
    {
        delay(10);
        cua = 0;
        Serial.println(cua);
    }
    delay(10);
}
Serial2.println("Done");
Serial.println("Done");
delay(1000);
}

if (inchar=='0')
{
    inchar=Serial2.read();
    if (inchar=='1')
    {
        inchar=Serial2.read();
        if (inchar=='6')
        {
            inchar=Serial2.read();
            if (inchar=='5')
            {
                inchar=Serial2.read();
                if (inchar=='5')
                {
                    inchar=Serial2.read();
                    if (inchar=='1')
                    {
                        inchar=Serial2.read();
                        if (inchar=='6')
                        {
                            inchar=Serial2.read();
                        }
                    }
                }
            }
        }
    }
}
```



## PHỤ LỤC

---

```
Serial2.print("AT+CMGS=\\""); // Gửi tin nhắn đến thuê bao bằng cách sử dụng lệnh
AT+CMGS="số điện thoại".
Serial2.print(phoneNumber);
Serial2.println("\\");
Serial.println();
Serial.print("AT+CMGS=\\""); // Gửi tin nhắn đến thuê bao bằng cách sử dụng lệnh
AT+CMGS="số điện thoại".
Serial.print(phoneNumber);
Serial.println("\\");
delay(1000);
Serial2.print(message);
Serial.print(message);
delay(500);
Serial2.write(0x1A); //Gửi nội dung tin nhắn và kết thúc bằng kí tự có mã ASCII
0x1A.
Serial2.write(0x0D); //CR
Serial2.write(0x0A); //LF
delay(1000);
}
///////////
/*HANH TRINH*/
void HANHTRINH1()
{
    HANHTRINH = digitalRead(buttonPin);
}
///////////
/* KIEM TRA TAI KHOAN SIM*/
void kiemtrataikhoan()
{
    while (k == true)
    {
        Serial2.write("ATD*101#;"); //Specify the number to call
        Serial2.write(0x0D); //CR
        Serial2.write(0x0A); //LF
        boolean j = false;
        delay(5000);
        {
            while ((Serial2.available() >0))
            {
                char tk = Serial2.read();
                if(tk == '\\')
                {
                    j = true;
                }
                if(j == true)

```

```
{  
    taikhoan +=tk;  
    delay(20);  
}  
if (tk == '.')  
{  
    j = false;  
    break;  
}  
}  
Serial.println(taikhoan);  
k = false;  
}  
}
```

