

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ GTVT
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP LỚN
XÂY DỰNG CÁC HỆ THỐNG NHÚNG

Tên đề tài:
THIẾT KẾ HỆ THỐNG CỬA TỰ ĐỘNG

GIÁO VIÊN HƯỚNG DẪN: LÃ QUANG TRUNG
NHÓM SINH VIÊN THỰC HIỆN:

- 1. TRƯƠNG THÀNH ĐẠT**
- 2. NGUYỄN TUẤN DŨNG**
- 3. ĐỖ ĐÌNH ANH ĐỨC**

LỚP : 73DCTT23

HÀ NỘI 01-2024

NHẬN XÉT ĐÁNH GIÁ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Chữ ký của giáo viên

LỜI NÓI ĐẦU

Trong thời đại công nghệ phát triển vượt bậc, hệ thống nhúng đóng vai trò quan trọng trong việc xây dựng các giải pháp tự động hóa thông minh. Với sự kết hợp chặt chẽ giữa phần cứng và phần mềm, hệ thống nhúng không chỉ mang lại hiệu quả cao mà còn nâng cao chất lượng cuộc sống, đáp ứng nhu cầu ngày càng đa dạng của xã hội hiện đại.

Từ mong muốn áp dụng kiến thức lý thuyết vào thực tiễn, nhóm chúng tôi đã lựa chọn đề tài "**Hệ thống cửa tự động**" để nghiên cứu và triển khai trong khuôn khổ môn học *Xây dựng hệ thống nhúng*. Đề tài hướng đến thiết kế một hệ thống cửa tự động thông minh với các tính năng bảo mật cao, hiện đại và thân thiện với người dùng, thông qua việc sử dụng vi điều khiển Arduino cùng các module cảm biến, thiết bị ngoại vi và linh kiện điện tử.

Báo cáo này trình bày chi tiết quá trình nghiên cứu, thiết kế, lập trình và thử nghiệm hệ thống, đồng thời nhấn mạnh tính ứng dụng thực tiễn của hệ thống nhúng trong lĩnh vực tự động hóa. Qua đề tài này, nhóm không chỉ củng cố kiến thức chuyên môn mà còn rèn luyện kỹ năng làm việc nhóm, tư duy sáng tạo và khả năng giải quyết vấn đề một cách hiệu quả.

Nhóm hy vọng báo cáo sẽ mang lại cái nhìn rõ nét về tiềm năng của hệ thống nhúng và đóng góp một phần ý tưởng cho các ứng dụng tự động hóa trong tương lai.

Mục lục

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI.....	6
1.1 Đặt vấn đề	6
1.2. Mục tiêu nghiên cứu	7
1.3. Phương pháp nghiên cứu.....	7
CHƯƠNG 2. GIỚI THIỆU LINH KIỆN SỬ DỤNG TRONG NGHIÊN CỨU ĐỀ TÀI.....	8
2.1. Arduino Mega 2560	8
2.2. Động cơ Servo SG90	9
2.3. KEYPAD	9
2.4. Module RFID RC522 và thẻ RFID S50 13,56 MHz.....	10
2.5.. Màn hình LCD 20x4	12
2.6. Cảm biến chuyển động HC-SR501	13
2.7. Còi chirp 5v	14
2.8. Đèn led.....	14
CHƯƠNG 3. SƠ ĐỒ CHỨC NĂNG.....	15
CHƯƠNG 4. NGUYÊN LÝ HOẠT ĐỘNG TỪNG THÀNH PHẦN	16
4.1 Thư viện và khai báo biến	16
4.2. Hàm setup().....	17
4.3. Hàm loop()	18
4.4. Master Mode (handleMasterMode).....	18
4.5. Normal Mode (handleNormalMode)	19
KẾT LUẬN.....	24

Bảng Phân Công Công Việc

STT	Họ & Tên	Mã Sinh Viên	Công Việc	Ghi Chú
1	Trương Thành Đạt	73DCTT23465	RFID và thẻ : Viết hàm đọc UID, thêm/xóa thẻ hợp lệ, quản lý Master Mode	TN
2	Đỗ Đình Anh Đức	73DCTT23413	Mật khẩu và bảo mật: Xử lý nhập mật khẩu, lưu và tải mật khẩu, cảnh báo bảo mật.	
3	Nguyễn Tuấn Dũng	73DCTT22471	Điều khiển cửa và cảm biến: Điều khiển servo, cảm biến PIR, và nút nhấn vật lý.	

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

1.1 Đặt vấn đề

Một căn nhà “thông minh” cần phải thông minh ngay từ thêm cửa . Một chiếc khóa thông minh sẽ dẫn lối vào tiện nghi trong cuộc sống thường ngày của mỗi người .

Khóa cửa thông minh là một thiết bị cơ điện có tác dụng thực hiện các nhiệm vụ mở/ khóa khóa cho cửa khi nhận được lệnh từ một thiết bị được xác thực, sử dụng kết nối không dây với một khóa mã để thực hiện quá trình xác nhận .

Cũng như khóa cửa truyền thống, khóa cửa thông minh bao gồm hai bộ phận là ổ khóa và chìa khóa . Tuy nhiên, chìa khóa ở khóa thông minh không tồn tại ở dạng vật chất, mà nó nằm trong ứng dụng của điện thoại thông minh, thẻ từ hoặc cảm biến sinh trắc học (như vân tay, võng mạc, giọng nói, ...) . Với những loại khóa đơn giản hơn, “chìa khóa” có thể đơn giản là một đoạn mã số . Khóa cửa thường được gắn trực tiếp vào ổ khóa thường, bao gồm các bộ phận thu và phát tín hiệu, cũng như thiết bị mở và khóa chốt .

Các hệ thống khóa cửa thông minh được sử dụng phổ biến hiện nay là:

- Khóa vân tay: Hệ thống này cho phép vân tay của người sử dụng được mã hóa trên thiết bị và chỉ khi có bàn tay đó đặt vào thì khóa mới tự động mở (*tính năng này có thể được phát triển thêm trong đề tài này*)
- Khóa mật mã: Là loại khóa dùng mật mã thay cho chìa khóa để mở cửa
- Khóa bằng thẻ RFID: RFID là viết tắt của cụm từ Radio Frequency Identification, là công nghệ nhận dạng các đối tượng dựa trên bước sóng vô tuyến

Không chỉ thực hiện thao tác đóng hay mở cửa, khóa thông minh còn cho phép chủ nhân căn nhà có thể tạo quyền cho bạn bè, người thân hay những người khác mở được khóa để vào nhà bằng chìa khóa ảo . Chiếc chìa khóa ảo này có thể được gửi đi qua e-mail, hay tin nhắn SMS . Nhận được mã khóa trong tay, người nhận có thể mở được cửa nhà vào những thời điểm đã được chủ nhà chỉ định.

Khóa cửa thông minh còn có thể chấp nhận hay từ chối quyền truy nhập từ xa thông qua ứng dụng điện thoại . Một vài chiếc khóa được cài đặt sẵn wifi để thực hiện các hoạt động giám sát như thông báo cho chủ nhà khi có người vào nhà hay truyền đi hình ảnh từ camera về người ra vào.

Ngoài ra, còn có rất nhiều chức năng, tiện ích khác được các nhà sản xuất phát minh và ứng dụng vào sản phẩm khóa thông minh theo những cách khác nhau: đèn LED để sử dụng trong bóng tối, quét mã vân tay, quét võng mạc, tích hợp chuông báo động,...

Khóa cửa thông minh đem lại nhiều tiện nghi và yên tâm cho các hộ gia đình và nơi công sở . Việc sử dụng khóa thông minh giúp tiết kiệm thời gian và công sức theo nhiều cách khác nhau:

- Một trong những tiện ích dễ thấy nhất là tránh được việc làm mất chìa khóa. Chìa khóa thường tương đối nhỏ gọn nhưng cũng vì thế nên dễ bị rơi, mất . Với khóa thông minh, người dùng chỉ cần nhớ được mật mã, hoặc thậm chí không cần phải nhớ mã số sử dụng ổ khóa có nhận dạng vân tay.
- Với khóa thông minh, người dùng không cần phải mất thời gian và tiền bạc làm thêm bản sao chìa khóa . Tất cả những gì cần làm chỉ đơn giản là chia sẻ mã khóa qua điện thoại hay e-mail.

- Thao tác nhanh chóng, thuận tiện . Người dùng sẽ không phải vật lộn với một ổ khóa đã cũ trong khi đang vội vã, hay tìm cách xoay xở khi đang phải mang vác nhiều đồ đạc.
- Do không cần phải tác động vật lý nhiều như khóa truyền thống nên ổ khóa thông minh thường bền hơn.
- Tự động khóa khi cửa đóng.
- Hoạt động với nguồn điện riêng.
- Một vài loại khóa thông minh giúp người dùng có thể kiểm soát, xem được hình ảnh và lưu trữ dữ liệu về người ra vào, do đó có thể kiểm soát tình trạng an ninh của căn hộ và tòa nhà.
- Thiết kế sang trọng, đa dạng, phù hợp với nhiều phong cách kiến trúc khác nhau.
- Dễ dàng thiết lập, cài đặt loại mã khóa phù hợp.
- Dễ dàng quản lý đối với những tòa nhà, trung tâm lớn.

Ngoài những tiện ích trên, khóa thông minh cũng tiềm ẩn những rủi ro như:

- Với những loại khóa dùng mã số, mã khóa có thể dễ bị tiết lộ hay bị phá mã.
- Chi phí lắp đặt và bảo trì tương đối cao.
- Với các hệ thống khóa cửa của các tòa nhà, trung tâm lớn, do khóa được vận hành trên hệ thống máy chủ tự động nên có thể tiềm tàng rủi ro máy chủ bị xâm nhập.
- Để duy trì tính an toàn và bảo mật, khóa cửa thông minh cần liên tục được bảo trì và nâng cấp để chống lại xâm nhập.
- Mặc dù có những rủi ro nhất định, khóa cửa thông minh chắc chắn vẫn sẽ được sử dụng trong tương lai . Trong những năm tiếp theo, việc sử dụng khóa cửa thông minh như khóa vân tay, khóa dùng thẻ, khóa mã số sẽ dần thay thế cách khóa truyền thống do tính năng vượt trội trong việc đảm bảo giám sát ra vào ở mức độ cao nhất cho các khu vực cần có sự kiểm soát ra vào chặt chẽ.

1.2. Mục tiêu nghiên cứu

Hệ thống cửa thông minh cần thực hiện những nhiệm vụ sau:

- Tự động khóa sau khi mở
- Sử dụng đồng thời loại “chìa khóa thông minh” là thẻ RFID và KeyPad
- Có thể mở/đóng cửa từ phía trong
- Có hệ thống còi báo động khi nhập đúng mật khẩu
- Đèn led báo hiệu trạng thái cửa .
- Cảm biến không đóng cửa khi phát hiện chuyển động

1.3. Phương pháp nghiên cứu

- Tìm hiểu tổng quan về lý thuyết của đề tài
- Đọc hiểu các tài liệu liên quan đến đề tài
- Thảo luận nhóm để thống nhất ý kiến

- Thiết kế phần cứng
- Thiết kế phần mềm
- Thực nghiệm và kiểm chứng sản phẩm

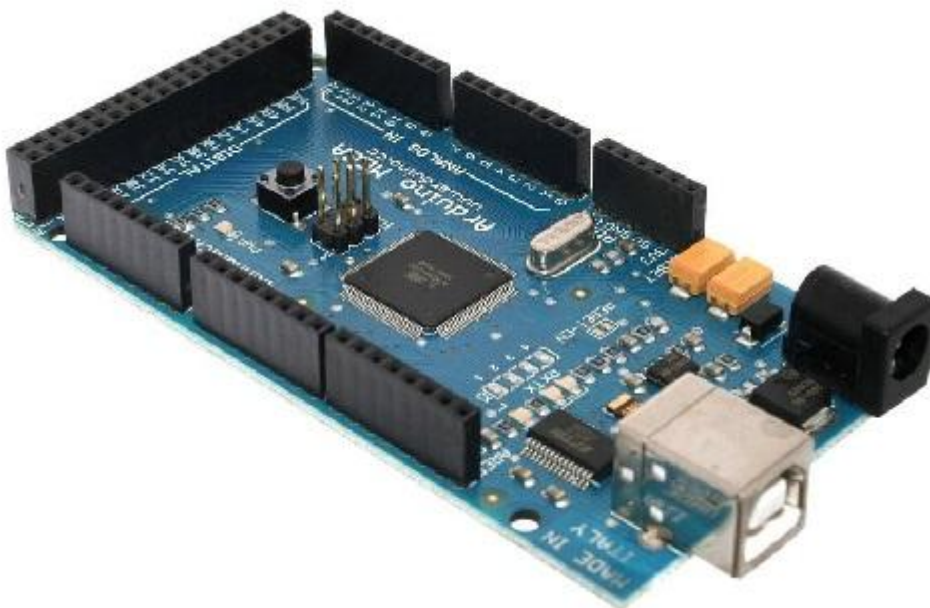
CHƯƠNG 2. GIỚI THIỆU LINH KIỆN SỬ DỤNG TRONG NGHIÊN CỨU ĐỀ TÀI

2.1. Arduino Mega 2560

Arduino Mega2560 khác với tất cả các vi xử lý trước giờ vì không sử dụng FTDI chip điều khiển chuyển tín hiệu từ USB để xử lý. Thay vào đó, nó sử dụng ATmega16U2 lập trình như là một công cụ chuyển đổi tín hiệu từ USB. Ngoài ra, Arduino Mega2560 cơ bản vẫn giống Arduino Uno R3, chỉ khác số lượng chân và nhiều tính năng mạnh mẽ hơn, nên các bạn vẫn có thể lập trình cho con vi điều khiển này bằng chương trình lập trình cho Arduino Uno R3.

Thông số kỹ thuật:

- Vi điều khiển chính: ATmega2560
- IC nạp và giao tiếp UART: CH340.
- Nguồn nuôi mạch: 5VDC từ cổng USB hoặc nguồn ngoài cắm từ giắc tròn DC (nếu sử dụng nguồn ngoài từ giắc tròn DC Hshop.vn khuyên bạn nên cấp nguồn từ 6~9VDC để đảm bảo mạch hoạt động tốt, nếu bạn cắm 12VDC thì IC ổn áp rất nóng, dễ cháy và gây hư hỏng mạch).
- Số chân Digital I/O: 54 (trong đó 15 chân có khả năng xuất xung PWM)
- Số chân Analog Input: 16
- Dòng điện DC Current trên mỗi chân I/O: 20mA
- Dòng điện DC Current chân 3.3V: 50mA
- Flash Memory: 256 KB trong đó 8 KB sử dụng cho bootloader.
- SRAM: 8 KB
- EEPROM: 4 KB
- Clock Speed: 16 MHz
- LED_BUILTIN: 13
- Kích thước: 101.52 x 53.3 mm



2.2. Động cơ Servo SG90

Động cơ RC Servo 9G là động phổ biến dùng trong các mô hình điều khiển nhỏ và đơn giản như cánh tay robot . Động cơ có tốc độ phản ứng nhanh, được tích hợp sẵn Driver điều khiển động cơ, dễ dàng điều khiển góc quay bằng phương pháp điều độ rộng xung PWM .

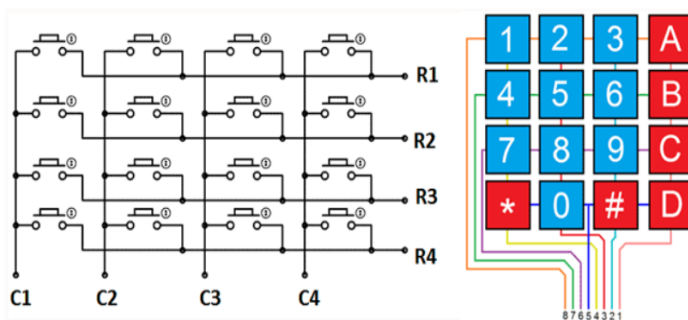
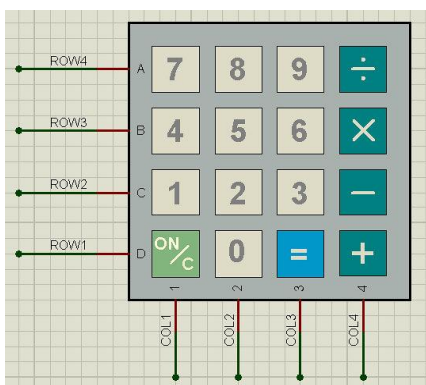
Kích thước:	23x12.2x29 mm
Mô-men xoắn:	1,8 kg/cm (4,8 V DC)
Tốc độ hoạt động:	60 ⁰ trong 0,1 giây
Điện áp hoạt động:	4,8V DC (~5V DC)
Nhiệt độ hoạt động:	0~55 °C
Delay:	10 us



2.3. KEYPAD

Bàn phím mềm 4×4 keypad có thiết kế nhỏ gọn, dễ kết nối và sử dụng, các chân của 16 phím được nối theo ma trận, tín hiệu khi nhấn phím sẽ là tín hiệu GND (0VDC) hoặc Vcc (5VDC) tùy vào cách quét phím của các bạn kích vào chân Vi điều khiển, bàn phím còn tích hợp vị trí để lắp thêm tụ chống dội (chống nhiễu), phù hợp cho các ứng dụng điều khiển bằng phím bấm.

Bàn phím tích hợp trong nhiều module mạch điện tử như kit phát triển, kit học tập



giao tiếp các vi điều khiển như Pic, 8051, AVR,

STM,.....

Chân	8
Số nút nhấn .	16
Điện áp hoạt động .	3.3v – 5v .
Nhiệt độ hoạt động .	0°C – 70°C .

2.4. Module RFID RC522 và thẻ RFID S50 13,56 MHz

a) Tổng quan về RFID

RFID (Radio Frequency Identification) là công nghệ nhận dạng đối tượng bằng sóng vô tuyến . Công nghệ này cho phép nhận biết các đối tượng thông qua hệ thống thu phát sóng radio, từ đó có thể giám sát, quản lý hoặc lưu vết từng đối tượng . Ví dụ, bạn đi siêu thị, bỏ hàng vào xe đẩy và chỉ đơn giản đẩy thẳng xe qua cổng giám sát . Một thiết bị tự động nhận dạng từng món hàng bạn mua và tự động trừ vào tài khoản thanh toán của bạn . Nhanh và tiện lợi biết bao! Đó chỉ là một trong rất nhiều ứng dụng tiện ích của công nghệ RFID .

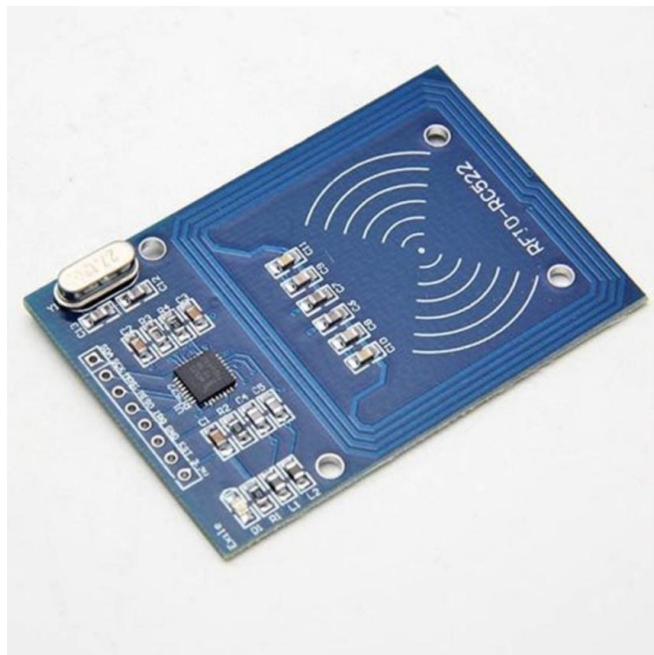
Công nghệ RFID được ứng dụng rộng rãi trong nhiều lĩnh vực, nhất là trong quản lý và tồn trữ hàng hóa . Ví dụ, dùng những thẻ RFID theo dõi nhiệt độ gắn lên hàng hóa có thể giúp nhà sản xuất theo dõi nhiệt độ trong kho lạnh . Những thẻ này sẽ truyền dữ liệu qua đầu đọc, đầu đọc liên tục truyền dữ liệu thu được từ các thẻ để truyền về máy tính trung tâm và lưu lại dữ liệu thu được . Từ đó, nhà sản xuất có thể truy cập vào internet từ bất cứ nơi nào cũng có thể theo dõi được dữ liệu bảo quản hàng hóa của mình trong các kho lạnh .

Ngoài ra còn có thể sử dụng thẻ RFID cấy vào vật nuôi để nhận dạng nguồn gốc và theo dõi vật nuôi tránh thất lạc và bị đánh cắp . Trong thư viện, các thẻ RFID được gắn với các cuốn sách giúp giảm thời gian tìm kiếm và kiểm kê, chống được tình trạng ăn trộm sách . Một số lĩnh vực có khả năng sử dụng một số lượng lớn các thẻ RFID như thẻ thông minh, chứng minh nhân dân, hộ chiếu điện tử, hàng hóa trong siêu thị, quản lý hành lý trong hàng không, hệ thống giao thông công cộng, các ngành may mặc, giày dép... .

b) Module RFID RC522

Module RFID RC522 sử dụng IC MF RC522 của Phillip dùng để đọc và ghi dữ liệu cho thẻ NFC tần số 13,56 MHz . Dưới đây là bảng thông số kỹ thuật chính của module RFID RC522.

Nguồn cấp:	3,3 V DC, 13-26 mA
Dòng ở chế độ chờ:	10-13 mA
Dòng ở chế độ nghỉ:	< 80 uA
Tần số sóng mang:	13,56 MHz
Khoảng cách hoạt động:	0~60 mm (mifare card)
Giao thức truyền thông:	SPI
Tốc độ truyền dữ liệu:	< 10 Mbit/s
Kích thước:	40x60 mm
Các loại card hỗ trợ:	Mifare 1 S50, Mifare 1 S70, Mifare Ultra Light, Mifare Pro, Mifare Desfire



c) Thẻ RFID S50 13,56MHz

Thẻ RFID S50 13.56 MHz (thẻ NFC) dùng để đọc/ghi dữ liệu với tốc độ cao . Thẻ được ứng dụng trong nhiều lĩnh vực như bảo mật, nhà thông minh, chấm công, gửi xe thông minh, v.v... . Thẻ tương thích với module RFID RC522 13.56MHz và các module tương tự .

Chip:	Mifare 1 S50
Bộ nhớ:	8 Kb, 16 phân vùng
Tần số hoạt động:	13,56 MHz
Khoảng cách đọc:	2,5~10 mm

Thời gian đọc:	1~2 ms
Độ bền:	100.000 lần
Thời gian lưu trữ dữ liệu:	10 năm
Kích thước:	Tùy từng loại



2.5.. Màn hình LCD 20x4

Màn hình LCD 1602 xanh lá sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến, nhiều code mẫu và dễ dàng sử dụng hơn nếu đi kèm mạch chuyển tiếp I2C

THÔNG SỐ KỸ THUẬT

- Điện áp hoạt động là 5V.
- Kích thước: 80 x 36 x 12.5mm
- Chữ trắng, nền xanh dương
- Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
- Tên các chân được ghi ở mặt sau của màn hình LCD hỗ trợ việc kết nối, đi dây điện.
- Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu
- Có bộ ký tự được xây dựng hỗ trợ tiếng Anh và tiếng Nhật, xem thêm HD44780 datasheet để biết thêm chi tiết.



2.6. Cảm biến chuyển động HC-SR501

Cảm biến thân nhiệt chuyển động PIR (Passive infrared sensor) HC-SR501 được sử dụng để phát hiện chuyển động của các vật thể phát ra bức xạ hồng ngoại (con người, con vật, các vật phát nhiệt,...), cảm biến có thể chỉnh được độ nhạy để giới hạn khoảng cách bắt xa gần cũng như cường độ bức xạ của vật thể mong muốn, ngoài ra cảm biến còn có thể điều chỉnh thời gian kích trễ (giữ tín hiệu bao lâu sau khi kích hoạt) qua biến trở tích hợp sẵn.

THÔNG SỐ CẢM BIẾN THÂN NHIỆT CHUYỂN ĐỘNG PIR

- Phạm vi phát hiện: góc 360 độ hình nón, độ xa tối đa 6m.
- Nhiệt độ hoạt động: 32-122 ° F (050 ° C)
- Điện áp hoạt động: DC 3.8V – 5V
- Mức tiêu thụ dòng: $\leq 50 \mu\text{A}$
- Thời gian báo: 30 giây có thể tùy chỉnh bằng biến trở.
- Độ nhạy có thể điều chỉnh bằng biến trở.
- Kích thước: 1,27 x 0,96 x 1.0 (32,2 x 24,3 x 25,4 mm)



2.7. Còi chip 5v

Loa Buzzer 5V (Còi chip, còi bíp) là sản phẩm còi báo thường được sử dụng trong các mạch điện tử, được thiết kế nhỏ gọn, chân cắm thích hợp sử dụng báo động, báo hiệu âm thanh cho tín hiệu.

Thông số kỹ thuật Loa Buzzer 5V

- Nguồn : 3.5V – 5.5V
- Dòng điện tiêu thụ: <25mA
- Tần số cộng hưởng: 2300Hz \pm 500Hz
- Biên độ âm thanh: >80 dB
- Nhiệt độ hoạt động: -20 °C đến +70 °C
- Kích thước : Đường kính 12mm, cao 9,7mm



2.8. Đèn led

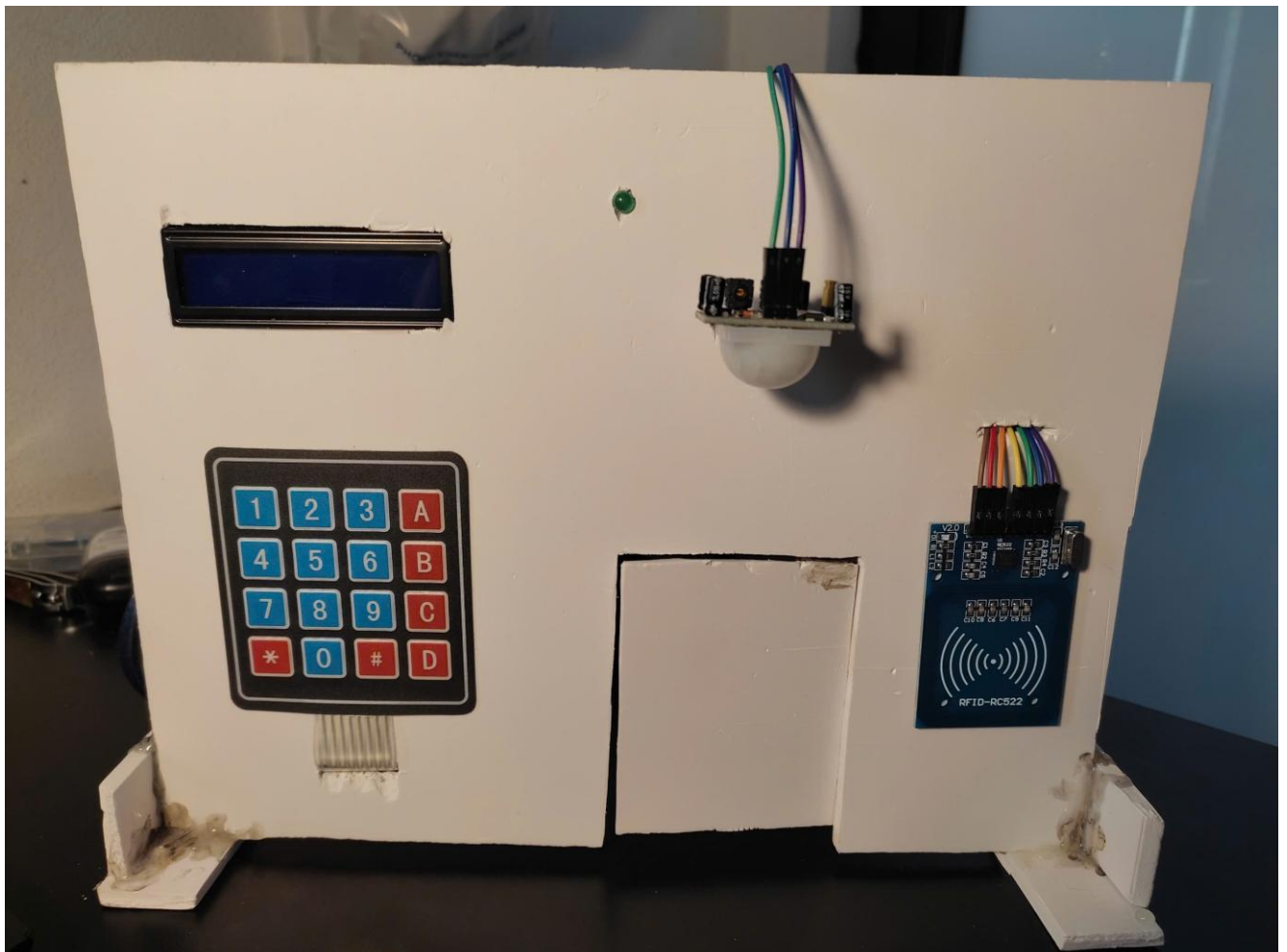
Dùng để làm đèn soi đường(tự động bật khi trời tối, tắt khi trời sáng)

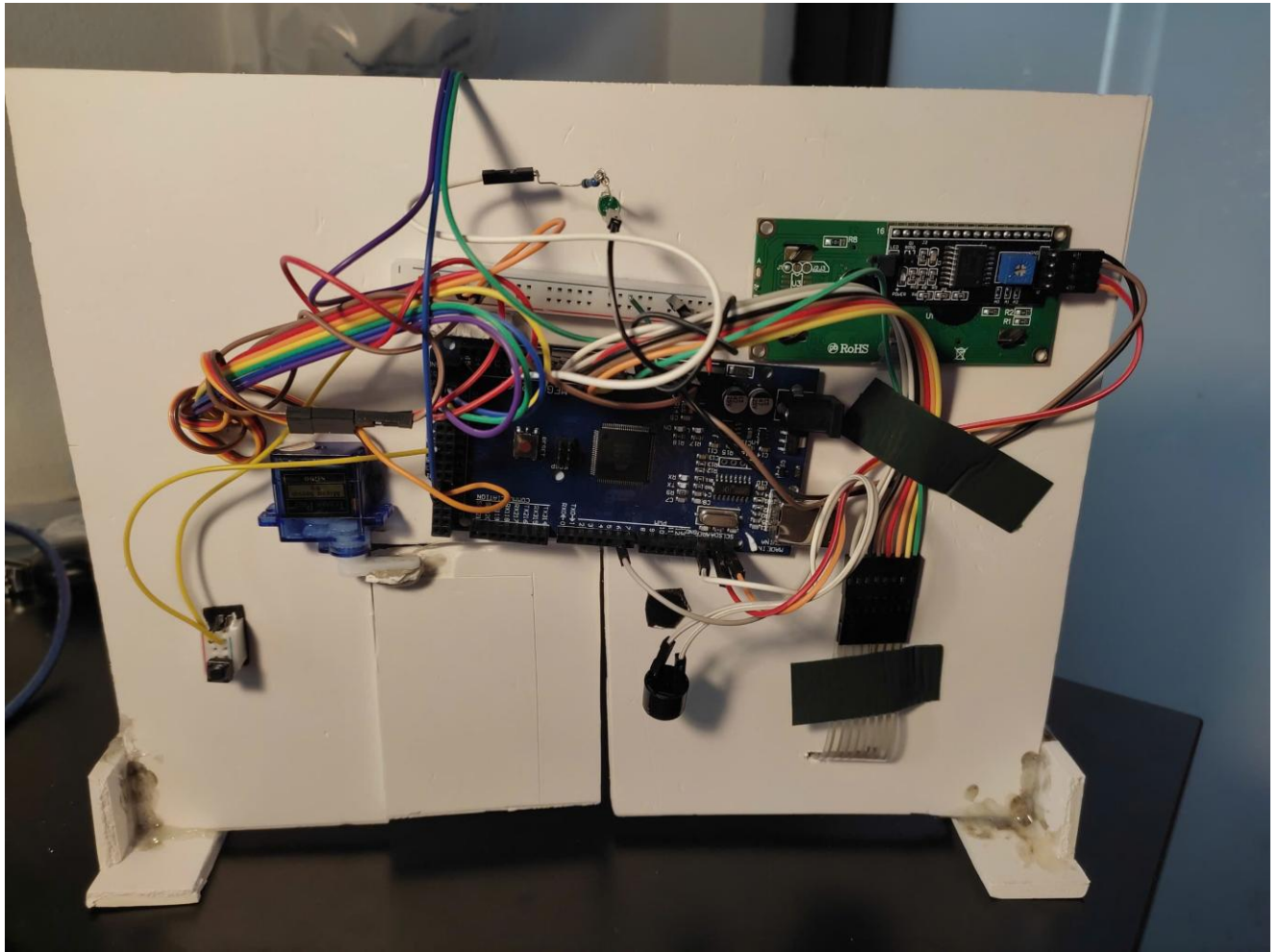
THÔNG SỐ, ĐẶC ĐIỂM KỸ THUẬT:

- Cân nặng: 0.3g
- Vỏ trong
- Màu sắc: trắng
- Đường kính: 5mm



CHƯƠNG 3. SƠ ĐỒ CHỨC NĂNG





CHƯƠNG 4. NGUYÊN LÝ HOẠT ĐỘNG TỪNG THÀNH PHẦN

4.1 Thư viện và khai báo biến

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Keypad.h>
#include <SPI.h>
#include <MFRC522.h>
#include <Servo.h>
#include <EEPROM.h>

// Định nghĩa các chân cho Keypad
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
byte rowPins[ROWS] = {A0, A1, A2, A3};
byte colPins[COLS] = {A4, A5, A6, A7};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// Đối tượng LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);

// Định nghĩa các chân và hằng số
#define RST_PIN      49
#define SS_PIN       53
#define MAX_UIDS     10
```



```

#define UID_LENGTH    4
#define BUZZER_PIN    6
#define LED_GREEN_PIN A10
#define SERVO_PIN     29
#define PIR_PIN       31
#define BUTTON_PIN    39

// EEPROM addresses
#define EEPROM_START_ADDRESS 0
#define UID_TOTAL_ADDRESS 100
#define PASSWORD_ADDRESS 150

// Mã bí mật để đổi mật khẩu
#define SECRET_CODE "9999"

// RFID và Servo
MFRC522 rfid(SS_PIN, RST_PIN);
Servo myServo;

// Các biến toàn cục
String inputPassword = "";
String correctPassword = "1234"; // Mật khẩu mặc định
int servoAngle = 0;
int wrongAttempts = 0;
const String masterUID = "24F9DABA";
bool isMasterMode = false;
bool isChangingPassword = false;

// Mảng lưu trữ các UID được phép
String authorizedUIDs[MAX_UIDS];
int totalUIDs = 0;

```

4.2. Hàm setup()

Chức năng:

Khởi tạo LCD, RFID, Servo, các chân IO, và tải dữ liệu từ EEPROM.

Code:

```

void setup() {
    // Khởi tạo LCD
    lcd.init();
    lcd.backlight();
    lcd.print("Initializing...");

    // Khởi tạo các chân IO
    pinMode(BUTTON_PIN, INPUT_PULLUP);
    pinMode(BUZZER_PIN, OUTPUT);
    pinMode(LED_GREEN_PIN, OUTPUT);
    pinMode(PIR_PIN, INPUT);

    // Khởi tạo SPI và RFID
    SPI.begin();
    rfid.PCD_Init();

    // Khởi tạo Servo
    myServo.attach(SERVO_PIN);
    myServo.write(0);

    // Đọc dữ liệu từ EEPROM
    loadAuthorizedUIDs();
    loadSavedPassword();
}

```

```

Serial.begin(9600);
lcd.clear();
lcd.print("System Ready");
delay(1000);
}

```

4.3. Hàm loop()

Chức năng:

Quản lý chế độ chính của hệ thống:

- **Master Mode:** Quản lý thẻ RFID.
- **Normal Mode:** Điều khiển hoạt động bình thường.

Code:

```

void loop() {
  if (isMasterMode) {
    handleMasterMode();
  } else {
    handleNormalMode();
  }
}

```

4.4. Master Mode (handleMasterMode)

Chức năng:

Cho phép thêm hoặc xóa thẻ RFID qua Keypad.

Code:

```

void handleMasterMode() {
  static bool waitingForCard = false;
  static char lastCommand = '\0';

  if (!waitingForCard) {
    lcd.clear();
    lcd.print("Master Mode:");
    lcd.setCursor(0, 1);
    lcd.print("A:Add B:Del #:Exit");
  }

  char key = keypad.getKey();
  if (key && !waitingForCard) {
    switch (key) {
      case 'A':
        waitingForCard = true;
        lastCommand = 'A';
        lcd.clear();
        lcd.print("Scan new card");
        break;

      case 'B':
        waitingForCard = true;
        lastCommand = 'B';
        lcd.clear();
        lcd.print("Scan to delete");
        break;

      case '#':
        isMasterMode = false;
        lcd.clear();
        lcd.print("Normal Mode");
        delay(1000);
        break;
    }
  }
}

```

```

}

if (waitingForCard) {
  if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
    String cardUID = "";
    for (byte i = 0; i < rfid.uid.size; i++) {
      cardUID += String(rfid.uid.uidByte[i], HEX);
    }
    cardUID.toUpperCase();

    if (lastCommand == 'A') {
      addUID(cardUID);
    } else if (lastCommand == 'B') {
      removeUID(cardUID);
    }

    waitingForCard = false;
    lastCommand = '\0';

    rfid.PICC_HaltA();
    rfid.PCD_StopCrypto1();

    delay(1000);
  }
}
}

```

4.5. Normal Mode (handleNormalMode)

Chức năng:

Quản lý hoạt động bình thường: kiểm tra nút nhấn, mật khẩu, và thẻ RFID.

Code:

```

void handleNormalMode() {
  if (digitalRead(BUTTON_PIN) == LOW) {
    openDoor();
    delay(500);
  }

  char key = keypad.getKey();
  if (key) {
    handleKeypad(key);
  }

  if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) {
    String rfidUID = "";
    for (byte i = 0; i < rfid.uid.size; i++) {
      rfidUID += String(rfid.uid.uidByte[i], HEX);
    }
    rfidUID.toUpperCase();

    Serial.print("Card detected: ");
    Serial.println(rfidUID);

    if (rfidUID == masterUID) {
      isMasterMode = true;
      tone(BUZZER_PIN, 800, 200);
      lcd.clear();
      lcd.print("Master Mode");
      delay(1000);
    } else {
      bool authorized = false;

```

```

for (int i = 0; i < totalUIDs; i++) {
  if (rfidUID == authorizedUIDs[i]) {
    authorized = true;
    break;
  }
}

if (authorized) {
  openDoor();
} else {
  lcd.clear();
  lcd.print("Access Denied");
  tone(BUZZER_PIN, 1000, 1000);
  delay(2000);
}
}

rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
}
}

```

4.6 OpenDoor

Chức năng chính của hàm này là mở cửa khi phát hiện điều kiện truy cập hợp lệ. Sau khi cửa mở, hệ thống sẽ giám sát chuyển động để tự động đóng cửa.

Code:

```

void openDoor() {
  lcd.clear();
  lcd.print("Access Granted");
  lcd.setCursor(0, 1);
  lcd.print("Welcome!");

  tone(BUZZER_PIN, 500);
  digitalWrite(LED_GREEN_PIN, HIGH);
  myServo.write(90); // Mở cửa bằng servo
  delay(1500);
  noTone(BUZZER_PIN);

  unsigned long startTime = millis();
  bool motionDetected = false;

  // Giám sát cảm biến chuyển động trong 5 giây
  while (millis() - startTime < 5000) {
    if (digitalRead(PIR_PIN) == HIGH) {
      motionDetected = true;
      digitalWrite(LED_GREEN_PIN, HIGH);
      lcd.clear();
      lcd.print("Motion detected!");
      startTime = millis(); // Reset thời gian
    }
    delay(100);
  }

  // Đóng cửa sau khi không phát hiện chuyển động
  myServo.write(0);
  digitalWrite(LED_GREEN_PIN, LOW);
  delay(1000);
}

```

4.7. Quản lý thẻ

* *Hàm loadAuthorizedUIDs:*

Hàm này tải danh sách UID thẻ được lưu trong EEPROM, đảm bảo các thẻ hợp lệ vẫn được nhận diện sau khi hệ thống khởi động lại.

Code:

```
void loadAuthorizedUIDs() {
    totalUIDs = EEPROM.read(UID_TOTAL_ADDRESS);
    if (totalUIDs > MAX_UIDS) totalUIDs = 0;

    for (int i = 0; i < totalUIDs; i++) {
        String uid = "";
        for (int j = 0; j < UID_LENGTH * 2; j++) {
            char c = EEPROM.read(EEPROM_START_ADDRESS + (i * UID_LENGTH * 2) + j);
            if (c != 0) uid += c;
        }
        authorizedUIDs[i] = uid;
    }
}
```

* *Hàm saveAuthorizedUIDs:*

Hàm này lưu danh sách UID thẻ hợp lệ vào EEPROM, cho phép hệ thống lưu trữ dữ liệu một cách bền vững.

Code:

```
void saveAuthorizedUIDs() {
    EEPROM.write(UID_TOTAL_ADDRESS, totalUIDs);
    for (int i = 0; i < totalUIDs; i++) {
        String uid = authorizedUIDs[i];
        for (int j = 0; j < uid.length(); j++) {
            EEPROM.write(EEPROM_START_ADDRESS + (i * UID_LENGTH * 2) + j, uid[j]);
        }
    }
}
```

* *Hàm addUID*

Hàm này thêm một thẻ mới vào danh sách UID hợp lệ nếu chưa tồn tại trong hệ thống.

Code:

```
void addUID(String uid) {
    for (int i = 0; i < totalUIDs; i++) {
        if (authorizedUIDs[i] == uid) {
            lcd.clear();
            lcd.print("Card exists!");
            tone(BUZZER_PIN, 400, 500);
            delay(1000);
            return;
        }
    }

    if (totalUIDs < MAX_UIDS) {
        authorizedUIDs[totalUIDs++] = uid;
        saveAuthorizedUIDs();
        lcd.clear();
    }
}
```

```

    lcd.print("Card Added!");
    tone(BUZZER_PIN, 800, 200);
    delay(1000);
} else {
    lcd.clear();
    lcd.print("Memory Full!");
    tone(BUZZER_PIN, 400, 500);
    delay(1000);
}
}
}

```

* **Hàm removeUID**

Hàm này xóa một thẻ khỏi danh sách UID hợp lệ nếu nó tồn tại trong hệ thống.

Code:

```

void removeUID(String uid) {
    bool found = false;
    for (int i = 0; i < totalUIDs; i++) {
        if (authorizedUIDs[i] == uid) {
            for (int j = i; j < totalUIDs - 1; j++) {
                authorizedUIDs[j] = authorizedUIDs[j + 1];
            }
            totalUIDs--;
            saveAuthorizedUIDs();
            found = true;
            break;
        }
    }

    lcd.clear();
    if (found) {
        lcd.print("Card Removed!");
        tone(BUZZER_PIN, 800, 200);
    } else {
        lcd.print("Card Not Found!");
        tone(BUZZER_PIN, 400, 500);
    }
    delay(1000);
}

```

4.8. Quản lý mật khẩu

Hàm saveNewPassword và loadSavedPassword

Hai hàm này hỗ trợ lưu và tải mật khẩu từ EEPROM.

Code:

```
void saveNewPassword(String newPassword) {
    EEPROM.write(PASSWORD_ADDRESS, newPassword.length());
    for (int i = 0; i < newPassword.length(); i++) {
        EEPROM.write(PASSWORD_ADDRESS + 1 + i, newPassword[i]);
    }
}

void loadSavedPassword() {
    int passLength = EEPROM.read(PASSWORD_ADDRESS);
    if (passLength > 0 && passLength < 20) {
        String savedPassword = "";
        for (int i = 0; i < passLength; i++) {
            savedPassword += char(EEPROM.read(PASSWORD_ADDRESS + 1 + i));
        }
        if (savedPassword.length() > 0) {
            correctPassword = savedPassword;
        }
    }
}
```

KẾT LUẬN

Trong bài báo cáo này, chúng tôi đã tìm hiểu và thực hiện thiết kế một hệ thống cửa tự động thông minh với khả năng bảo mật cao, linh hoạt trong quản lý và dễ sử dụng. Hệ thống được xây dựng dựa trên vi điều khiển Arduino Mega 2560, kết hợp các module cảm biến và thiết bị ngoại vi như RFID, Keypad, màn hình LCD, cảm biến PIR và Servo. Các nội dung chính đã được trình bày bao gồm:

- **Phân tích chức năng:** Hệ thống hỗ trợ hai chế độ hoạt động chính là *Master Mode* để quản lý truy cập và *Normal Mode* để vận hành thông thường.
- **Triển khai mã nguồn:** Toàn bộ mã nguồn được tối ưu hóa nhằm đảm bảo hoạt động ổn định, dễ bảo trì và mở rộng.
- **Cơ chế bảo mật:** Hệ thống hỗ trợ mật khẩu có thể thay đổi, quản lý thẻ RFID qua EEPROM, và phát hiện chuyển động qua cảm biến PIR để đảm bảo an toàn tối đa.

Hệ thống đạt được các yêu cầu thiết kế ban đầu và có khả năng ứng dụng thực tiễn trong việc kiểm soát truy cập cho nhà ở, văn phòng hoặc cơ sở kinh doanh. Trong tương lai, việc tích hợp thêm các tính năng kết nối không dây như Wi-Fi hoặc Bluetooth để giám sát và điều khiển từ xa sẽ là một hướng phát triển tiềm năng nhằm cải thiện trải nghiệm người dùng và mở rộng phạm vi ứng dụng của hệ thống.