

**TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM**

**KHOA CÔNG NGHỆ ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN TỰ ĐỘNG HÓA**

----🙣🕮🙡----

****

**ĐỒ ÁN CHUYÊN NGÀNH**

**Mô hình két sắt mở khóa bằng RFID và passcode**

**GVHD: ThS. TRẦN HOÀN**

**SVTH: Lý Hoàng Nam**

**MSSV: 2032181068**

**TP. HỒ CHÍ MINH, tháng 09 năm 2021**

**TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM**

**KHOA CÔNG NGHỆ ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN TỰ ĐỘNG HÓA**

**----🙣🕮🙡----**

****

**ĐỒ ÁN CHUYÊN NGÀNH**

**Mô hình két sắt mở khóa bằng RFID và passcode**

**GVHD: ThS. TRẦN HOÀN**

**SVTH: Lý Hoàng Nam**

**MSSV: 2032181068**

**TP. HỒ CHÍ MINH, tháng 09 năm 2021**

|  |  |
| --- | --- |
| TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM  KHOA CN ĐIỆN – ĐIỆN TỬ  BỘ MÔN: TỰ ĐỘNG HÓA | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |
|  | *TP. HCM, ngày 17 tháng 09 năm 2021* |

**NHẬN XÉT ĐỒ ÁN CHUYÊN NGÀNH**

**CỦA GIẢNG VIÊN HƯỚNG DẪN**

|  |  |  |
| --- | --- | --- |
| **Tên đồ án:** | | |
| **MÔ HÌNH KÉT SẮT MỞ KHÓA BẰNG RFID VÀ PASSCODE** | | |
| **Sinh viên thực hiện:** | | **Giảng viên hướng dẫn:** |
| Lý Hoàng Nam | 2032181068 | ThS. TRẦN HOÀN |
| **Đánh giá Đồ án**   1. Về cuốn báo cáo:   Số trang 34 Số chương 4  Số bảng số liệu 1 Số hình vẽ 12  Số tài liệu tham khảo 6 Sản phẩm 1  Một số nhận xét về hình thức cuốn báo cáo:   1. Về nội dung đồ án: 2. Về tính ứng dụng: 3. Về thái độ làm việc của sinh viên:   **Đánh giá chung:**  **Điểm từng sinh viên:**  Lý Hoàng Nam:…..**/10** | | |

|  |  |
| --- | --- |
|  | **Người nhận xét**  (Ký tên và ghi rõ họ tên) |

**LỜI CÁM ƠN**

Thông qua đồ án này em xin bày tỏ lòng biết ơn sâu sắc gửi đến quý thầy cô khoa Điện – Điện Tử của Trường Đại học Công Nghiệp Thực Phẩm Tp.HCM và đặc biệt tới thầy Trần Hoàn đã tạo điều kiện cho em có thể nghiêm cứu và hoàn thành được đồ án chuyên ngành này. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì em nghĩ bài báo cáo này của em rất khó có thể hoàn thiện được. Một lần nữa, em xin chân thành cảm ơn thầy đã giúp đỡ em trong quá trình làm đồ án. Bước đầu đi vào thực tế, tìm hiểu về lĩnh vực sáng tạo trong nghiên cứu khoa học, kiến thức của em còn hạn chế và còn nhiều bỡ ngỡ. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô và các bạn học cùng lớp để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Em xin chân thành cảm ơn thầy!

TP. Hồ Chí Minh, ngày 17 tháng 09 năm 2021

Tác giả

***Lý Hoàng Nam***

|  |  |
| --- | --- |
| TRƯỜNG ĐH CÔNG NGHIỆP THỰC PHẨM TP. HCM  KHOA CN ĐIỆN – ĐIỆN TỬ  BỘ MÔN: TỰ ĐỘNG HÓA | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập - Tự do - Hạnh phúc |
|  | *TP. HCM, ngày 17 tháng 09 năm 2021.* |

**ĐỀ CƯƠNG CHI TIẾT**

|  |  |
| --- | --- |
| **TÊN ĐỒ ÁN: MÔ HÌNH KÉT SẮT MỞ KHÓA BẰNG RFID VÀ PASSCODE** | |
| **Giảng viên hướng dẫn:ThS. TRẦN HOÀN** | |
| **Thời gian thực hiện:** Từ ngày 24/3/2021 đến ngày 24/6/2021 | |
| **Sinh viên thực hiện: Lý Hoàng Nam** | |
| **Nội dung đề tài:**   * Sử dụng mạch RFID và KEYPAD để mở, khoá két sắt * Hiển thị LCD | |
| **Kế hoạch thực hiện:**   * Từ ngày 24/3/2021 đến ngày 26/3/2021 : thực hiện nhận đề tài * Từ ngày 27/3/2021 đến ngày 4/4/2021: thực hiện chuẩn bị linh kiện * Từ ngày 5/4/2021 đến ngày 25/4/2021: thực hiện xây dựng phần cứng * Từ ngày 26/4/2021 đến ngày 9/6/2021: thực hiện soạn thảo code * Từ ngày 10/6/2020 đến ngày 24/6/2021: thực hiện chuẩn bị báo cáo | |
| **Xác nhận của giảng viên hướng dẫn** | TP. HCM, ngày 12 tháng 06 năm 2021  **Sinh viên**  **Lý Hoàng Nam** |

# MỤC LỤC

[MỤC LỤC i](#_Toc82717187)

[DANH MỤC KÝ HIỆU, CỤM TỪ VIẾT TẮT iii](#_Toc82717188)

[DANH MỤC BẢNG BIỂU iv](#_Toc82717189)

[DANH MỤC HÌNH ẢNH v](#_Toc82717190)

[CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc82717191)

[1.1 Đặt vấn đề 1](#_Toc82717192)

[1.2 Mục tiêu đề tài 1](#_Toc82717193)

[1.3 Phương pháp nghiên cứu 1](#_Toc82717194)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 2](#_Toc82717195)

[2.1 Tổng quan về Arduino 2](#_Toc82717196)

[2.1.1 Arduino 2](#_Toc82717197)

[2.1.2 IC Atmega 328 4](#_Toc82717198)

[2.1.3 Sơ đồ chân Arduino Uno R3 6](#_Toc82717199)

[2.1.3.1 Chân cấp nguồn cho cảm biến: 7](#_Toc82717200)

[2.1.3.2 Chân tín hiệu Analog: 8](#_Toc82717201)

[2.1.3.3 Các chân đọc và xuất tín hiệu digital: 8](#_Toc82717202)

[2.2 Tổng quát về LCD 1602 và module I2C 9](#_Toc82717203)

[2.2.1 LCD 1602 9](#_Toc82717204)

[2.2.2 Module I2C 11](#_Toc82717205)

[2.3 Giới thiệu về module RELAY 1 kênh 5V 13](#_Toc82717206)

[2.3.1 RELAY 5V 13](#_Toc82717207)

[2.3.2 Tính năng 13](#_Toc82717208)

[2.3.3 Thông số kĩ thuật 13](#_Toc82717209)

[2.4 Giới thiệu về khoá điện DC 12V 14](#_Toc82717210)

[2.4.1 Khóa điện DC 12V 14](#_Toc82717211)

[2.4.2 Tính năng 14](#_Toc82717212)

[2.4.3 Thông số kỹ thuật 15](#_Toc82717213)

[2.5 Giới thiệu bàn phím ma trận Keypad 4X4 16](#_Toc82717214)

[2.5.1 Keypad 4X4 16](#_Toc82717215)

[2.5.2 Cấu tạo và nguyên lý hoạt động 17](#_Toc82717216)

[2.6 Giới thiệu về module RFID RC522 20](#_Toc82717217)

[2.6.1 RFID RC522 20](#_Toc82717218)

[2.6.2 Thông số kĩ thuật: 20](#_Toc82717219)

[2.6.3 Chân kết nối 21](#_Toc82717220)

[CHƯƠNG 3: CƠ SỞ THỰC HIỆN 22](#_Toc82717221)

[3.1 Sơ đồ kết nối của mô hình 22](#_Toc82717222)

[3.2 Lưu đồ giải thuật 23](#_Toc82717223)

[CHƯƠNG 4: TỔNG KẾT VÀ ĐỊNH HƯỚNG ĐỀ TÀI 24](#_Toc82717224)

[4.1 Kết quả đạt được 24](#_Toc82717225)

[4.2 Hướng phát triển đề tài 25](#_Toc82717226)

[PHỤ LỤC 27](#_Toc82717227)

[Code chương trình 27](#_Toc82717228)

# DANH MỤC KÝ HIỆU, CỤM TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **KÝ HIỆU** | **THUẬT NGỮ** |
| RFID | Radio Frequency Identification |
| GND | Ground |
| SRAM | Static Random Access Memory |
| EEPROM | Electrically Eraseble Programmble Read Only Memory |
| ICSP | In – Cỉuit Serial Programming |
| DC | Điện áp một chiều |
| VĐK | Vi điều khiển |
| RFID | Radio Frequency Identification |

# DANH MỤC BẢNG BIỂU

[Bảng 2.1 Bảng giao tiếp I2C LCD Arduino 13](#_Toc82601179)

# DANH MỤC HÌNH ẢNH

[Hình 2.1 Mạch Arduino Uno R3 2](#_Toc82609019)

[Hình 2.2 IC Atmega 328 4](#_Toc82609020)

[Hình 2.3 Sơ đồ chân Arduino Uno R3 6](#_Toc82609021)

[Hình 2.4 Màn hình LCD 1602 9](#_Toc82609022)

[Hình 2.5 Module I2C LCD 16X2 11](#_Toc82609023)

[Hình 2.6 Module Relay 1 kênh 5V 13](#_Toc82609024)

[Hình 2.7 Khoá điện DC 12V 14](#_Toc82609025)

[Hình 2.8 Keypad 4x4 16](#_Toc82609026)

[Hình 2.9 Keypad 4x4 trong Proteus 17](#_Toc82609027)

[Hình 2.10 Cấu tạo của keypad 4x4 17](#_Toc82609028)

[Hình 2.11 Mô hình Keypad áp dụng thuật toán “Quét phím” 18](#_Toc82609029)

[Hình 2.12 Module RFID RC522 20](#_Toc82609030)

[Hình 3.1 Sơ đồ kết nối của mô hình 22](#_Toc82609031)

[Hình 4.1 Mặt trước mô hình 24](#_Toc82609032)

[Hình 4.2 Mặt sau mô hình 25](#_Toc82609033)

# CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

## Đặt vấn đề

Ngày nay với sự phát triển của xã hội hiện đại ai trong số chúng ta cũng cần có thiết bị bảo vệ tài sản trong nhà như khóa cửa, khoá tủ, thiết bị cảnh báo chống trộm hay camera nhưng có lẽ thiết bị được sử dụng nhiều nhất vẫn chính là khoá cửa. Hiện nay trên có rất nhiều loại khoá cửa nhưng hầu hết là khoá cơ khí, các khoá cơ khí này gặp vấn đề lớn đó là tính bảo mật không cao dễ bị phá bởi các chìa khoá đa năng.

Đa số các loại khoá kỹ thuật số đang có bán trên thị trường thường có giá khá cao và chủ yếu là loại khoá tay nắm ta thường thấy trong khách sạn hay chung cư hoặc khoá sử dụng phương pháp cài đặt mã số để khoá hoặc mở và người sử dụng có thể cài đặt số bất kỳ. Bên cạnh loại chỉ có chức năng khoá bằng số thì có thể dùng thẻ để mở. Nếu như bạn lỡ quên mã số thì có thể dùng thẻ để mở khoá.

## Mục tiêu đề tài

Để nâng cao yêu cầu về tính bảo mật để bảo vệ tài sản và giao diện dễ sử dụng. Em đã quyết định thực hiện đề tài mô hình két sắt mở khóa bằng RFID và passcode.Sử dụng thẻ RFID và Keypad để mở khóa cửa.

## Phương pháp nghiên cứu

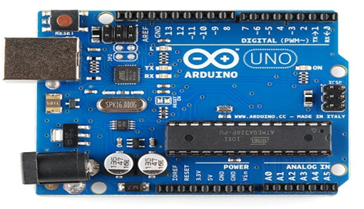
Tham khảo tài liệu: tài liệu tham khảo chủ yếu qua nguồn internet.

Tự nghiên cứu: nghiên cứu từ tài liệu được thu thập qua nguồn internet, quá trình thực hiện đồ án phải trải qua việc nghiên cứu các nguồn tài liệu, chắt lọc những thông tin có giá trị, đồng thời tự suy nghĩ tìm ra cách giải quyết vấn đề.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1 Tổng quan về Arduino

### 2.1.1 Arduino



Hình 2.1 Mạch Arduino Uno R3

Arduino một nền tảng [mã nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phần cứng và phần mềm. Phần cứng Arduino (các board mạch vi xử lý) được sinh ra tại thị trấn Ivrea ở Ý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. 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.

Được giới thiệu vào năm 2005, những nhà thiết kế của Arduino cố gắng mang đến một phương thức dễ dàng, không tốn kém cho những người yêu thích, sinh viên và giới chuyên nghiệp để tạo ra những thiết bị có khả năng tương tác với môi trường thông qua các [cảm biến](https://vi.wikipedia.org/wiki/C%E1%BA%A3m_bi%E1%BA%BFn) và các cơ cấu chấp hành. Những ví dụ phổ biến cho những người yêu thích mới bắt đầu bao gồm các robot đơn giản, điều khiển nhiệt độ và phát hiện chuyển động. Đi cùng với nó là một môi trường phát triển tích hợp (IDE) chạy trên các [máy tính cá nhân](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_c%C3%A1_nh%C3%A2n) thông thường và cho phép người dùng viết các chương trình cho Aduino bằng ngôn ngữ C hoặc C++.

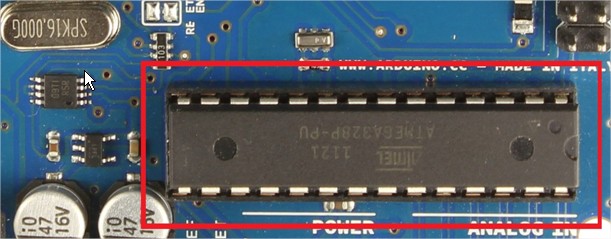
Giá của các board Arduino dao động xung quanh €20, hoặc $27, nếu được "làm giả" thì giá có thể giảm xuống thấp hơn $9. Các board Arduino có thể được đặt hàng ở dạng được lắp sẵn hoặc dưới dạng các kit tự-làm-lấy. Thông tin thiết kế phần cứng được cung cấp công khai để những ai muốn tự làm một mạch Arduino bằng tay có thể tự mình thực hiện được ([mã nguồn mở](https://vi.wikipedia.org/wiki/M%C3%A3_ngu%E1%BB%93n_m%E1%BB%9F)). Người ta ước tính khoảng giữa năm 2011 có trên 300 ngàn mạch Arduino chính thức đã được sản xuất thương mại, và vào năm 2013 có khoảng 700 ngàn mạch chính thức đã được đưa tới tay người dùng.

Arduino thật ra là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với người ít am hiểu về điện tử và lập trình. Và điều làm nên hiện tượng Arduino chính là mức giá rất thấp và tính chất nguồn mở từ phần cứng tới phần mềm.

Arduino là nền tảng giúp phát triển các ứng dụng vi điều khiển nhanh và thuận tiện.

Ứng dụng rộng rãi trong các thiết kế (cả nghiên cứu lẫn thương mại). Được hỗ trợ bởi cộng đồng lớn.

### 2.1.2 IC Atmega 328

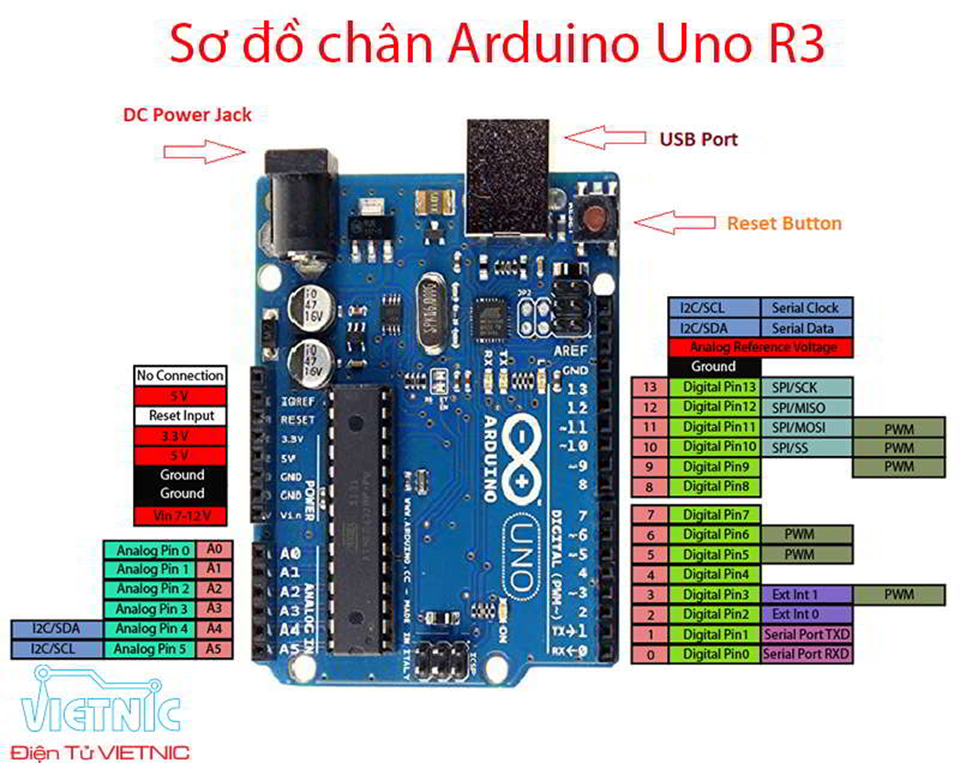


Hình 2.2 IC Atmega 328

IC Atmega 328 là phần quan trọng nhất của board Arduino UNO, IC này giống như một bộ não dùng để thu thập dữ liệu và xử lý chúng để truyền đi. Với 23 chân có thể sử dụng cho các kết nối vào và ra I/O, 32 thanh ghi, 3 bộ timer/couter có thể lập trình, có các ngắt nội và ngoại ( 2 lệnh trên 1 vector ngắt), giao thức truyền thông nối tiếp USART, SPI, I2C. Ngoài ra có thể sử dụng bộ biến đổi số tương tự 10 bit (ADC/DAC) mở rộng tới 8 kênh, khả năng lập trình được bộ giám sát thời gian hoạt động tới 5 chế độ nguồn; có thể sử dụng tới 6 kênh điều chế độ rộng xung (PWM), hỗ trợ nạp khởi động. ATmega328 có khả năng hoạt động trong 1 dải điện áp rộng (1.8V -5.5V), tốc độ thực thi ( thông lượng ) 1 MIPS trên 1MHz. Atmega328 là linh hồn của board mạch Arduino. Một số thông tin cơ bản của IC này:

* 32kB bộ nhớ Flash: những đoạn lệnh lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng hiếm khi nào cần quá 20KB bộ nhớ.
* 2KB cho SRAM (Static Random Access Memory): giá trị của biến khai báo khi lập trình sẽ được lưu ở đây. Nhưng khi bị mất điện dữ liệu trên SRAM sẽ bị mất.
* 1KB cho EEPROM (Electrically Eraseble Programmble Read Only Memory): đây giống như một chiếc ổ cứng mini, nơi có thể đọc và ghi dữ liệu vào đây mà không phải lo bị mất khi mất điện giống như dữ liệu trên SRAM.
* Nut Reset: nút này thường được sử dụng khi muốn chạy lại chương tình hay đôi khi chương trình bị lỗi chúng ta muốn reset lại chương trình đó.
* ICPS Atmega:ICSP (In – Cỉuit Serial Programming) là cá chân giao tiép của chíp Atmega 16U2. Các chân này thường được sử dụng tỏng các dự án lớn về Arduino.

### 2.1.3 Sơ đồ chân Arduino Uno R3



Hình 2.3 Sơ đồ chân Arduino Uno R3

Với 23 chân có thể sử dụng cho các kết nối vào và ra I/O, 32 thanh ghi, 3 bộ timer/couter có thể lập trình, có các ngắt nội và ngoại ( 2 lệnh trên 1 vector ngắt), giao thức truyền thông nối tiếp USART, SPI, I2C. Ngoài ra có thể sử dụng bộ biến đổi số tương tự 10 bit (ADC/DAC) mở rộng tới 8 kênh, khả năng lập trình được bộ giám sát thời gian hoạt động tới 5 chế độ nguồn; có thể sử dụng tới 6 kênh điều chế độ rộng xung (PWM), hỗ trợ nạp khởi động. ATmega328 có khả năng hoạt động trong 1 dải điện áp rộng (1.8V -5.5V), tốc độ thực thi ( thông lượng ) 1 MIPS trên 1MHz. Atmega328 là linh hồn của board mạch Arduino.

#### 2.1.3.1 Chân cấp nguồn cho cảm biến:

* GND (Ground): cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.
* 5V: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.
* 3.3V: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.
* Vin (Voltage Input): để cấp nguồn ngoài cho Arduino uno, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.
* IORREF: điện áp hoạt động của vi điều khiển trên Arduino uno có thể được đo ở chân này. Và dĩ nhiên nó luôn 5V. Mặc dù vậy nhưng không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.
* RESET: việc nhấn nút RESET trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10kΩ.

**Lưu ý:**

Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó phải hết sức cẩn thận, kiểm tra các cực âm – dương của nguồn trước khi cấp cho Arduino UNO. Việc làm chập mạch nguồn vào của Arduino UNO sẽ biến nó thành một miếng nhựa chặn giấy. Nên dùng nguồn từ cổng USB nếu có thể.

Các chân 3.3V và 5V trên Arduino là các chân dùng để cấp nguồn ra cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng board.

Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng board.

Cấp điện áp trên 13V vào chân RESET trên board có thể làm hỏng vi điều khiển ATmega328.

Cường độ dòng điện vào/ra ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển.

Cấp điệp áp trên 5.5V vào các chân Digital hoặc Analog của Arduino uno sẽ làm hỏng vi điều khiển.

Cường độ dòng điện qua một chân Digital hoặc Analog bất kì của Arduino uno vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó nếu không dùng để truyền nhận dữ liệu, phải mắc một điện trở hạn dòng.

#### 2.1.3.2 Chân tín hiệu Analog:

Arduino UNO có 6 chân analog (A0 🡒 A5) cung cấp độ phân giải tín hiệu 10bit (0 🡒 210 -1) để đọc giá trị điện áp trong khoảng 0V 🡒 5V). Với chân AREF trên board, có thể đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu cấp điện áp 2.5V vào chân này thì có thể dùng các chân analog để đo điện áp trong khoảng từ 0V 🡒 2.5V với độ phân giải vẫn là 10bit. Đặc biệt Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

### 2.1.3.3 Các chân đọc và xuất tín hiệu digital:

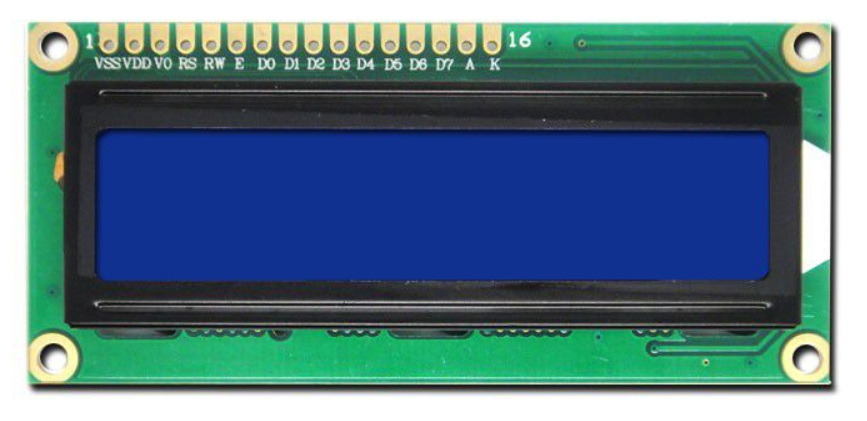
* Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu.chúng chỉ có 2 mức điện áp là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull – up từ được cài đặt ngay trong vi điều khiển Atmega328 (mặc định thì các điện trở này không được kết nối).

Một Số Chân Digital Có Các Chức Năng Đặc Biệt Như Sau:

* Chân serial: 0 (RX) và 1 (TX): dùng để gửi (trasmit – TX) và nhận (receive –RX) dữ liệu TTL Serial.Arduino UNO có thể giao tiếp với thiết bị khác thông qua 2 chân này.
* Chân PWM(Pulse Width Modulation)(~) 3, 5, 6, 9, 10, và 11: cho phép xuất ra xung PWM với độ phân giải 8bit (giá trị từ 0 → 28-1 tương ứng với 0V → 5V) bằng hàm analogWrite(). Nói một cách đơn giản, có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.
* Chân giao tiếp SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).  Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.

## 2.2 Tổng quát về LCD 1602 và module I2C

### 2.2.1 LCD 1602



Hình 2.4 Màn hình LCD 1602

Ngày nay thiết bị hiển thị LCD 1602 (Liquid Crystal Display) được sử dụng trong rất nhiều các ứng dụng VĐK. LCD 1602 rất nhiều ưu điểm so với các dạng hiển thị khác như: khả năng hiển thị kí tự đa dạng (chữ, số, kí tự đồ hoạ): dễ dàng đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác như, tiêu tốn rất ít tài nguyên hệ thống, giá thành rẻ,…

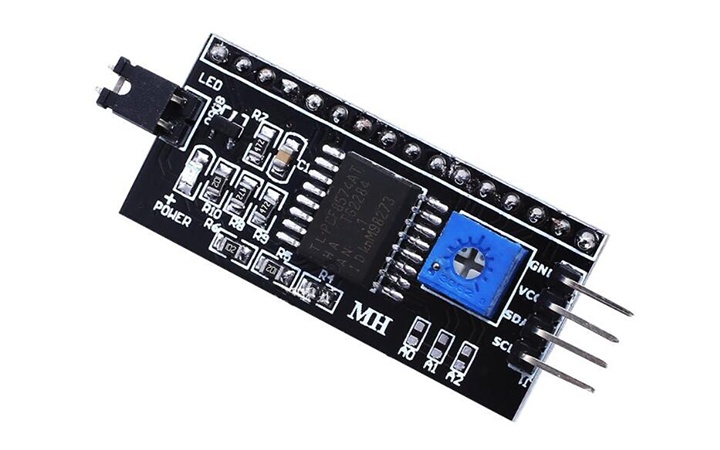
Thông Số Kĩ Thuật Của Sản Phẩm LCD 1620:

* Điện áp MAX: 7V
* Điện áp MIN: -0,3V
* Hoạt động ổn định: 2.7 – 5.5V
* Điện áp ra mức cao: >2.4V
* Điện áp ra mức thấp: <0.4V
* Dòng điện cấp nguồn: 350uA – 600 uA
* Nhiệt độ hoạt động: -30 – 75 độ C

Chức Năng Của Từng Chân LCD 1602:

* Chân số 1 – VSS: Chân nối đất cho LCD được nối với GND của mạch điều khiển.
* Chân số 2 – VDD: Chân cấp nguồn cho LCD, được nối với GND của mạch điều khiển.
* Chân số 3 – VE: Điều chỉnh độ tương phản của LCD.
* Chân số 4 – RS: Chân chọn thanh ghi, được nối với logic “0” hoặc logic “1”.
  + 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.
* Chân số 5 – R/W: Chân chọn chế dộ đọc/ghi (Read/Write), được nối với logic “0” để ghi hoặc nối với logic “1” đọc.
* Chân số 6 – E: Chân cho phép (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 này như sau:
  + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào thanh ghi bên trong 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.
* Chân số 7 đến D0 đến D7: 8 đường của bus dữ liệu dùng để trao đổi thông tin mới MPU. Có 2 chế độ sử dụng 8 đường bus này là: chế độ 8 bit(dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7) và chế độ 4 bit (dữ liệu được truyền trên 4 bit đường từ DB4 tới DB7, bit MSB là DB7).
* Chân số 15 – A: Nguồn dương cho đèn nền.
* Chân số 16 – k: nguồn âm cho đèn nền.

### 2.2.2 Module I2C



Hình 2.5 Module I2C LCD 16X2

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

Module I2C LCD ra đời và giải quyết vấn để này cho bạn.

Thay vì phải mất 6 chân vi điều khiển để kết nối với LCD 16x2 (RS, EN, D7, D6, D5 và D4) thì module IC2 bạn chỉ cần tốn 2 chân (SCL, SDA) để kết nối.

Module I2C hỗ trợ các loại LCD sử dụng driver HD44780(LCD 16x2, LCD 20x4, …) và tương thích với hầu hết các vi điều khiển hiện nay.

Ưu điểm:

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

Thông số kĩ thuật:

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

Các Lỗi Thường Gặp Khi Sử Dụng I2C LCD:

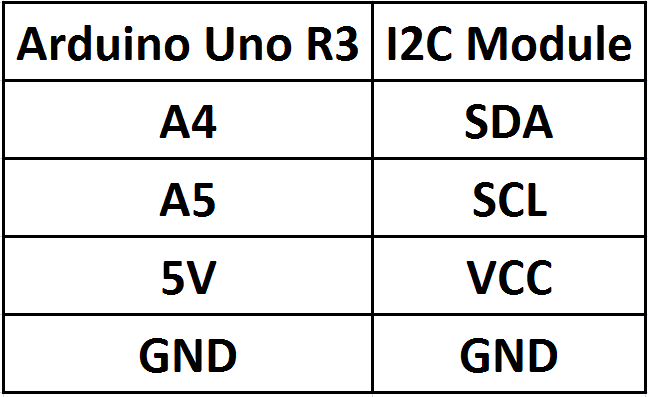
* Hiển thị một dãy ô vuông.
* Màn hình chỉ in ra một ký tự đầu.
* Màn hình nhấp nháy.

Các lỗi này chủ yếu là do sai địa chỉ bus, để fix lỗi các bạn thay địa chỉ mặc định là “0x27” thành “0x3F”.

Trong trường hợp vẫn không được các bạn fix lỗi bằng cách nạp code tìm địa chỉ bus của I2C.

Sau khi tìm xong các bạn thay địa chỉ vừa tìm được vào vị trí “0x27” là xong.

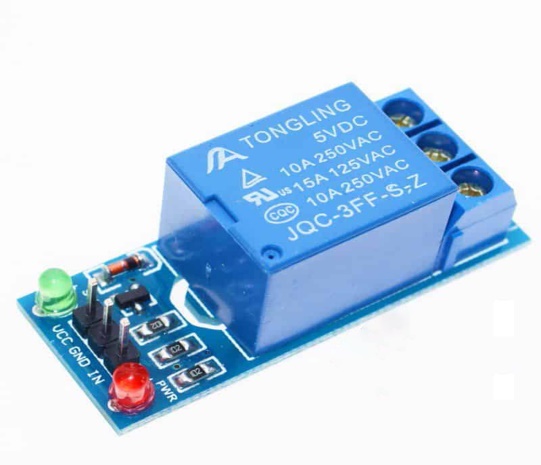
Giao tiếp I2C LCD Arduino:



Bảng 2.1 Bảng giao tiếp I2C LCD Arduino

## 2.3 Giới thiệu về module RELAY 1 kênh 5V

### 2.3.1 RELAY 5V



Hình 2.6 Module Relay 1 kênh 5V

### 2.3.2 Tính năng

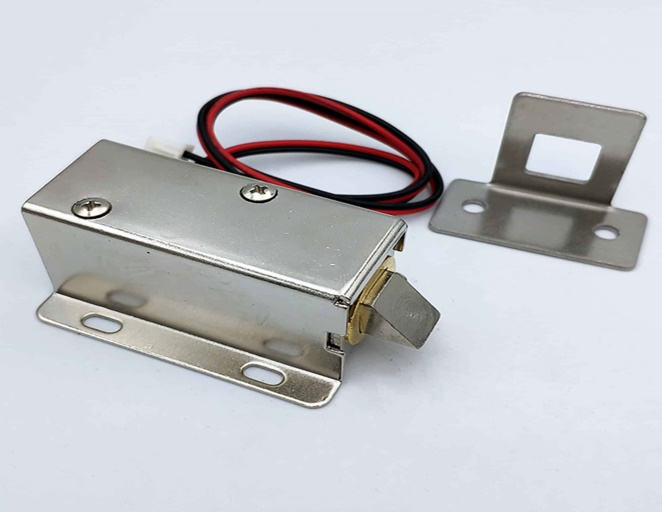
* Module Relay 1 Kênh 5V 10-15A Âm Kích được sử dụng để điều khiển thiết bị điện tử công suất cao như đèn, quạt điện và điều hòa không khí,…
* Module Relay 1 Kênh 5V 10-15A Âm Kích được sử dụng rộng rãi và phổ biến trong các hệ thống IoT, nhà thông minh, vườn thông minh,… để điều khiển thiết bị tắt mở một cách dễ dàng và nhanh chóng.

### 2.3.3 Thông số kĩ thuật

* Điện áp hoạt động: 5V DC.
* Tín hiệu kích: Low (0V).
* Tiếp điểm đóng ngắt: 250V-10A (AC) hoặc 30V-10A (DC)..
* Kích thước: 52 \* 41 \* 19 mm.
* Các chân của Module Relay 1 Kênh:
  + DC +: Kết nối cực dương của nguồn điện.
  + DC-: Kết nối cực âm của nguồn điện.
  + IN1: Theo cài đặt của người dùng, nó có thể ở mức cao hoặc thấp.
  + NO1: Cổng Thường Mở.
  + COM1: Cổng chung rơle.
  + NC1: Cổng Thường Đóng

## 2.4 Giới thiệu về khoá điện DC 12V

### 2.4.1 Khóa điện DC 12V



Hình 2.7 Khoá điện DC 12V

### 2.4.2 Tính năng

Khóa điện DC 12V 01 có chức năng hoạt động như một ổ khóa cửa sử dụng Solenoid để kích đóng mở bằng điện, được sử dụng nhiều trong nhà thông minh hoặc các loại tủ, cửa phòng… Khóa sử dụng điện áp 12 VDC là loại thường đóng với chất lượng tốt, độ bền cao.

### 2.4.3 Thông số kỹ thuật

* Chất liệu vỏ và chốt: Thép không gỉ
* Điện áp:12V
* Dòng điện khóa điện: 0.3A, 0.4A, 1.2A
* Công suất tiêu thụ 9.6w
* Sử dụng solenoid từ
* Tốc độ phản ứng <1s
* Thời gian kích liên tục <10s

## 2.5 Giới thiệu bàn phím ma trận Keypad 4X4

### 2.5.1 Keypad 4X4

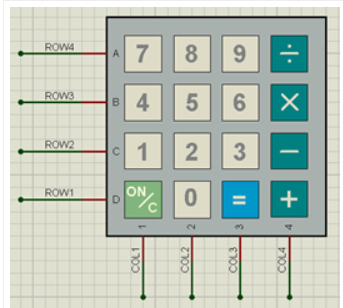


Hình 2.8 Keypad 4x4

Keypad là một thiết bị nhập chứa các nút bấm cho phép người dùng nhập các chữ số, chữ cái hoặc ký tự điều khiển. Keypad không chứa tất cả bảng mã ASCII như keyboard vì thế nó thường được sử dụng trong các ứng dụng chuyên dụng và tương đối đơn giản, ở đó, số lượng nút nhấn thay đổi phụ thuộc vào ứng dụng.

Ma trận bàn phím **4x4**bao gồm **16 phím bấm,** chúng được kết nối với nhau tạo thành **4 hàng, 4 cột**và cho ra **8 chân kết nối** (4 chân hàng ngang (Row: **R1-> R4),** và 4 chân cột dọc (Column: **C1-> C4).** Bình thường để đọc được phím từ 16 phím đơn, ta cần phải dùng đến 16 chân của VĐK, tuy nhiên cách mắc thành ma trận phím 4x4 dưới đây sẽ giúp ta rút gọn còn 8 chân kết nối.

Keypad 4×4 được coi là thiết bị đơn giản cho phép người dùng thực hiện thao tác nhập số. Do đó, Keypad 4×4 được sử dụng khá phổ biến, dưới đây là hình ảnh nguyên lý của Keypad trong phần mềm mô phỏng Proteus:

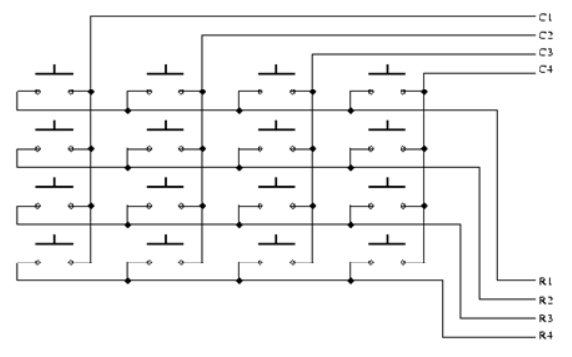


Hình 2.9 Keypad 4x4 trong Proteus

### 2.5.2 Cấu tạo và nguyên lý hoạt động

* **Cấu tạo**

KeyPad 4×4 gồm: 8 đầu vào/ra và 16 phím nhấn kết nối theo sơ đồ dưới đây:



Hình 2.10 Cấu tạo của keypad 4x4

Các phím bấm được chia thành 4 hàng và 4 cột, 1 đầu của nút bấm được nối với đầu vào cột(C), đầu kia được nối với đầu vào hàng(R).

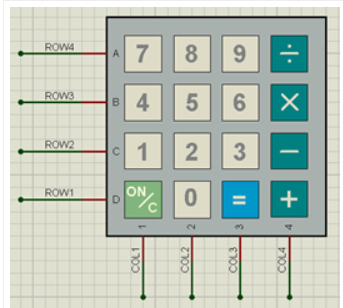
* **Nguyên lý hoạt động**

Để làm việc với Keypad 4×4, người lập trình thường sử dụng giải thuật “quét phím”. Giải thuật này yêu cầu VĐK liên tục đưa các tín hiệu đầu ra ở hàng (hoặc cột) và thu lại đầu vào ở cột (hoặc hàng), nếu phím được bấm, đầu phát tín hiệu sẽ được kết nối với đầu thu, từ đó xác định được phím đã bấm.

Việc lựa chọn đầu ra/vào hình thành 2 phương pháp quét phím: theo chiều dọc và theo chiều ngang. Trong báo cáo này, tín hiệu xuất ra ở các hàng và thu lại ở các cột.

Giả sử một nút ‘2’ được nhấn, khi đó đường C và 2 được nối với nhau. Nếu đường C được nối với GND, khi đó, điện áp ở chân số 2 sẽ mang điện áp 0V. Tương tự như thế với các phím cùng hàng C.

Áp dụng thuật toán vào mô hình KeyPad dưới đây:



Hình 2.11 Mô hình Keypad áp dụng thuật toán “Quét phím”

**Bước 1:** set các chân ROW1, ROW2, ROW3, ROW4 như các chân Output và giữ chúng ở mức cao, các chân COL1, COL2, COL3, COL4 như các chân input có điện trở kéo lên.

**Bước 2:** đưa tín hiệu đầu ra ở các chân ROW1 = 1, ROW2 = 1, ROW3 = 1 và ROW4 =1. Kiểm tra tín hiệu ở các chân COL1, COL2, COL3, COL4 luôn bằng 1 dù có phím nào được nhấn hay không.

**Bước 3:** đưa tín hiệu đầu ra ở các chân ROW1 = 0, ROW2 = 1, ROW3 = 1 và ROW4 =1.

Kiểm tra COL1, 2, 3 và 4, nếu phím thuộc hàng 1 được nhấn sẽ giá trị COL nhận được bằng 0, ví dụ: ON/C được nhấn, COL1 = 0, COL2, 3, 4 = 1. Nếu phím thuộc các hàng khác (2, 3, 4) được nhấn, các chân COL1, COL2, COL3, COL4 luôn bằng 1.

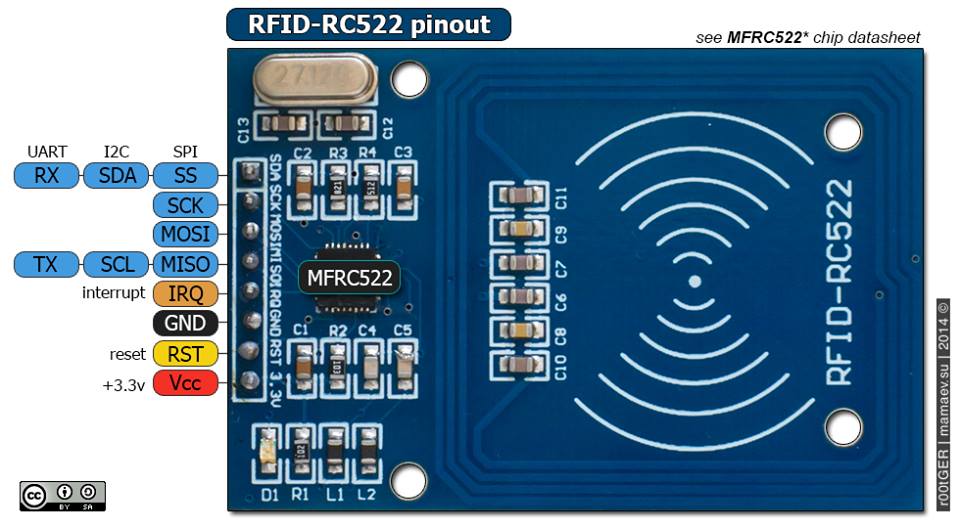
Bằng thao tác trên, sẽ xác định phím đã được nhấn nếu nó ở hàng 1.

**Bước 4:** tiếp tục đưa tín hiệu đầu ra ở các chân ROW1 = 1, ROW2 =0, ROW3 = 1, ROW4 = 1 để xác định phím bấm được nhấn nếu nó ở hàng 2.

**Bước 5:** thực hiện quá trình dịch chân đầu ra mang điện áp mức 0 một cách liên tục và xác định phím được bấm.

## 2.6 Giới thiệu về module RFID RC522

### 2.6.1 RFID RC522

****

Hình 12 Module RFID RC522

Module đọc thẻ RC522 có thể đọc được các loại thẻ có kết nối không dây như NFC, thẻ từ (loại dùng làm thẻ giảm giá, thẻ xe bus, tàu điện ngầm…).

### 2.6.2 Thông số kĩ thuật:

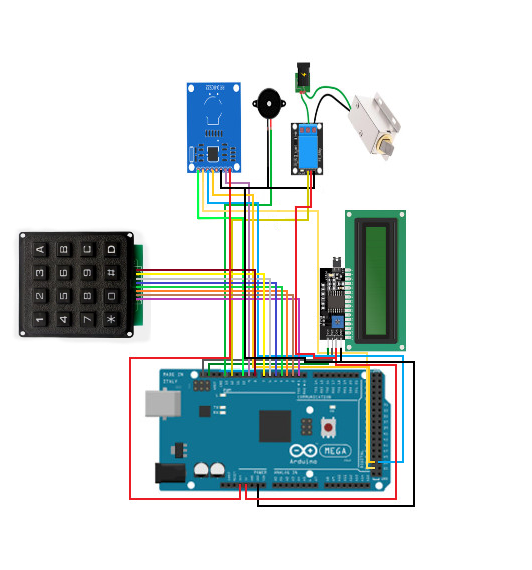
* Điện áp nuôi: 3.3V;
* Dòng điện nuôi :13-26mA
* Tần số hoạt động: 13.56MHz
* Khoảng cách hoạt động: 0 ~ 60 мм
* Cổng giao tiếp: SPI, tốc độ tối đa 10Мbps
* Kích thước: 40мм х 60мм
* Có khả năng đọc và ghi[.](http://arduino.vn/bai-viet/833-lap-trinh-va-su-dung-modul-doc-rfid-rc522)
* Nhiệt độ hoạt động: -20 đến 80℃.
* Độ ẩm hoạt động: 5% - 95%
* Hỗ trợ ISO / IEC 14443A / MIFARE

### 2.6.3 Chân kết nối

* 1: SDA(CS)-Chân lựa chọn chip khi giao tiếp SPI( Kích hoạt ở mức thấp).
* 2: SCK-Chân xung trong chế độ SPI.
* 3: MOSI(SDI)-Master Data Out- Slave In trong chế độ giao tiếp SPI.
* 4: MISO(SDO)-Master Data In- Slave Out trong chế độ giao tiếp SPI.
* 5: IRQ – Chân ngắt.
* 6: GND – Chân mass.
* 7: RST – Chân reset module.
* 8: 3V3.

# CHƯƠNG 3: CƠ SỞ THỰC HIỆN

## 3.1 Sơ đồ kết nối của mô hình



Hình 13 Sơ đồ kết nối của mô hình

## 3.2 Lưu đồ giải thuật



# CHƯƠNG 4: TỔNG KẾT VÀ ĐỊNH HƯỚNG ĐỀ TÀI

## 4.1 Kết quả đạt được

Về phần cứng: tìm hiểu được cách đi dây điện cho các linh kiện, đồng thời hoàn thành được mô hình theo đúng dự kiến.

Về phần mềm: làm được code cho mạch khoá cửa RFID và chương trình hoạt động ổn định.



Hình 14 Mặt trước mô hình



Hình 15 Mặt sau mô hình

## 4.2 Hướng phát triển đề tài

Với những ưu điểm vượt trội của khoá cửa bảo mật và thẻ chip RFID như độ an toàn và bảo mật cao, dễ dàng làm quen và sử dụng, chi phí đầu tư và sử dụng thấp… đồng thời khắc phục được những loại khoá cửa truyền thống ngày nay khi mà ngày càng lộ rõ những nhược điểm về vấn đề an ninh, sự bất tiện khi bị kẹt khoá, làm mất chìa…

Hiện nay công nghệ RFID đã được ứng dụng rất phổ biến trên toàn cầu, nó đã giúp cho việc quản lý, kiểm soát ra vào cửa 1 cách dễ dàng hơn, RFID là sự thay thế một cách thông minh nhất cho các hệ thống khoá cửa thông thường dùng bằng chìa khoá, với sự tiện dụng, nhanh chóng, tính thẩm mỹ cao, đặc biệt là sự an toàn tuyệt đối.

Tại Việt Nam hiên nay đã có nhiều ứng dụng công nghệ RFID được triển khai:

* Trạm thu phí tự động: Trên các cao tốc.
* Hệ thống đỗ xe tự động: tại hầm đỗ xe tòa nhà chung cư.
* Bãi giữ xe thông minh: tai các trung tâm thương mại, bệnh viện, siêu thị, ….
* Sổ khám bệnh điên tử: ứng dụng công nghệ RFID.

# PHỤ LỤC

## Code chương trình

#include <SPI.h>

#include <MFRC522.h>

#include <LiquidCrystal\_I2C.h> // Khai báo thư viện LCD sử dụng I2C

LiquidCrystal\_I2C lcd(0x27, 16, 2); // 0x27 địa chỉ LCD, 16 cột và 2 hàng

#include <Keypad.h> // Khai báo thư viện Keypad

#define SDA\_PIN 10

#define RST\_PIN 9

#define RELAY 12 //relay pin

#define ACCESS\_DELAY 2000

#define DENIED\_DELAY 1000

MFRC522 mfrc522(SDA\_PIN, RST\_PIN); // Create MFRC522 instance.

int speaker =13;

const byte ROWS = 4; // Bốn hàng

const byte COLS = 4; // Ba cột

char keys[ROWS][COLS] = {

{'1', '2', '3', 'A' }, {'4', '5', '6', 'B' }, {'7', '8', '9', 'C' }, {'\*', '0', '#', 'D'}};

byte rowPins[ROWS] = {8, 7, 6, 5};

byte colPins[COLS] = {4, 3, 2, 1};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

char STR[4] = {'0', '9', '0', '8'}; // Cài đặt mật khẩu tùy ý

char str[4] = {' ', ' ', ' ', ' '};

int i, j, count = 0;

void setup() {

Serial.begin(9600); // Initiate a serial communication

lcd.init();

lcd.backlight();

lcd.setCursor(0,0);

lcd.print(" KET SAT TM");

lcd.setCursor(0,1);

lcd.print(" MOI NHAP MK");

SPI.begin(); // Initiate SPI bus

mfrc522.PCD\_Init(); // Initiate MFRC522

pinMode(RELAY, OUTPUT);

digitalWrite(RELAY, HIGH);

Serial.println("Put your card to the reader...");

Serial.println();

pinMode(speaker,OUTPUT);

delay(3000);

}

void loop() {

char key = keypad.getKey(); // Ký tự nhập vào sẽ gán cho biến Key

if (key) // Nhập mật khẩu

{

if (i == 0) {

str[0] = key;

lcd.setCursor(6, 1);

lcd.print("\*"); // Ký tự được che bởi dấu \*

}

if (i == 1) {

str[1] = key;

lcd.setCursor(7, 1);

lcd.print("\*");

}

if (i == 2) {

str[2] = key;

lcd.setCursor(8, 1);

lcd.print("\*");

}

if (i == 3) {

str[3] = key;

lcd.setCursor(9, 1);

lcd.print("\*");

count = 1;

}

i = i + 1;

}

if (count == 1) {

if (str[0] == STR[0] && str[1] == STR[1] && str[2] == STR[2] &&

str[3] == STR[3]) {

lcd.setCursor(0,0);

lcd.print(" CHINH XAC");

lcd.setCursor(1,1);

lcd.print(" MOI QUET THE");

if ( ! mfrc522.PICC\_IsNewCardPresent())

{

return;

}

if ( ! mfrc522.PICC\_ReadCardSerial())

{

return;

}

Serial.print("UID tag :");

String content= "";

byte letter;

for (byte i = 0; i < mfrc522.uid.size; i++)

{

Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");

Serial.print(mfrc522.uid.uidByte[i], HEX);

content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));

content.concat(String(mfrc522.uid.uidByte[i], HEX));

}

Serial.println();

Serial.print("Message : ");

content.toUpperCase();

if (content.substring(1) == "9A AC 31 BF") //THẺ ĐÚNG

{

Serial.println("Authorized access");

Serial.println();

delay(ACCESS\_DELAY);

digitalWrite(RELAY, LOW);

lcd.setCursor(0,1);

lcd.print(" KET SAT DA MO");

digitalWrite(speaker,HIGH);

delay(500);

digitalWrite(speaker,LOW);

}

else {

Serial.println(" Access denied"); //THẺ SAI LOA SẼ KÊU

digitalWrite(RELAY, HIGH);

digitalWrite(speaker,HIGH);

delay(4000);

digitalWrite(speaker,LOW);

}

}

else {

lcd.clear();

lcd.print("SAI PASSWORD!");

digitalWrite(speaker,HIGH); //SAI MẬT KHẨU LOA SẼ KÊU

delay(3000);

lcd.clear();

lcd.print("THU LAI!");

digitalWrite(speaker,LOW);

delay(4000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print(" KET SAT TM");

lcd.setCursor(0,1);

lcd.print("KET SAT DA KHOA");

}

i = 0;

count = 0;

}

switch (key) {

case '#': // ĐÓNG KÉT SẮT

delay(300);

lcd.clear();

lcd.print(" DONG KET SAT!");

digitalWrite(RELAY, HIGH);

digitalWrite(speaker,HIGH);

delay(1000);

digitalWrite(speaker,LOW);

digitalWrite(RELAY, HIGH);

lcd.clear();

lcd.print("MOI NHAP MK");

i = 0;

break;

}

}

**TÀI LIỆU THAM KHẢO**

[1] <https://toc.123doc.net/document/2856748-tong-quan-ve-arduino-uno-r3.ht>

[2] <http://ttz.vn/gioi-thieu-tong-quan-ve-cong-nghe-the-rifd.htm>

[3] <https://suntechled.vn/tong-quan-ve-man-hinh-quang-cao-lcd/>

[4] <http://dammedientu.vn/gioi-thieu-chuan-giao-tiep-i2c/>

[5] <http://arduino.vn/bai-viet/302-module-relay-cach-su-dung-ro-le-va-nhung-ung-dung-hay-cua-no>

[6] <http://linhkien123.vn/chot-khoa-dien-dc-12v-loai-to-822665.html>