# *Chương 1:*

**GIỚI THIỆU YÊU CẦU – GIỚI HẠN CỦA ĐỀ TÀI**

* 1. **GIỚI THIỆU**

Khóa cửa là một dụng cụ thường được làm bằng kim loại để ngăn việc mở cửa, được làm ra với mục đích không cho người khác mở được cửa nếu không được cho phép. Muốn mở cửa đang khóa thì phải có chìa khóa làm bằng kim loại đối với khóa cơ và khóa ổ, hoặc các yêu cầu khác như thẻ từ, vân tay, nhận dạng hình ảnh, nhập mã số,…đối với khóa điện tử.

Khóa điện tử là khái niệm chỉ khóa cửa điện tử, được sử dụng để thay thế các khóa cơ truyền thống bằng việc sử dụng các công nghệ thông minh giúp cho các ổ khóa có thể mở được bằng điều khiển từ xa hoặc mở bằng mã số, vân tay,... Trước đây, tính năng này thường được áp dụng cho nhưng nơi cần an ninh tuyệt đối, nhưng giờ đây nó được ứng dụng rộng rãi để tạo ra các ổ khóa điện tử thông minh hữu ích trong cuộc sống.

Ưu điểm của khóa điện tử:

* Tiện lợi, không cần mang theo, không sợ mất hay để quên chìa khóa.
* An toàn, bảo mật cao.
* Dễ sử dụng.
* Đa dạng về kiểu dáng.

Bên cạnh sự phát triển của các bộ vi xử lí, vi điều khiển, IC số ngày càng thông dụng và hoàn thiện hơn. Khóa điện tử cũng ngày càng được tối ưu hơn: nhỏ gọn hơn, xử lý nhanh hơn, dễ sử dụng hơn,... Nhưng có thể nói sự xuất hiện của Arduino vào năm 2005 tại Italia đã mở ra một hướng đi mới. Sự xuất hiện của Arduino đã hỗ trợ cho con người rất nhiều trong lập trình và thiết kế. Phần cứng của thiết bị đã được tích hợp nhiều chức năng cơ bản và mã nguồn mở. Thư viện rất phong phú và được chia sẻ miễn phí. Dựa vào những yếu tố đó chúng em chọn đề tài **“THIẾT KẾ MẠCH KHÓA SỐ DÙNG ARDUINO”** để nghiên cứu và chọn làm đề tài Đồ án môn học 1 này.

Nội dung đề tài này sẽ xây dựng hệ thống theo dõi, điều khiển đóng mở cửa ra vào, có thể tùy chỉnh mã số theo ý muốn, cảnh báo khi nhập sai mã và cố gắng mở khóa khi không được cho phép.

* 1. **GIỚI HẠN**
* Sử dụng Board Arduino Uno để làm khối điều khiển trung tâm.
* Sử dụng LCD 16x2 để hiển thị.
* Sử dụng Keypad 4x4 để nhập mật mã.
* Sử dụng động cơ Servo để điều khiển chốt cửa.
* Sử dụng còi chip 5v để phát tiếng báo động.
* Giới hạn mã số ở 8 chữ số, không dùng được chữ cái hoặc ký tự đặc biệt.

# *Chương 2:*

**THIẾT KẾ SƠ ĐỒ KHỐI**

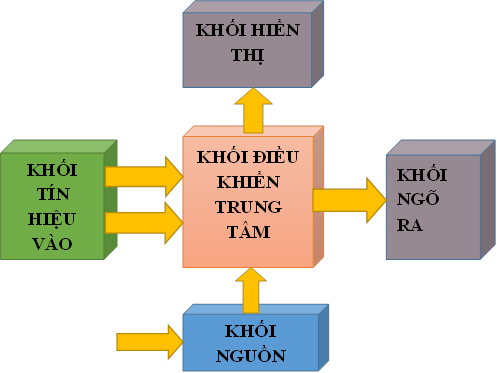
* 1. **GIỚI THIỆU**

Đề tài này giám sát việc đóng mở cửa bằng mã số. Người dùng có thể quản lý được việc đóng mở cửa, nhận được cảnh báo khi có người cố gắng mở cửa mà chưa được cho phép. Có thể trực tiếp thay đổi mã số khi cần. Màn hình hiển thị đầy đủ thông tin cần thiết cho người dùng và có thể sử dụng ngay cả khi mất điện.

## **THIẾT KẾ SƠ ĐỒ KHỐI**

* + 1. **VẼ SƠ ĐỒ KHỐI**

Theo yêu cầu của đề tài thì nhóm chúng em tiến hành thiết kế sơ đồ khối của mạch khóa số điện tử dùng Arduino như ***Hình 2.1:***



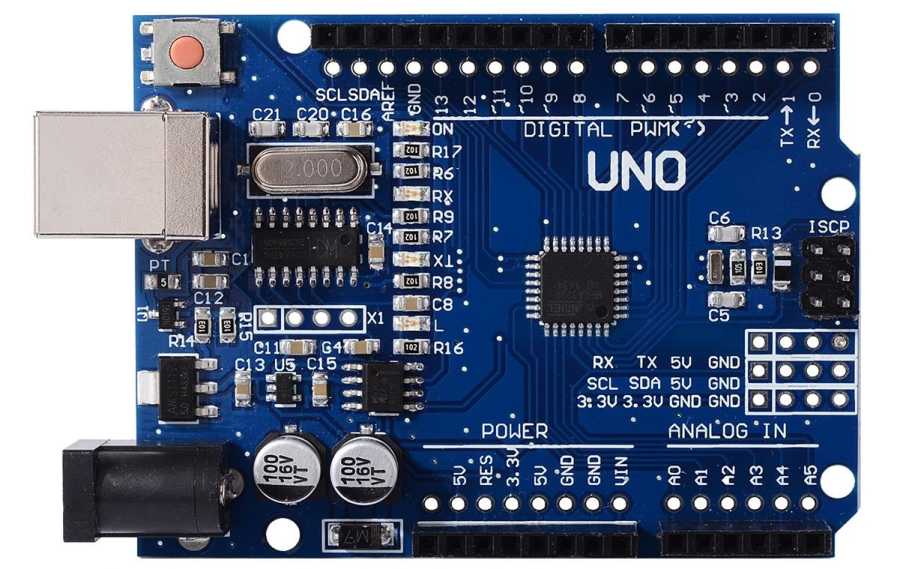
Hình 1 Sơ đồ khối hệ thống khóa số điện tử

* + 1. **CHỨC NĂNG VÀ GIẢI THÍCH TỪNG KHỐI**
* Khối nguồn: Cung cấp nguồn cho toàn mạch, trong đề tài này sử dụng loại nguồn 5VDC cấp cho mạch điều khiển, động cơ và chuông báo động.
* Khối điều khiển trung tâm: Ở đây sử dụng bộ điều khiển trung tâm là Board Arduino Uno, tạo ra các lệnh điều khiển, điều khiển các hoạt động của hệ thống, nhận tín hiệu từ Keypad và gửi tín hiệu điều khiển sang các khối khác.
* Khối tín hiệu vào: Sử dụng Keypad 4x4 đề nhập mã số, sau đó gửi về Board Arduino để xử lý.
* Khối hiển thị: Sử dụng LCD 16x2 để hiển thị mã số và kết quả.
* Khối ngõ ra: Sử dụng động cơ Servo để điều khiển đóng mở cửa và chuông để cảnh báo.
  1. **THIẾT KẾ SƠ ĐỒ NGUYÊN LÍ**
     1. **KHỐI ĐIỀU KHIỂN TRUNG TÂM**

Trong những năm gần đây, sự phát triển mạnh mẽ của Arduino đã cho thấy tầm ảnh hưởng rất nhiều tới thế giới hiện tại, với cộng đồng mạng hỗ trợ, thư viện hỗ trợ hầu hết các module, các sản phẩm thiết bị ngoại vi khác, thấy được sự quan trọng và phù hợp với yêu cầu trong đề tài này, nên nhóm chúng em chọn Arduino. Các loại bộ điều khiển trung tâm khác cũng có thể sử dụng nhưng do thiết kế và lập trình phức tạp hơn, nên lựa chọn Arduino để có thể đơn giản hóa các khâu như thiết kế, lập trình, …

Bộ điều khiển trung tâm chúng em chọn sử dụng Arduino UNO R3 vì có thể lập trình cũng như kết nối dễ dàng hơn nhờ cộng đồng người dùng đông đảo. Arduino UNO R3 sử dụng vi điều khiển ATmega328 họ 8 bit, với 14 chân digital dùng để đọc hoặc xuất tín hiệu. Arduino UNO có thể được cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyên dùng là 7-12V DC và giới hạn là 6-20V. Thường thì cấp nguồn bằng pin vuông 9V là hợp lí nhất nếu không có sẵn nguồn từ cổng USB. Nếu cấp nguồn vượt quá ngưỡng giới hạn trên, sẽ làm hỏng Arduino UNO.

Bên cạnh đó, bộ điều khiển trung tâm cũng như các thiết bị ngõ ra được lắp đặt trong một hộp điều khiển để đảm bảo việc hệ thống vận hành được an toàn và tốt hơn.



Hình 2 Board Arduino UNO R3

* Các chân năng lượng:
* 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.
* IOREF: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn 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.
* Các cổng vào ra: 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).
* Chân Serial: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết
* Chân PWM: 3, 5, 6, 9, 10, và 11: cho phép bạn 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, bạn 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.
* LED 13: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.
* 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, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn 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.

***Bảng 2.1****: Thông số kỹ thuật Board Arduino UNO.*

|  |  |
| --- | --- |
| Điện áp hoạt động | 5V |
| Điện áp vào (đề nghị) | 7V – 12V |
| Điện áp vào (giới hạn) | 6V – 20V |
| Chân số (Digital I/O) | 14 ( 6 chân PWM ) |
| Chân tương tự | 6 ( độ phân giải 10bit ) |
| Dòng tối đa trên mỗi chân I/O | 30 mA |
| Dòng ra tối đa ( 5V ) | 500 mA |
| Dòng ra tối đa ( 3,3V ) | 50 mA |
| Flash Memory | 32 KB trong đó 0,5 KB được sử dụng bởi Bootloader |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Xung nhịp | 16 MHz |

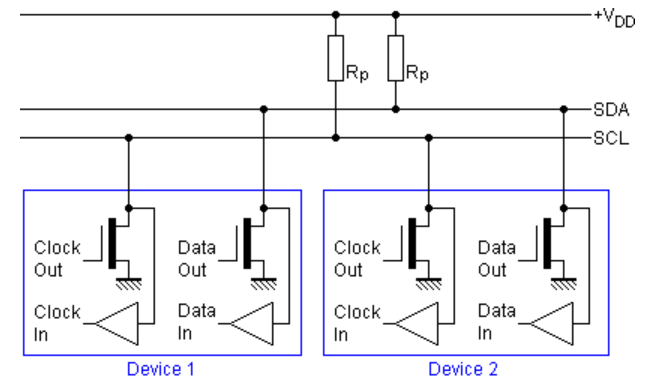
* + 1. **KHỐI HIỂN THỊ**

1. **Chuẩn giao tiếp I2C:**

* *Khái niệm:*I2C là viết tắt của cụm từ Tiếng Anh “Inter-Integrated Circuit”, là một loại bus nối tiếp được phát triển bởi hãng sản xuất linh kiện điện tử Philips.
* *Đặc điểm:*I2C sử dụng hai đường truyền tín hiệu *(như hình 2.7):*

+ Một đường xung nhịp đồng hồ (SCL) chỉ do Master phát đi.

+ Một đường dữ liệu (SDA) theo 2 hướng.



Hình 3 Sơ đồ kết nối thiết bị theo chuẩn giao tiếp I2C

Bản chất của I2C là dữ liệu trên đường SDA chỉ được ghi nhận ở sườn lên của chân CLK. SCL và SDA luôn được kéo lên nguồn bằng một điện trở kéo lên có giá trị xấp xỉ 4,7 kΩ (tùy vào từng thiết bị và chuẩn giao tiếp, có thể dao động trong khoảng 1 kΩ đến 4.7kΩ.

* Chế độ hoạt động:

Các dữ liệu trên bus I2C có thể được truyền trong ba chế độ khác nhau, cụ thể như bên dưới:

+ Chế độ tiêu chuẩn: đây là chế độ tiêu chuẩn ban đầu được phát hành vào đầu những năm 80, có tốc độ dữ liệu tới đa 100kbps và sử dụng 7-bit địa chỉ, 112 địa chỉ tới.

+ Tăng cường hoặc chế độ nhanh: tốc độ dữ liệu tối đa được tăng lên đến 400kbps. Để ngăn chặn gây tiếng ồn, ngõ vào của thiết bị Fast-mode là Schmitt-triggered. Chân SCL và SDA của một thiết bị I2C ở trạng thái trở kháng cao khi không cấp nguồn.

* Chế độ cao tốc: Chế độ này đã được tạo ra chủ yếu để tăng tốc độ dữ liệu lên đến 36 lần nhanh hơn so với chế độ tiêu chuẩn. Nó cung cấp 1,7Mbps (với Cb =400µF) và 3,4Mbps (với C>b = 100pF).

Một bus I2C có thể hoạt động ở nhiều chế độ khác nhau:

* Một chủ một tớ (one master - one slave).
* Một chủ nhiều tớ (one master - multi slave).
* Nhiều chủ nhiều tớ (Multi master - Multi slave)
* Dù ở chế độ nào, một giao tiếp I2C đều dựa vào quan hệ chủ/tớ. Giả thuyết một thiết bị A muốn gửi dữ liệu đến thiết bị B, quá trình được thực hiện như sau:
* Thiết bị A (Chủ) xác định đúng địa chỉ của thiết bị B (tớ), cùng với việc xác định địa chỉ, thiết bị A sẽ quyết định việc đọc hay ghi vào thiết bị tớ.
* Thiết bị A gửi dữ liệu tới thiết bị B.
* Thiết bị A kết thúc quá trình truyền dữ liệu.

Khi A muốn nhận dữ liệu từ B, quá trình diễn ra như trên. Tuy nhiên, A sẽ nhận dữ liệu từ B. Trong trường hợp này, A là chủ còn B vẫn là tớ.

1. **LCD 16x2:**

Hiện nay, màn hình hiển thị có rất nhiều loại được sử dụng vào từng mục đích khác nhau như: LCD 10.1 inch cảm ứng điện dung Waveshare, màn hình cảm ứng Arduino TFT Shield 2,4inch, màn hình LCD Graphic, LCD text, LCD HMI. Tùy vào mục đích và cách sử dụng cũng như ứng dụng của dự án mà chúng ta sử dụng màn hình loại nào cho tiện lợi. Màn hình text LCD 16x2 sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 kí tự, phổ biến và giá thành rẽ. Thích hợp cho việc hiển thị các nội dung đơn giản, dễ sử dụng.



Hình 4 LCD 16x02

LCD có nhiều loại và số chân của chúng cũng khác nhau nhưng có 2 loại phổ biến là loại 14 chân và loại 16 chân, sự khác nhau là các chân nguồn cung cấp, còn các chân điều khiển thì không thay đổi.

***Bảng 2.2:*** *Các chân của LCD.*

|  |  |  |  |
| --- | --- | --- | --- |
| Thứ tự | Tên tín hiệu | I/O | Mô tả |
| 1 | VSS | Nguồn | GND |
| 2 | VDD | Nguồn | +5V |
| 3 | VO | Điện áp | Điều khiển ánh sáng nền |
| 4 | RS | INPUT | Register Select |
| 5 | R/W | INPUT | Read/Write |
| 6 | E | INPUT | Enable (strobe) |
| 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 |

Trong 16 chân của LCD được chia ra làm 3 dạng tín hiệu như sau:

* Các chân cấp nguồn: Chân số 1 là chân nối mass (0V), chân thứ 2 là Vdd nối với nguồn +5V. Chân thứ 3 dùng để chỉnh contrast thường nối với biến trở.
* Các chân điều khiển: Chân số 4 là chân RS dùng để điều khiển lựa chọn thanh ghi. Chân R/W dùng để điều khiển quá trình đọc và ghi. Chân E là chân cho phép dạng xung chốt.
* Các chân dữ liệu D7÷D0: Chân số 7 đến chân số 14 là 8 chân dùng để trao đổi dữ liệu giữa thiết bị điều khiển và LCD.

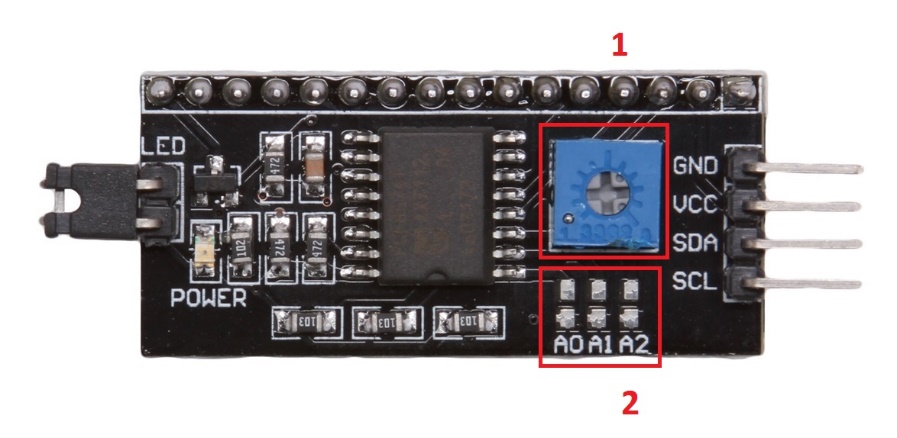
1. **Kết nối với I2C:**

Thông thường, để sử dụng màn hình LCD, phải mất rất nhiều chân trên Arduino để điều khiển. Do vậy, để đơn giản hóa công việc, người ta đã tạo ra một loại mạch điều khiển màn hình LCD sử dụng giao tiếp I2C. Nói một cách đơn giản, chỉ tốn 2 dây để điều khiển màn hình, thay vì 8 dây như cách thông thường.

Module: **Lcd I2C- PCF8574**

Điện áp sử dụng: 5VDC

Giao tiếp: I2C

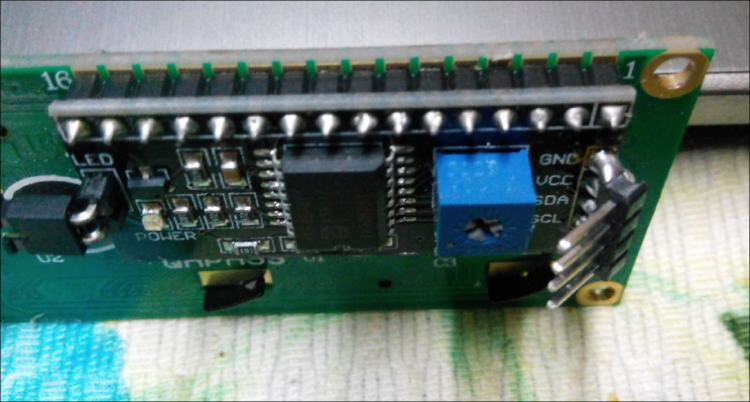


Hình 5 Module giao tiếp I2C

1 : Biến trở điều chỉnh độ tương phản của màn hình LCD.

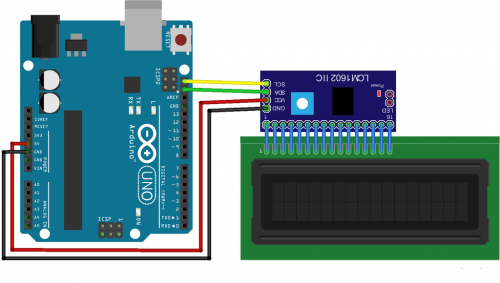
2 : Đây là khu vực thay đổi địa chỉ I2C, nếu muốn kết nối nhiều thiết bị trên 1 đường I2C địa chỉ 0x27, thì bạn phải thay đổi nó, mặc định nhà sản xuất là A0, A1, A2 =1 (mức cao) như thế này Module có địa chỉ là 0X27 hoặc 0x3F.

Chỉ cần hàn mạch như hình 2.6 bên dưới là xong. Bởi vì giao tiếp I2C được thiết kế riêng nhằm giúp LCD giao tiếp với Board xử lý một cách dễ dàng, nên rất dễ kết nối. 2 chân SDA và SCL là 2 chân tín hiệu dùng cho giao tiếp I2C.



Hình 6 Kết nối Module I2C với LCD

1. **Kết nối với Arduino Uno:**



Hình 7 Kết nối LCD I2C với Arduino Uno

***Bảng 2.3:*** *Kết nối module với màn hình Arduino*

|  |  |
| --- | --- |
| **Module màn hình LCD (16x2)** | **Arduino** |
| GND | GND |
| VCC | 5V |
| SDA | SDA |
| SCL | SCL |

Để sử điều khiển I2C bằng Arduino ta thêm thư viện LiquidCrystal\_I2C.h vào phần mềm lập trình.

Các lệnh điều khiển cơ bản:

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

LiquidCrystal\_I2C lcd**(**0x3f**,** 16**,** 2**);** // 0x3f địa chỉ LCD, 16 cột và 2 hàng

void setup**()** **{**

lcd**.**init**();** //Khởi động màn hình. Bắt đầu cho phép Arduino sử dụng màn hình

lcd**.**begin**(**16**,** 2**);**

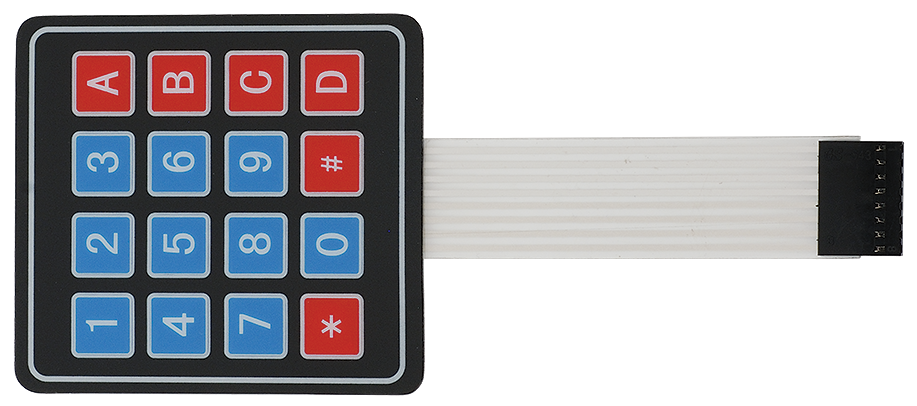
lcd**.**backlight**();** //Bật đèn nền

lcd**.**clear**();** //Xóa màn hình

lcd**.**print**(**" "**);}** //In ra màn hình

* + 1. **KHỐI TÍN HIỆU VÀO**

Hệ thống với mật khẩu nhập vào là các con số và một vài nút nhấn điều khiển đơn giản, không yêu cầu các ký tự đặc biệt nên chúng em sử dụng keypad4x4. Bàn phím này có 16 nút nhấn, xếp thành hình vuông 4 ngang 4 dọc nhưng chỉ cần 8 chân kết nối để điều khiển.



Hình 8 Keypad 4x4

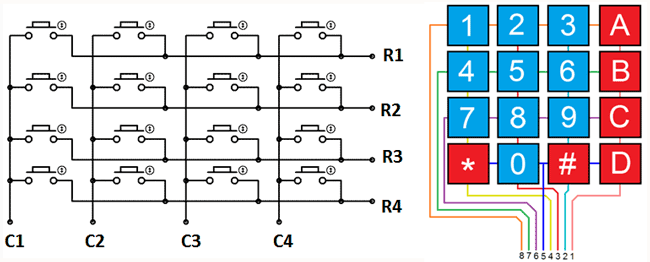
Với cách kết nối dạng ma trận thì 16 phím chỉ dùng 8 tín hiệu: 4 cho hàng và 4 cho cột – gọi là ma trận 4×4 sẽ được 16 phím.

Trong ma trận 4×4 thì có 4 hàng và 4 cột, hàng được chọn là tín hiệu vào – cột được chọn là tín hiệu ra, hàng thì treo lên nguồn Vcc qua điện trở nên mức logic của hàng luôn là mức 1.

Các phím nhấn thường hở nên 4 hàng luôn ở mức 1 hay R4R3R2R1 =1111.

Cột là tín hiệu ra nên chúng ta điều khiển xuất dữ liệu ra cột tùy ý. Để phân biệt các phím thì mỗi phím có 1 tên được đánh theo số thập lục phân từ ‘0’ đến ‘F’.

Để xem có phím nào nhấn hay không ta tiến hành quét từng cột bằng cách cho 1 cột ở mức 0, 3 cột còn lại ở mức 1 và kiểm tra tất cả các hàng, nếu tất cả các hàng vẫn ở mức logic 1 tức là không có nhấn phím, nếu có 1 hàng xuống mức 0 thì đã có nhấn phím.



Hình 9 Ma trận phím 4x4

Mã quét cột là 4 bit thấp bao gồm các trạng thái: “1110B”, “1101B”, “1011B”, “0111B” kết hợp với 4 bit cao của hàng luôn là “1111B” nên dữ liệu kết hợp theo byte viết dạng số hex là “FEH”, “FDH”, “FBH”, “F7H”.

Gán cột bắt đầu từ 0, hàng gán bằng FFH, mã phím ban đầu gán bằng FFH.

Xuất lần lượt từng mã quét cột ra port quét phím, tiến hành kiểm tra từng hàng xem hàng nào bằng 0 thì phím tương ứng được nhấn. Khi đó ta gán giá trị tương ứng cho từng hàng.

Cuối cùng kiểm tra hàng xem có bằng FFH hay không, nếu bằng thì không phím nào được nhấn trong lúc thực hiện chương trình quét, thoát với mã phím bằng FFH, nếu khác FFH thì đã có nhấn phím, tiến hành tính toán mã phím và thoát với mã phím là của phím đã nhấn.

Nếu quét cột thứ 0 cột bằng 0, theo công thức mã phím sẽ có giá trị từ 0H đến 3H.

Nếu quét cột thứ 1 cột bằng 1, theo công thức mã phím sẽ có giá trị từ 4H đến 7H.

Nếu quét cột thứ 2 cột bằng 2, theo công thức mã phím sẽ có giá trị từ 8H đến BH.

Nếu quét cột thứ 3 cột bằng 3, theo công thức mã phím sẽ có giá trị từ CH đến FH



Hình 10 Các chân Keypad

***Bảng 2.3:*** *Kết nối Keypad với Arduino*

|  |  |
| --- | --- |
| Bàn phím | Arduino Uno |
| R1 | 2 |
| R2 | 3 |
| R3 | 4 |
| R4 | 5 |
| C1 | 6 |
| C2 | 7 |
| C3 | 8 |
| C4 | 9 |

* + 1. **KHỐI NGÕ RA**

1. **Động cơ RC Servo:**

Sử dụng động cơ RC Servo 9G để điều khiển đóng mở cửa. Động cơ Servo có kích thước nhỏ, tốc độ phản ứng nhanh, tích hợp sẵn Drive điều khiển động cơ bên trong nên có thể dễ dàng điều chỉnh góc quay chích xác

. 

Hình 11 Động cơ RC Servo 9G

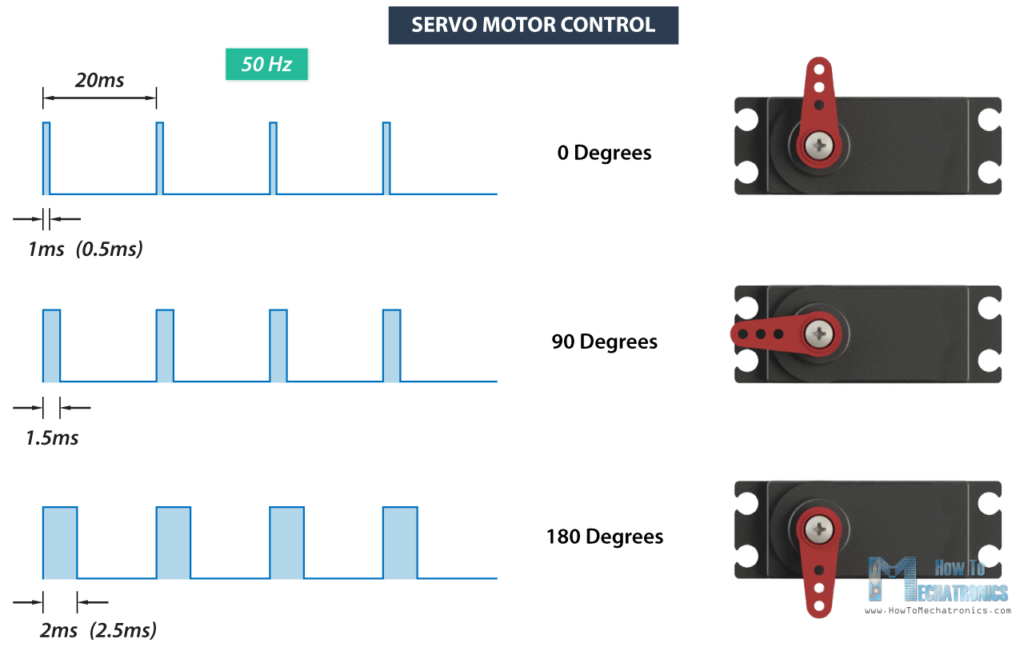
Servo là một dạng động cơ điện đặc biệt. Không giống như động cơ thông thường khi cung cấp nguồn vào là quay liên tục, servo chỉ quay khi được điều khiển (bằng xung PPM) với góc quay nằm trong khoảng bất kì từ 0o ­­-180o. Mỗi loại servo có kích thước, khối lượng và cấu tạo khác nhau. Có loại thì nặng chỉ 9g (chủ yếu dùng trên máy bay mô mình), có loại thì sở hữu một momen lực lớn (vài chục Newton/m), hoặc có loại thì khỏe và nhông sắc chắc chắn,...

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 bầt kỳ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác.

Bên trong một servo cổ diển có bốn thành phần chính, động cơ DC, hộp số, biến trở và mạch điều khiển. Động cơ DC có tốc độ cao và mô-men xoắn thấp nhưng hộp số giảm tốc độ xuống khoảng 60 vòng / phút, đồng thời tăng mô-men xoắn.

Chiết áp được gắn trên bánh răng cuối cùng hoặc trục đầu ra, do đó động cơ cũng quay chiết áp, tạo ra một điện áp liên quan đến góc tuyệt đối của trục đầu ra. Trong mạch điều khiển, điện áp chiết áp này được so sánh với điện áp đến từ đường tín hiệu. Nếu cần, bộ điều khiển kích hoạt mạch cầu H tích hợp cho phép động cơ quay theo hai hướng cho đến khi hai tín hiệu đạt mức chênh lệch bằng không.

Một động cơ servo được điều khiển bằng cách gửi một loạt các xung qua đường tín hiệu. Tần số của tín hiệu điều khiển phải là 50Hz hoặc một chu kỳ xung là 20ms. Độ rộng của xung xác định vị trí góc của servo và các loại servo này thường có thể xoay 180 độ (chúng có giới hạn vật lý khi di chuyển).



Hình 12 Tín hiệu điều khiển động cơ RC Servo

Nói chung các xung có tại 1ms tương ứng với vị trí 0 độ, 1,5ms đến 90 độ và 2ms đến 180 độ. Mặc dù thời gian tối thiểu và tối đa của các xung đôi khi có thể thay đổi theo các loại khác nhau và chúng có thể là 0,5ms cho 0 độ và 2,5ms cho vị trí 180 độ.

***Bảng 2.4:*** *Thông số kỹ thuật Động cơ RC Servo 9G*

|  |  |
| --- | --- |
| Điện áp hoạt động | 4.8 - 5 VDC |
| Kích thước | 21 x 12 x 20 mm |
| Tốc độ | 0.12 sec / 60 degrees |
| Lực kéo | 1.6 kg.cm |

1. **Còi Buzzer:**

Sử dụng còi Buzzer để làm còi báo động. Vì còi Buzzer có tuổi thọ cao, hiệu suất ổn định, nhỏ gọn phù hợp với thiết kế.

Buzzer còn gọi là loa mini hay còi báo. Nó dùng để phát ra các âm thanh. Gồm 2 chân: Chân GND nối GND arduino và chân tín hiệu nối pin Arduino.



Hình 13 Còi Buzzer

***Bảng 2.5:*** *Thông số kỹ thuật còi Buzzer*

|  |  |
| --- | --- |
| Điện áp hoạt động | 3.5 – 5.5 VDC |
| Kích thước | Đường kính 12mm, cao 9.7mm |
| Biên độ âm | >80 dB |
| Tần số cộng hưởng | 2300Hz ± 500Hz |
| Dòng điện tiêu thụ | <25mA |

* + 1. **KHỐI NGUỒN**

Dựa vào thông số kỹ thuật của nhà sản xuất, chọn mức điện áp sử dụng là 5VDC.

***Bảng 2.6:*** *Liệt kê dòng điện sử dụng cho mạch điều khiển*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên linh kiện** | **Dòng điện cực đại (mA)** | **Số lượng** | **Giá trị (A)** |
| 1 | Arduino Uno | 500 | 1 | 0.5 |
| 2 | LCD 16x02 và module I2C | 40 | 1 | 0.04 |
| **Tổng cộng:** | | | | **0.54** |

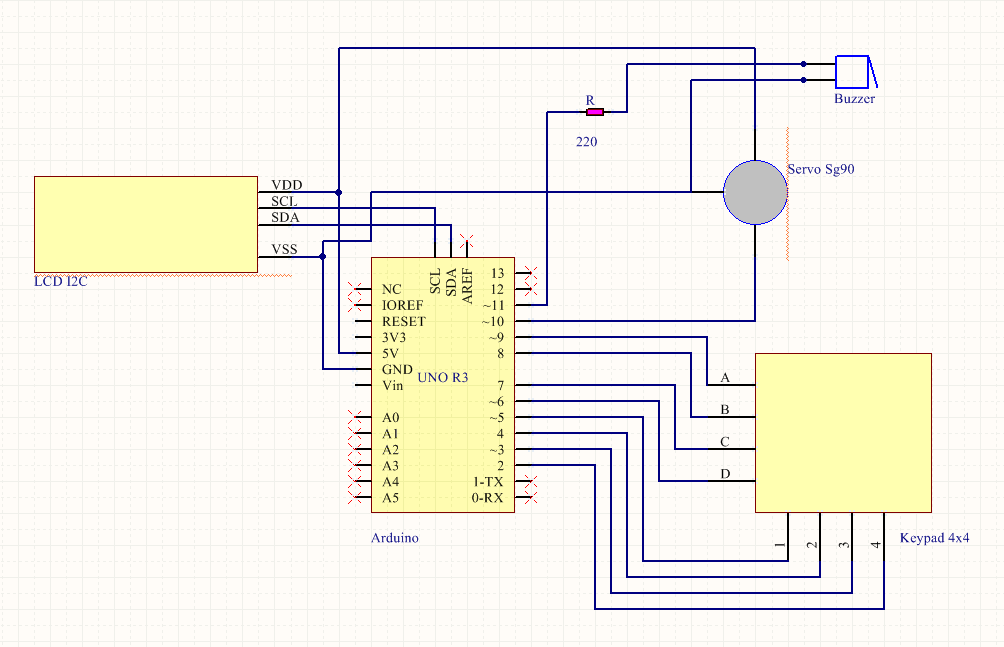
***Bảng 2.7:*** *Liệt kê dòng điện sử dụng cho mạch điều khiển các thiết bị ngõ ra*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên linh kiện** | **Dòng điện cực đại (mA)** | **Số lượng** | **Giá trị (A)** |
| 1 | Động cơ Servo | 1000 | 1 | 1 |
| 2 | Còi Buzzer | 25 | 1 | 0.025 |
| **Tổng cộng:** | | | | **1.025** |

Từ các bảng trên, ta có:

**I = Iđk + Ingra = 0.54 +1.025 = 1.79 A**

* **Chọn dòng điện cấp cho board điều khiển: I = 2 (A)**
  + 1. **SƠ ĐỒ NGUYÊN LÝ TOÀN MẠCH**

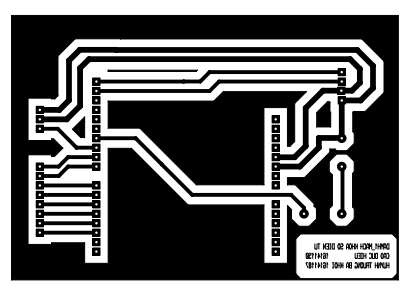


Hình 14 Sơ đồ nguyên lý Mạch khóa số tự động sử dụng Arduino UNO

# *Chương 3:*

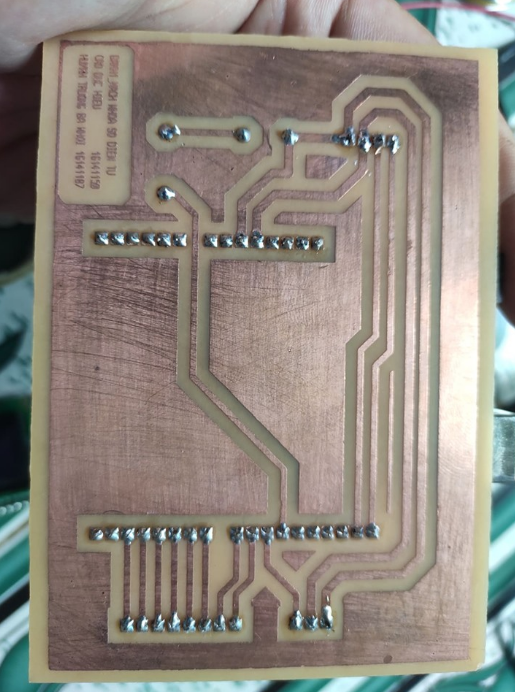
**THI CÔNG MẠCH – VIẾT CHƯƠNG TRÌNH**

## **3.1 THI CÔNG PCB**

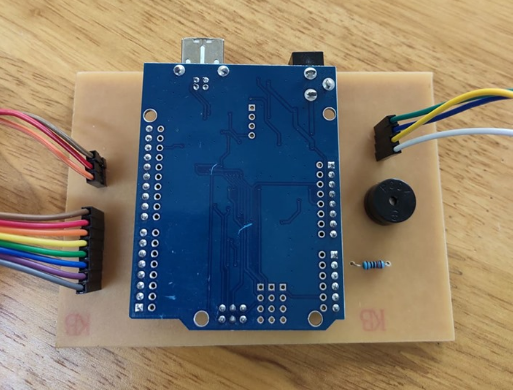
****

Hình 15 Sơ đồ mạch in

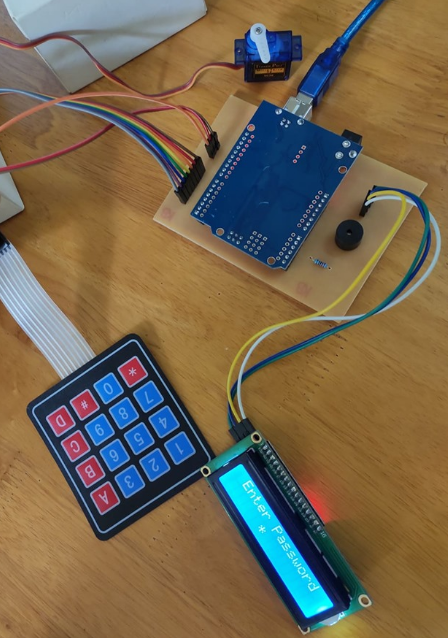
## **3.2 HÀN LINH KIỆN**



Hình 16 Mạch sau khi hàn xong



Hình 17 Mạch đã gắn linh kiện



Hình 18 Mạch hoàn chỉnh

## **3.3 VIẾT CHƯƠNG TRÌNH ĐIỀU KHIỂN**

### 3.3.1 YÊU CẦU GIẢI THUẬT

Yêu cầu giải thuật: Ban đầu khi cấp nguồn, hiển thị LCD. Nhấn phím A để bắt đầu nhập mật khẩu. Nhấn các số từ 0 – 9 của Keypad để tiến hành nhập mật khẩu, các số nhập vào hiển thị trên LCD rồi chuyển thành “ \* ” ngay sau đó. Mạch sẽ so sánh mật khẩu nhập vào và hoạt động theo hai hướng:

* Mật khẩu nhập vào đúng: Servo hoạt động để điều khiển mở chốt cửa và thông báo lên LCD.
* Mật khẩu nhập vào sai: thông báo lên LCD và tiến hành nhập lại mật khẩu. Nếu nhập sai mật khẩu 4 lần liên tiếp, mạch sẽ khóa chức năng nhập mật khẩu và bật còi để phát cảnh báo.

### 3.3.2 LƯU ĐỒ GIẢI THUẬT

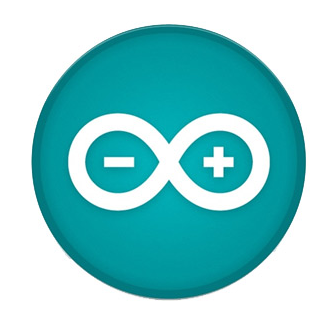


Hình 19 Lưu đồ chương trình chính điều khiển Mạch khóa số điện tử sử dụng Arduino

Lưu đồ chương trình chính:

* Khi cấp nguồn mạch bắt đầu hoạt động. LCD hiển thị “HELLO”, chưa cho phép nhập mật khấu. Khi nhấn phím A của Keypad, LCD hiển thị “Enter Password” ở hàng 1, biến i=1 cho phép bắt đầu nhập mật khẩu. Chạy chương trình “Nhập mật khẩu”.
* Trong quá trình nhâp mật khẩu nếu ta nhấn phím “#” của Keypad chương trình sẽ dừng lại cho phép nhập mật khẩu và quay trở về ban đầu, biến i=0.
* Khi nhập xong mật khẩu, biến count=1, thực hiện kiểm tra mật khẩu. So sánh mảng mật khẩu vừa mới nhập srt[ ] với mảng mật khẩu đã đặt trước SRT[ ]. Mật khẩu nhập vào đúng sẽ chạy chương trình mở cửa. Mật khẩu sai biến i=1, biến count=0, cho phép nhập lại mật khẩu đồng thời tăng biến cảnh báo (cb++). Nếu nhập sai mật khẩu 3 lần liên tiếp, biến cảnh báo cb=3 chạy chương trình cảnh báo.

### 3.3.3 PHẦN MỀM LẬP TRÌNH ARDUINO IDE

****

Hình 21 Biểu tượng của phần mềm

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 và các cơ cấu chấp hành.

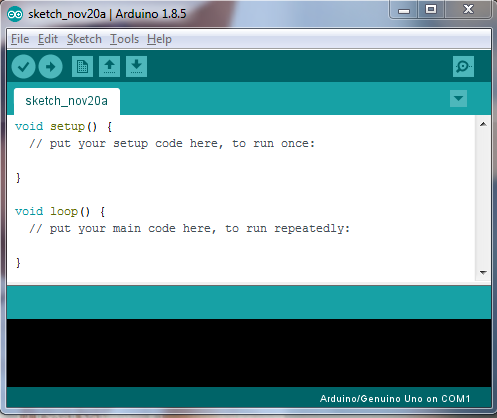
Arduino thật ra là một bo mạch vi xử lý được dùng để lập trình tương tác với các thiết bị phần cứng như cảm biến, động cơ, đèn hoặc các thiết bị khác. Đặc điểm nổi bật của Arduino là môi trường phát triển ứng dụng cực kỳ dễ sử dụng, với một ngôn ngữ lập trình có thể học một cách nhanh chóng ngay cả với 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 thực sự đã gây sóng gió trên thị trường người dùng DIY (là những người tự chế ra sản phẩm của mình) trên toàn thế giới trong vài năm gần đây, gần giống với những gì Apple đã làm được trên thị trường thiết bị di động. Số lượng người dùng cực lớn và đa dạng với trình độ trải rộng từ bậc phổ thông lên đến đại học đã làm cho ngay cả những người tạo ra chúng phải ngạc nhiên về mức độ phổ biến.

Các ứng dụng nổi bật của board mạch Arduino: robot đơn giản, điều khiển nhiệt độ, phát hiện chuyển động, game tương tác....

* **Sơ lược môi trường lập trình Arduino Software:**

Thiết kế bo mạch nhỏ gọn, trang bị nhiều tính năng thông dụng mang lại nhiều lợi thế cho Arduino, tuy nhiên sức mạnh thực sự của Arduino nằm ở phần mềm. Môi trường lập trình đơn giản dễ sử dụng, ngôn ngữ lập trình Wiring dễ hiểu và dựa trên nền tảng C/C++ rất quen thuộc với người làm kỹ thuật. Và quan trọng là số lượng thư viện code được viết sẵn và chia sẻ bởi cộng đồng nguồn mở là cực kỳ lớn.



Hình 22 Giao diện môi trường lập trình cho Arduino

Giao diện lập trình đơn giản, dễ sử dụng. Có một button Verify để biên dịnh chương trình và button Upload để nạp chương trình vào board Arduino. Ngoài ra IDE còn cung cấp một số ví dụ cơ bản, người dùng có thể mở để tham khảo chỉ qua vài thao tác đơn giản.

### 3.3.4 CHƯƠNG TRÌNH ĐIỀU KHIỂN

#include <Servo.h> // Khai báo thư viện sử dụng cho động cơ

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

LiquidCrystal\_I2C lcd**(**0x3f**,** 16**,** 2**);** // 0x3f địa chỉ LCD, 16 cột và 2 hàng

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

Khai báo các thư viện sử dụng, sau dòng khai báo thư viện I2C phải khai báo loại LCD sử dụng với địa chỉ của LCD và số cột số hàng, ở đây là LCD 16 cột 2 hàng.

const byte ROWS **=** 4**;** // Bốn hàng

const byte COLS **=** 4**;** // bốn cột

char keys**[**ROWS**][**COLS**]** **=** **{**

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

byte rowPins**[**ROWS**]** **=** **{**2**,** 3**,** 4**,** 5**};** //chân kết nối

byte colPins**[**COLS**]** **=** **{**6**,** 7**,** 8**,** 9**};**

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

Servo myServo**;**

Khai báo loại Keypad sử dụng số hàng với số cột là 4, chuỗi ma trận mã ký tự cho Keypad lúc đó ứng với vị mã phím sẽ lấy ra ký tự trong chuỗi. Khai báo các chân kết nối cho Keypad theo hàng và cột. Dòng cuối là khai báo tên cho động cơ Servo sử dụng.

char STR**[**4**]** **=** **{**'2'**,** '0'**,** '1'**,** '8'**};** // Cài đặt mật khẩu tùy ý

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

int i**,** count **=** 0**,** cb **=** 0**;**

Tạo mảng để lưu mật khẩu chính và mật khẩu nhập vào từ Keypad, có thể tùy ý đặt mật khẩu chính. Khai báo các biến sử dụng trong chương trình. Kiểu int là kiểu số nguyên chính được dùng trong chương trình Arduino, kiểu int chiếm 2byte bộ nhớ và với Arduino Uno nó có giá trị 16bit.

void setup**()**

**{**

myServo**.**attach**(**10**);** // Khai báo chân điều khiển động cơ

pinMode**(**11**,**OUTPUT**);**

lcd**.**init**();** // Khai báo sử dụng LCD

lcd**.**begin**(**16**,** 2**);**

lcd**.**backlight**();**

lcd**.**clear**();**

lcd**.**print**(**" HELLO"**);**

**}**

Chương trình điều khiểu. Ở đây ta khai báo chân điều khiển cho Servo là chân số 10, ngõ ra chân 11 để điều khiển còi Buzzer. Khai báo sử dụng LCD, màu nền, xóa màn hình và hiển thị dòng chữ “HELLO” lên LCD.

void loop**()**

**{**

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

**switch** **(**key**)**

**{**

**case** 'A'**:**

lcd**.**clear**();**

lcd**.**print**(**" Enter Password"**);**

i **=** 1**;**

**break;**

**}**

Bắt đầu chương trình chính. Đầu tiên ta gán ký tự nhập vào từ Keypad cho biến Key. Ta dùng lệnh switch/ case để điều khiển riêng biệt cho mã phím, chỉ khi nào mã phím nhập vào là “A” thì mới cho biến i=1 để cho phép nhập mật khẩu và LCD in ra dòng “Enter Password” để thông báo là được phép nhập mật khẩu.

**if** **(**key **&&** i**>**0**)** // Nhập mật khẩu

**{**

**if** **(**i **==** 1**)**

**{**

str**[**0**]** **=** key**;**

lcd**.**setCursor**(**6**,** 1**);**

lcd**.**print**(**str**[**0**]);**

delay**(**1000**);** // Ký tự hiển thị trên màn hình LCD trong 1s

lcd**.**setCursor**(**6**,** 1**);**

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

**}**

**if** **(**i **==** 2**)**

**{**

str**[**1**]** **=** key**;**

lcd**.**setCursor**(**7**,** 1**);**

lcd**.**print**(**str**[**1**]);**

delay**(**1000**);**

lcd**.**setCursor**(**7**,** 1**);**

lcd**.**print**(**"\*"**);**

**}**

**if** **(**i **==** 3**)**

**{**

str**[**2**]** **=** key**;**

lcd**.**setCursor**(**8**,** 1**);**

lcd**.**print**(**str**[**2**]);**

delay**(**1000**);**

lcd**.**setCursor**(**8**,** 1**);**

lcd**.**print**(**"\*"**);**

**}**

**if** **(**i **==** 4**)** **{**

str**[**3**]** **=** key**;**

lcd**.**setCursor**(**9**,** 1**);**

lcd**.**print**(**str**[**3**]);**

delay**(**1000**);**

lcd**.**setCursor**(**9**,** 1**);**

lcd**.**print**(**"\*"**);**

count **=** 1**;**

**}**

i **=** i **+** 1**;**

**}**

Ta dùng lệnh If để kiểm tra điều kiện, khi có phím nhấn với biến i>0 hay thì cho phép nhập mật khẩu. Mật khẩu có 4 số tương ứng với i=1,2,3,4. Với mỗi i có một cấu trúc If thực hiện nhập mật khẩu, ví dụ i=1, mã phím Keypad sẽ được lưu vào vị trí “0” của mảng str[ ], và in giá trị mã phím ra lên LCD ở vị trí cột thứ 7 hàng thứ 2,sau khoảng thời gian một giây chuyển thành “\*” để bảo vệ tính bảo mật cho mật khẩu. Sau mỗi lần nhấn phím biến i sẽ tăng lên một đơn vị (i++) để chuyển qua lệnh If tiếp theo, tại cấu trúc If với i=4 sau khi nhập xong mật khẩu biến count sẽ bằng 1 để cho phép thực hiện chương trình kiểm tra mật khẩu.

**if** **(**count **==** 1**)**

**{**

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

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

lcd**.**clear**();**

lcd**.**print**(**" Correct!"**);**

delay**(**2000**);**

myServo**.**write**(**180**);** // Mở cửa

lcd**.**clear**();**

lcd**.**print**(**" Opened!"**);**

i **=** 0**;**

count **=** 0**;**

**}**

**else** **{**

lcd**.**clear**();**

lcd**.**print**(**" Incorrect!"**);**

delay**(**3000**);**

lcd**.**clear**();**

lcd**.**print**(**" Try Again!"**);**

delay**(**3000**);**

lcd**.**clear**();**

lcd**.**print**(**" Enter Password"**);**

i **=** 1**;**

count **=** 0**;**

cb**++;**

**}**

**}**

Khi biến count=1, ta dùng lệnh If để thực hiện so sánh từng giá trị với nhau của hai mảng str[] và STR[]. Nếu từng giá trị của cả hai mảng giống nhau tức là mật khẩu nhập vào là đúng sẽ xuất ra LCD dòng chữ “Correct”, delay 2s, dùng lệnh myServo**.**write() để điều khiển góc quay của Servo mở chốt cửa. Sau khi mở chốt của LCD hiển thị “Opened”, reset biến i và biến count về 0.

Nếu mật khẩu nhập vào là sai, LCD hiển thị “Incorrect” delay 3s LCD hiển thị “Try Again” delay 3s LCD hiển thị “Enter Password” đồng thời set biế i về 1 và biến count về 0, cho phép nhập lại mật khẩu đồng thời tăng biến cảnh báo lên một đơn vị (cb++);

**if(**cb**==**3**)**//chỉ có 3 lần sai, tới lần 3 vô hiệu hóa khóa.

**{**

lcd**.**clear**();**

lcd**.**print**(**"VO HIEU HOA"**);**

lcd**.**setCursor**(**3**,** 1**);**

lcd**.**print**(**"1 phut'"**);**

digitalWrite**(**11**,**HIGH**);**

delay**(**60000**);**

digitalWrite**(**11**,**LOW**);**

cb**=**0**;**

i**=**0**;**

count **=** 0**;**

lcd**.**clear**();**

lcd**.**print**(**" HELLO"**);**

**}**

Nếu nhập sai mật khẩu 3 lần liên tiếp, biến cb=3. LCD hiển thị “VO HIEU HOA” ở hàng thứ nhất, “1 phut” ở hàng thứ hai, chân 11 ngõ ra mức cao để bật còi Buzzer, delay 1 phút, chân 11 ngõ ra mức thấp để tắt còi, reset các biến về 0, LCD hiển thị “HELLO”.

**switch** **(**key**)**

**{**

**case** '#'**:**

lcd**.**clear**();**

myServo**.**write**(**0**);**

lcd**.**print**(**" Closed!"**);**

delay**(**5000**);**

lcd**.**clear**();**

lcd**.**print**(**" HELLO"**);**

i **=** 0**;**

**break;**

**}**

**}**

Trong lúc đang thực hiện chương trình, nếu nhấn phím “#” của Keypad, Servo sẽ quay về góc 0 độ, LCD hiển thị “Closed” delay 5s, LCD hiển thị “HELLO”, đưa chương trình quay về ban đầu, reset tất cả các biến về 0.

# *Chương 4:*

**KẾT QUẢ THỰC HIỆN – HƯỚNG PHÁT TRIỂN**

* 1. **KẾT QUẢ THỰC HIỆN**

Qua quá trình nghiên cứu và thực hiện đề tài, nhóm sinh viên thực hiện đề tài nhận thấy đề tài này có khả năng ứng dụng vào thực tế cao, đáp ứng được nhu cầu ngày càng cao của con người và đi đúng xu hướng của thời hiện đại. Đồng thời đề tài này cũng là nguồn tài liệu có giá trị cho những bạn sinh viên khóa tiếp theo có thể tham khảo khi nghiên cứu những đề tài liên quan. Bên cạnh đó chúng em cũng bổ sung cho mình những kiến thức hay và bổ ích.

Biết cách sử dụng Arduino Uno: Arduino Uno là một trong những board mạch thông minh được sử dụng rộng rãi. Tuy nhiên, để sử dụng thành thạo và đầy đủ các chức năng của nó thì không dễ. Trong đề tài này nhóm thực hiện khả năng hiển thị thông tin theo yêu cầu trên LCD 16x2, nhận thông tin và giải mã để sử dụng từ Keypad 4x4, điều khiển cơ bản động cơ Servo. Từ đó nhóm phát triển ý tưởng làm khóa thông minh từ Arduino.

Biết cách sử dụng LCD 16x2 thông qua Arduino: LCD là một màn hình hiển thị đơn giản, giá thành rẻ và dễ sử dụng. Qua quá trình nghiên cứu, thông qua chuẩn giao tiếp I2C được điều khiển bởi Arduino, nhóm đã biết cách sử dụng LCD để hiển thị những thông tin mong muốn.

Biết cách sử dụng Keypad 4x4: Keypad 4x4 thực chất là một ma trận phím nhấn với 4 hàng 4 cột 16 phím nhấn. Qua quá trình nghiên, nhóm đã cách dùng Arduino để nhận dữ liệu phím nhấn từ Keypad sau đó sử lý mã phím để dùng cho với mục đích mong muốn.

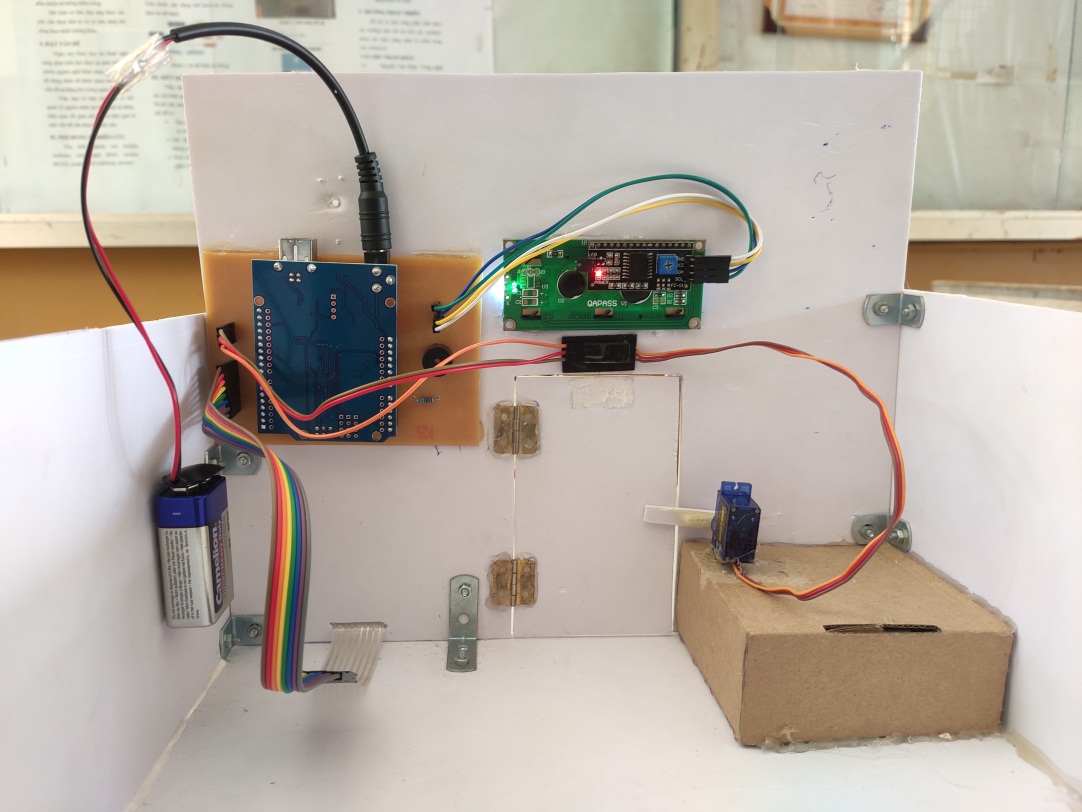
Nhận xét: Sau thời gian 15 tuần nghiên cứu và thực hiện đề tài, mạch cơ bản đã đáp ứng được yêu cầu thiết kế ban đầu. Sau đây là một số nhận xét:

* Ưu điểm: Giao diện thiết kế dễ sử dụng, tháo tác điều khiển đơn giản, thay đổi được mật khẩu tùy ý, mạch chạy đúng với yêu cầu,
* Hạn chế: Chỉ điều khiển được một chiều từ bên ngoài để mở cửa vào, chưa thể cảnh báo được cho chủ nhà khi ở xa.

**MÔ HÌNH MẠCH KHÓA SỐ ĐIỆN TỬ HOÀN THIỆN**



Hình 27 Mặt trước của sản phẩm



Hình 28 Mặt sau của sản phẩm

**TÀI LIỆU THAM KHẢO**

1. **Sách hướng dẫn:**

[1] Nguyễn Đình Phú, “Giáo trình vi xử lý – Vi điều khiển pic”, NXB Trường ĐH Sư Phạm Kỹ Thuật Tp.HCM, 2016.

[2] Phạm Quang Huy và Lê Cảnh Trung, “Lập trình điều khiển với Arduino”, NXB Khoa Học Kỹ Thuật, 2014.

[3] Nguyễn Văn Hiệp, “Giáo trình – Lập trình Android”, NXB Đại học Quốc gia Tp.HCM, 2014.

1. **Trang web tham khảo:**

[1] Cộng đồng Arduino Việt Nam, “Giới thiệu Arduino Uno R3”, 22/5/2014, <http://arduino.vn/bai-viet/42-arduino-uno-r3-la-gi>

[7] Cộng đồng Arduino Việt Nam, “Chuẩn giao tiếp I2C”, 08/02/2015,

<http://arduino.vn/bai-viet/369-giao-tiep-i2c-va-su-dung-module-realtime-clock-ds1307>

[9] Diễn đàn Arduino Việt Nam, “Động cơ RC Servo 9G ”, 19/06/2014,

<http://arduino.vn/tutorial/1423-avr-dieu-khien-dong-co-servo-sieu-chuan-voi-bien-do-goc-cuc-nho>