Simple Arduino Smart Door Locking System

Final project, **CS 207**, *University of Regina*, by Patel Parth Maheshkumar Student# 200420162, Patel Keyur Kalpeshkumar Student# 200422540, Shah Jinal Rakeshkumar Student# 200430734

I. Introduction

In the past few years, security has become a major concern for everyone. Especially, for parents who are both working and need to take help from baby sitters, care takers and/or kids who have grown up enough to stay alone at home. These parents have a constant fear of someone breaking in their locks and harming their children. Apart from that, many business owners and big home owners have expressed concerns about needing a better security system. Therefore, we have developed a lock system that is not only smart and digital but also efficient and elegant.

Previously there have been incidents where a house was broken into because they had simple key-based locks which could be easily broken. There was a need of an added layer of security over the key locks or completely replace the physical keys with digital keys that cannot be replicated or broken into easily. There are high end digital locks available in the market but they are so costly that only companies and large property owners can afford it, plus, they require installation by experts and regular service checks. The aim was to develop a lock system, which is digital and efficient, available in a price range that would be affordable to everyone and not just by extremely wealthy people.

In this paper, a detailed description of the smart lock system developed as part of the final project for course CS 207 at the University of Regina, taken under the guidance of Prof. Trevor Tomesh, is given. The project is a modification of an existing smart lock based on Arduino which uses a keypad for authentication, the base project *"Door lock system with Arduino"* is created on https://create.arduino.cc/projecthub by the user *jayesh_nawani* . As an improvement to the project, RFID reader and tags are used for authentication, along with a solenoid lock that can be used in real life. Further information on the base project and the modifications have been provided in the next section.

II. Background and Modification

The base project consists of Arduino as the controller, a numeric keypad for entering the pass-key and a motor that represents opening of the lock. The project is simple enough, if the pass-key entered on the numeric pad is correct, the motor rotates, indicating the opening of the door. However, if the entered pass-key is wrong the door will not open. Also, there exits an LCD (16x2) display to indicate the current state of lock.

To improvise on that project, certain modifications were decided upon. Firstly, a motor is a representation of a lock but does not ensure that an actual lock can be controlled by Arduino. Thus, we found a solenoid lock that can completely be controlled by Arduino. This solenoid lock is made of metal, just like regular key locks and can be used in real world applications. The only way this lock can be opened or closed is through Arduino. So, there is no concern of a thief using any simple methods to break in.

Secondly, the numeric keypad does not serve the purpose for door locks. The keypad used, was a basic wire matrix key pad which can easily be damaged or broken, requiring regular service checks. In addition, if the home owner wishes to provide temporary access to an individual, they will have to share the passcode. The problem with passcode is that there is no limit to how many people can use it at once. For instance, if an individual with temporary access shares the passcode to 10 people, all of them can enter the house whenever they want. However, when this numeric keypad is replaced by RFID reader, only those people can enter the house who have a valid RFID card. This restricts the access to the people who are meant to have the access. When the owner wishes to revoke the access, they simply have to take the valid RFID card back, without worrying about making any changes in the code. In comparison, if a numeric keypad is used, the owner will have to change the passcode in order to revoke access and then share the new passcode with others who already had access.

The common features between the base project and this project are the presence of LED, Buzzer and LCD screen. These are used to indicate the current state of the lock, approval and denial of access, and alarm for potential unauthorized entry.

The above-mentioned modifications ensure that this smart lock system is realistic and the holds scope of being turned into a product with the correct product design. The upcoming section will explain preparations required to build this project.

III. Build Preparations

The below mentioned libraries of Arduino IDE have been used in this project:

- MFRC522: This library can be added through Library Manager of the Arduino IDE, the documentation for the same is provided at https://github.com/miguelbalboa/rfid
- SPI: This library is available in the standard Arduino installation package, the documentation can be obtained from https://www.arduino.cc/en/Reference/SPI

The hardware requirements are as follows:

- **Arduino Uno** – The Arduino works as the controlling system, the code runs on it, input from sensors is taken and evaluated, actuators are sent instructions to perform particular task. Other versions of Arduino can be used as long as the wire connections are modified accordingly. Can be purchased from various sources including:
- **RFID sensor** – The RFID reader is the sensor that reads the value from the RFID cards and tags. RFID works on radio frequencies; therefore, they require to be in close contact with the cards, ensuring that it does not pick up any card or tags readings from far away.
- **RFID tags** – These tags can be little FOPs or Cards; they hold a specific alpha-numeric value in them which work as unique identifiers for them.
- **Relay board** – A relay board is a set of relay switches that can work independently. A single relay switch works exactly like switch for a light. It connects the power source and the actuator when turned on. These relay switches can be toggled using electrical signals sent to them from Arduino.
- **Solenoid lock** – These are metal locks that work as latches; they can be toggled using electric signals as they work on the concept of electro-magnetism. This lock is chosen so that it can be completely controlled by Arduino.
- **Buzzer and LED** – These are basic actuators, used to send audio and visual confirmations of the process state, for instance the LED turns on when the lock is opened.
- **Other basic components** – Jumper wires for connecting different pieces of hardware, 12V battery to control the solenoid lock and power cable for Arduino.

The appendixes contain pictures of each hardware component, making it easy to identify each of them, along with the schematics of project. Let’s take a look at the design and working of the project in the next section.

IV. Design & Build

The connections for this project can be a little confusing as they are components that use multiple wires and connections. To make it easier, the connections, programming and testing was done for each component individually. The appendix has tutorial connection diagrams for each component, however, to incorporate multiple devices changes have been made in terms of the pins on which connections are made. Also, one can refer to the code to find out which pins were used for which connections, careful commenting has been done.

Firstly, the RFID reader was connected to Arduino as it is the main component of the project. The reader sends an alpha-numeric value to the Arduino when a RFID card or FOP is tapped onto the reader. This alpha-numeric code is then compared with the codes that have been stored in code as a static value. If the code matches exactly then the door is opened, if not then the door is not opened. Once this interfacing was completed, then other components were added.

Secondly, the LCD (16x2) screen is interfaced to the Arduino, since it occupies highest number of data pins. The method of sending string input to LCD is as simple as printing something on the serial monitor, for instance lcd.print("Access Denied :("); . When the correct RFID card is scanned on the RFID reader, a message saying "Authorized access" is displayed on the screen. If wrong card is scanned, then "Access Denied" message is displayed.

Once the RFID and LCD were working, the basic code of the project could be tested, which means correct RFID cards were being identified. This indicated that as a next step we could interface the solenoid lock. The lock is a little tricky to be interface. The lock works on 12V signals. Arduino by default can only send 3.3V to 5V of signals, which means that the regular digital output pins on Arduino are not capable of controlling the lock. For this, a 12V battery was used, but the battery cannot be directly connected to the lock, as there would be no control mechanism in place. Therefore, relay board was used. A relay can connect the power source (12V battery) to the actuator(lock). The interesting fact of relay is that it can be toggled using a 5V signal from Arduino. To simplify, when the door is required to be opened, Arduino sends a 5V signal to relay, the relay in turn completes the connect between the 12V battery and the solenoid lock, resulting in the lock being open. When the lock is to be closed an invert signal is send to relay, resulting in the disconnect of battery and lock. This ensures that whenever Arduino sends a signal, the lock will toggle via the relay. In this project a relay board is used, as it makes the connections easy, in real world, a single relay would be used, this might require some changes in the connections.

Finally, the buzzer and LED can be interfaced. The job of LED and buzzer is solely to provide Audio-Visual confirmation of access approval or denial. The LED is used to indicate the approval,

whenever a correct RFID card is scanned, the LED is turned on. On the other hand, the buzzer is used was indicating unauthorized RFID card by sending tone signals. Which means, whenever a wrong RFID card is scanned, the buzzer will beep, working as an alarm.

V. User Manual

Once the individual interfacing is done, all the components can then be assembled together and testing can be done. For testing it is recommended to gather multiple RFID cards, for instance, if there are 5 RFID cards are available, 2 of them are to be added as valid and 3 as invalid. In this project only one card is added as valid to make sure it works for only one card of the home owner. More cards can be added any time.

While installing the system. The RFID reader, buzzer and LED are to be kept outside the door; rest of the components need to be inside the house to make sure no tampering can be done with system. It is highly recommended to test the system thoroughly before and after installing it, to ensure that when it is put to actual use there are no glitches in the system.

VI. Milestones and Setbacks

No	**Intended Date**	**Description**	**Status**	**Completion Date**
1	October 31	Components Purchased	Completed	Oct 31 (most of the parts) Nov 5 (Relay, battery)
2	November 10	Initial hardware assembly	Completed	Nov 10(RFID, LCD) Nov 15(Relay, lock) Nov 18(buzzer, LED)
3	November 20	Basic code	Completed	Nov 20(RFID, LCD) Nov 28(Rest)
4	November 30	Testing and code modification	Completed	Nov 30
5	December 5	Full working system	Completed	Dec 5

The delay in purchase of Relay and battery was caused by the lack of knowledge that Arduino will not be able to control the solenoid lock as the base project did not have that. The hardware assembly took more time than anticipated due to the high number of pins required for LCD and RFID. It became difficult to accommodate the remaining components, especially the relay, as the connections for that were complicated. Frequent modifications had to be done to the code as part of trial and error to make sure the relay and lock were working perfectly. Apart from the above mentioned, there were no other setbacks during the development of this project.

For future enhancement, the Arduino can be connected to the internet, so that every entry in the house can be logged and stored into a database along with timestamps. Another benefit of connecting the Arduino to internet is that it would enable the facility of sending a notification to the home owner upon entry to the house. This can help the owner to be aware of the entry and if possible, check any security cameras installed at the entrance. This project can be extended to become a completed home security system. After a point, the Arduino Uno might now have enough

data pins to support all enhancements, at that stage the Uno can be replaced with a Mega or even a Raspberry pi if the extensions are too big.

VII. Conclusion

All in all, the old key-based locks have been proved to be vulnerable to break ins. Supporting the fact that smarter home systems are required to be put at work. The first step would be to use a smart lock that can be controlled by simple yet efficient technologies like RFID, NIC and even mobile application control systems. For lower budgets, the project described in this paper is ideal for replacing key-based locks. This would not only ensure more security but has the possibility of having a full version of home security system. The home owners, more importantly working parents need to install such systems to have a more stress-free life.

Appendix A

Breadboard diagram, Wiring schematic and Wiring Tutorial

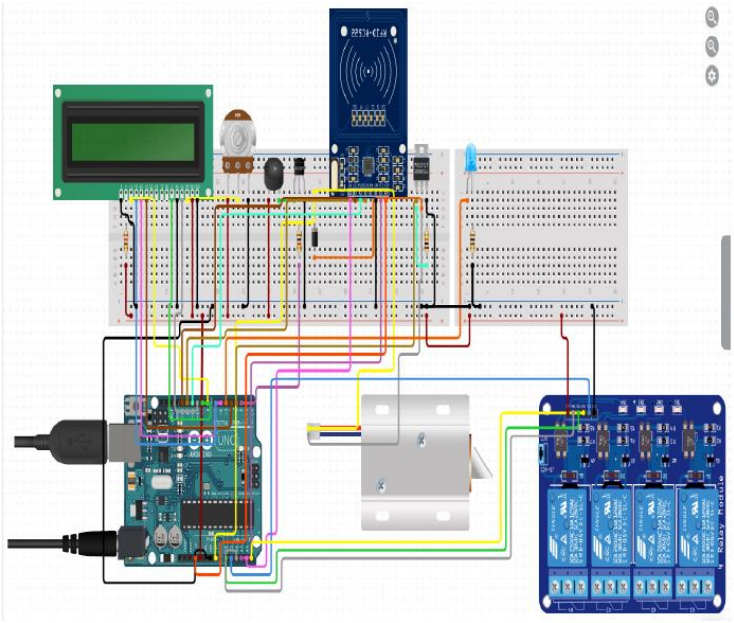


Fig 1 Breadboard Diagram

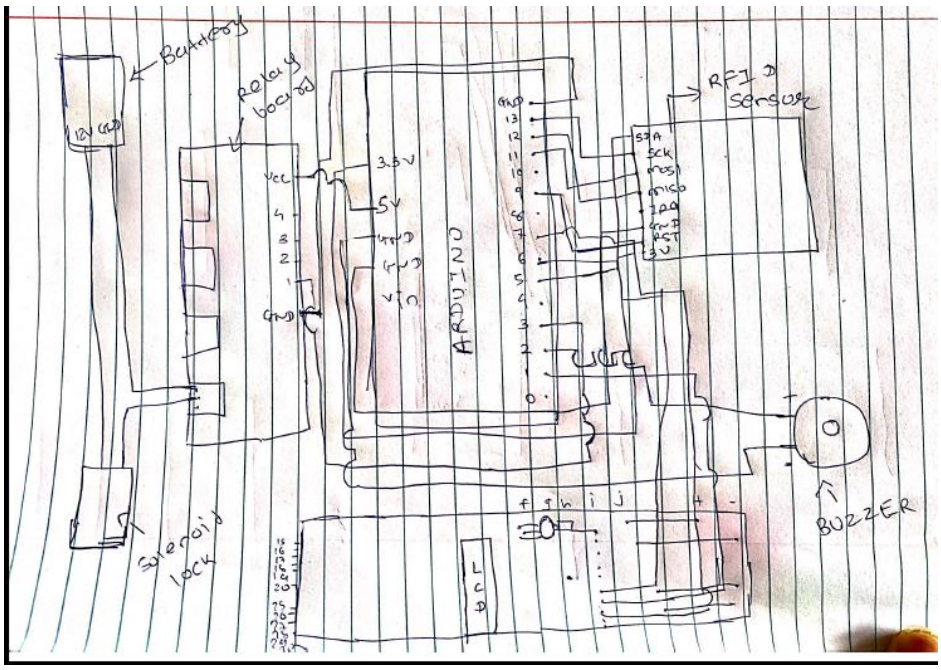


Fig 2 Schematic

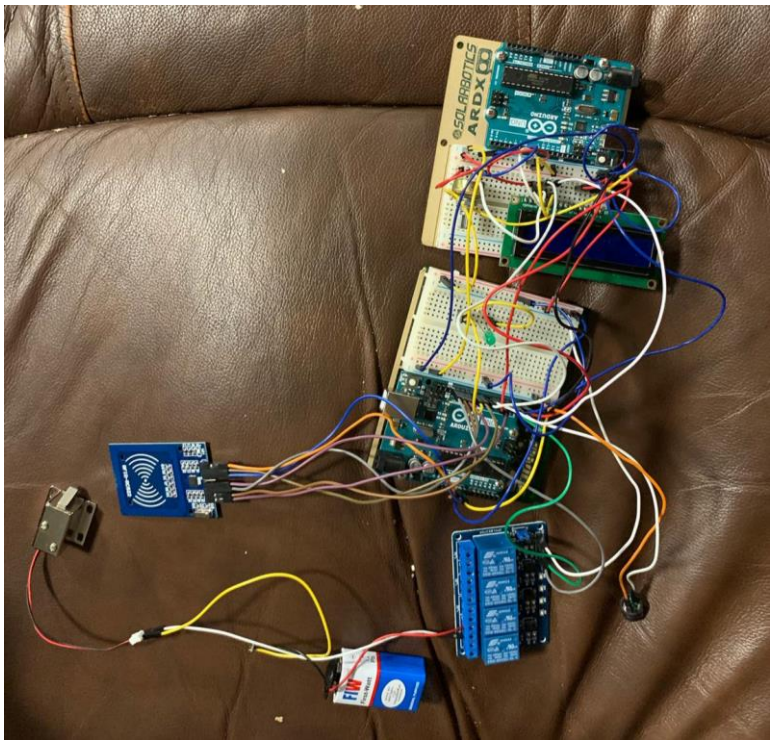
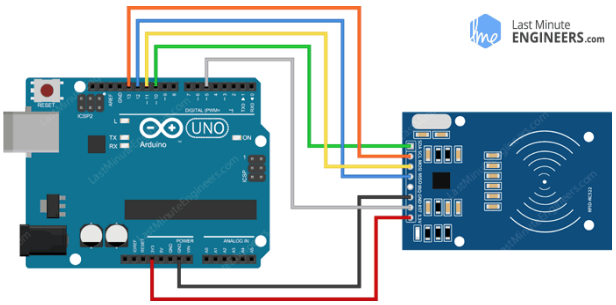
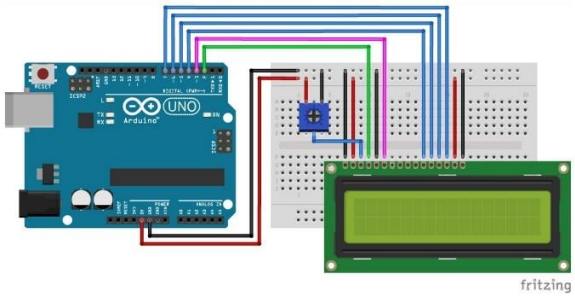


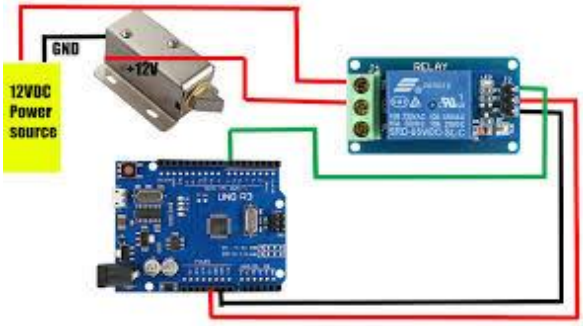
Fig 3 Final project Assembly



Source: https://lastminuteengineers.com
Fig 4 RFID connection



Source: https://www.makerguides.com
Fig 5 LCD connection diagram



Source: https://forum.arduino.cc/index.php?topic=567979.0
Fig 6: Relay connection

Appendix B

Project Code

```
#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN 5
#define RST_PIN 9
#define RELAY 3 //connect the relay to number 3 pin
#define BUZZER 2 // connect the buzzer to 2 pin
#define ACCESS_DELAY 2000
#define DENIED_DELAY 1000

//LED Light
const int pwm = 10 ;

//LCD
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 8, en = 7, d4 = 6, d5 = 4, d6 = 1, d7 = 0;
```

```
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);



MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.

void setup()
{
  lcd.begin(17, 2);
  //Serial.begin(9600);   // Initiate a serial communication
  SPI.begin();          // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522
  pinMode(RELAY, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  noTone(BUZZER);
  digitalWrite(RELAY, HIGH);

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

  //LCD
    // set up the LCD's number of columns and rows:

  // Print a message to the LCD.
 //LED light
 pinMode(pwm,OUTPUT) ;  //Set pin 2 as output
}

void loop()
{
   // Look for new cards
  if ( ! mfrc522.PICC_IsNewCardPresent())
  {
    return;
  }
  // Select one of the cards
  if ( ! mfrc522.PICC_ReadCardSerial())
  {
    return;
  }
  //Show UID on serial monitor
  //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) == "1A 66 FD 84") // enter your own card number after copying
it from serial monitor
  {
    //Serial.println("Authorized access");
 lcd.print("Access Success:)");
    //Serial.println();
    delay(500);
    digitalWrite(RELAY, LOW);
    delay(ACCESS_DELAY);
    digitalWrite(RELAY, HIGH);

     analogWrite(pwm,2500) ;     //setting pwm to 25
     delay(50) ;      //delay of 50 ms
    lcd.clear();
  }

 else   {

    //Serial.println(" Access denied");
    lcd.print("Access Denied :(");
    tone(BUZZER, 300);
    delay(DENIED_DELAY);
    noTone(BUZZER);
    lcd.clear();
  }

//LCD
// Turn off the blinking cursor:
  //lcd.noBlink();
  //delay(100);
  // Turn on the blinking cursor:
  //lcd.blink();
  //delay(3000);
}
```

References

- Nawani, J., 2017. *Door Lock System With Arduino*. [online] Arduino Project Hub. Available at: <https://create.arduino.cc/projecthub/jayesh_nawani/door-lock-system-with-arduino-fe95ab> [Accessed 20 October 2020].
- Mukherjee, A., 2016. *Security Access Using RFID Reader*. [online] Arduino Project Hub. Available at: <https://create.arduino.cc/projecthub/Aritro/security-access-using-rfid-reader-f7c746> [Accessed 5 November 2020].
- Shameel, M., 2017. *How To Use A Relay With Arduino*. [online] Arduino Project Hub. Available at: <https://create.arduino.cc/projecthub/muhammed-shameel-k-v/how-to-use-a-relay-with-arduino-e04e3c> [Accessed 10 November 2020].