ABDUL S. RATTU

Home Security System for Automatic Doors

Capstone Design Project Final Report

Spring 2013

School of Engineering

RUTGERS

The State University of New Jersey, USA

May 1st, 2013

ECE 468

Advisor: Prof. Michael Caggiano

Contents

Objective

Design a radio frequency identification "RFID" door lock that will provide advance and fastest accessible security with ease of use for home doors and gates.

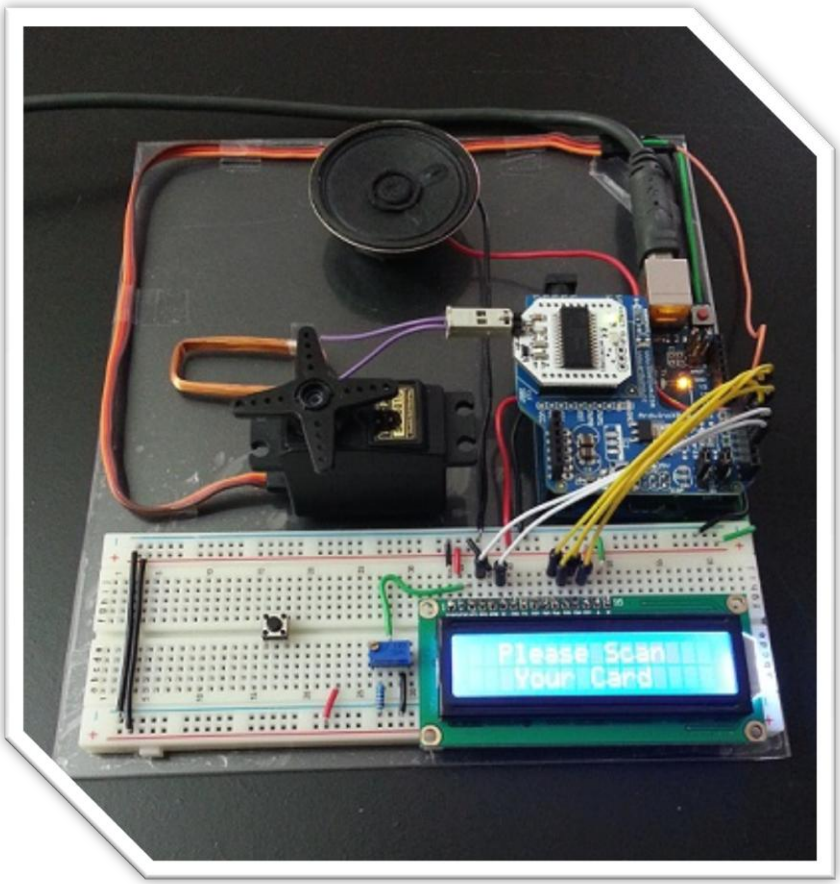
Introduction

Home security system for automatic doors provides advance security of today's standard for home owners. It will be used to open the home doors automatically just by scanning RFID tags wirelessly. In this project I worked to create a security system which is utilizing RFID technology to scan RFID keys at the door knob for automatic door lock purposes. This system is also connected with servo motor to perform a function of physical lock as well as there is a liquid crystal display attached which shows the welcome message on screen in case of authorized and unauthorized user attempts. This type of systems can be used in any type of doors and gates for high class and fastest accessible security with ease of use.

Motivation

Nowadays most hotel room doors uses a RFID card to unlock the door, however these RIFD cards are not as convenient as this design project. Usually you have to insert the RFID card into hotel room door and wait for green signal, after the green signal you still need to manually rotate the door knob in order to open the door, however this system is very convenient and different than today's standards. In this system user will wirelessly scan the card at the RFID antenna and door knob will rotate automatically for the user. It will also display the name of the person on the screen and welcome sound through the speaker.

Design (how it works)



Users will have significantly portable RFID tags that they will scan at RFID antenna. Software integrated hardware will determine if the user is authorized or not and then system will proceed accordingly.

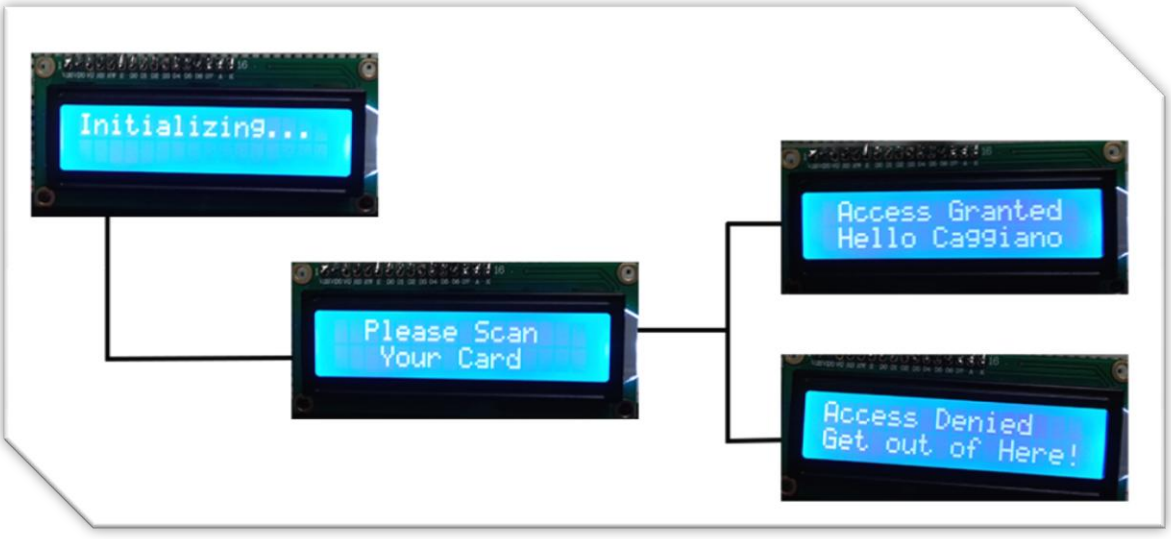
There are different scenarios about what happens if users will try to use this system. These scenarios are defined below,

Scenario 1

In the case where you are authorized to open the door, you will need to scan your card at the RFID antenna, it will then read your card's information and send it to microcontroller which will have understanding of how to distinguish who is authorized and who is not. If you are authorized it will rotate the servo motor for an angle of 90 degrees (or whatever will be the rotation requirements for practical purposes) and keep in that position for 5 seconds after that motor will rotate back to its previous position if no one will attempt to go inside so door will be locked again which is an essential part since we do not want door remains open for long time even after an authorized attempt.

Scenario 2

In the case where you are not authorized, after scanning an unauthorized card, screen will display a message that you are unauthorized, hence servo motor will not have any rotations at all. However if you will try to scan different cards to break in the system, an attached speaker will start making noise in response to unauthorized attempts.



Software

System is programmed into Arduino IDE Environment and using Arduino UNO Microcontroller. Arduino UNO is also attached with X-bee wireless shield & RFID module. Breadboard is being used to connect LCD panel and Speaker with Arduino Microcontroller.

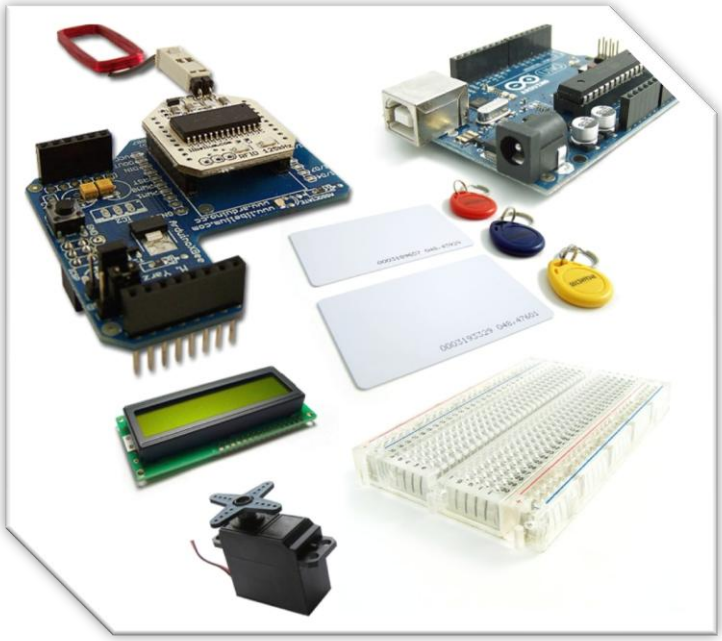
One of the important components of this project is software code. Software code of this system taught Arduino Microcontroller to run this security system according to requirements. Program coding also enables microcontroller to learn about the different devices that are attached with it, their nature and purposes such as Servo Motor, LCD panel and Speaker. Program code is also being used in this project for effective communications and fast processing between hardware components that are connected to microcontroller such as RFID module with reader antenna, X-bee Shield and components on the breadboard such as Servo Motor, LCD panel and Speaker.

The another important role that software is playing here is the security concerns. The software has very strong security that is not easy to break and read the RFID tag numbers. These numbers are being processed in the module but they are severely encoded. It is definitely not easy to decode this encryption and read the tag numbers. This software is cleaver enough to determine who is the authorized user and who is the unauthorized user and then it acts accordingly.

This software has another cool feature of displaying messages on the screen that is connected with hardware. The formatting of the screen has been done neatly. There are no difficulties one can have in order to read and understand the message of the screen. It is very user friendly.

Hardware

Below are the hardware parts that will be the part of this projects.



- **Arduino Microcontroller:** A microcontroller that can be attached with breadboard and functioned accordingly. It's programmable and comes with its own programming language called "Arduino IDE Environment" which is based on Java/Verilog/VHDL and mixture of other script languages.
- **RFID Reader/Module:** A microchip which can be attached to Arduino to enable radio frequency identifications by generating magnetic fields. It can be used to read/write RFID cards/tags/keys.
- **RFID Antenna:** This will help generating magnetic field within specific area. Also this is being used to read RFID cards/tags/keys.

- **RFID Cards and Tags:** These are the cards/tags that can be read by RFID Antenna and use as security keys. They store globally unique digital key in them.
- **X-bee Shield:** This is a shield that will enable Arduino microcontroller to communicate wirelessly.
- **Breadboard:** A board that helps creating electrical projects with electrical components. It is being used to connect speaker and LCD panel with microcontroller.
- **8 Ohm Speaker:** Breadboard compatible speaker that can make noise for security purposes.
- **Servo Motor:** A motor that represents a function of door knob. This is an essential hardware being used as an actual physical lock.
- **LCD Screen:** A liquid crystal display screen which is being used to display message to users about authorized and unauthorized attempts.
- **Resistors & LEDs etc:** These are other electrical components which are being used in this project.

Software Code:

Below is the full code of this project. I used comments in programming language notation of "//" within the code to explain the parts of the code. This code has been written in Arduino IDE environment. It consists of two different files, one file is the main Arduino code file which is actually being uploaded to Arduino and the other file is being incorporated in to main file which has some melodies defined for authorized and unauthorized attempts.

----------Code Started----------

```
#include <Servo.h>
#include <LiquidCrystal.h>
#include "pitches.h"

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);          //Assigning arduino pins to LCD
byte data[5];                                 // Here I am holding the ID
int val = 0;
byte sami[5] = {0x01,0x00,0x5C,0xFD,0xFD};
byte caggiano[5]  = {0x01,0x00,0x5D,0x42,0x7F};
int j_melody[] = {NOTE_G4,0,NOTE_A4,0,
NOTE_B4,0,NOTE_A4,0,NOTE_B4,0, NOTE_C5,0};
int j_noteDurations[] = {8,8,8,8,8,4,8,8,8,8,8,4};
int d_melody[] =
{NOTE_C4,0,NOTE_D4,0,NOTE_F4,0,NOTE_D4,0,NOTE_F4,0,NOTE_G4,0};
int d_noteDurations[] = {8,8,8,8,8,4,8,8,8,8,8,4};
int fail_melody[] = {NOTE_G2,0,NOTE_F2,0,NOTE_D2,0};
int fail_noteDurations[] = {8,8,8,8,8,4};
int speaker_pin = 8;          //Assigning arduino pin to Speaker
int servoPin = 9;             //Assigning arduino pin to Servo

//Creating Servo Object here
Servo doorLock;

 void setup()
 {
   //Attaches the Servo to my object
```

```
doorLock.attach(servoPin);

lcd.begin(16, 2);              //Set up the LCD's number of rows and columns
lcd.clear();
lcd.print("Initializing...");
delay (1000);

// Start serial port 19200 bps
Serial.begin(19200);

// Setting Auto Read Mode - EM4102 Decoded Mode - No password
// command: FF 01 09 87 01 03 02 00 10 20 30 40 37
Serial.write(0xFF);          //Header
Serial.write(0x01);           //Reserved
Serial.write(0x09);           //Length (Command + Data)
Serial.write(0x87);          //Command (0x87 sets auto mode behavior
Serial.write(0x01);          //Data 1: Enable Auto-Read
Serial.write(0x03);          //Data 2: Mode – Parity decoded – Manchester RF/64
Serial.write(0x02);           //Data 3: Total number of block to be read (2)
Serial.write((byte)0x00);    //Data 4: No password expected
Serial.write(0x10);           //Data 5: Password byte 1
Serial.write(0x20);           //Data 6: Password byte 2
Serial.write(0x30);           //Data 7: Password byte 3
Serial.write(0x40);           //Data 8: Password byte 4
Serial.write(0x37);           //Checksum

delay(500);
```

```
  while(Serial.available()>0)
  {
    Serial.read();
  }
  Serial.println();
  Serial.println("RFID Started, Waiting for Card...");
}

void loop()
{
  lcd.clear();                    //Clears LCD
  lcd.setCursor(0, 0);
  lcd.print("  Please Scan");
  lcd.setCursor(0, 1);
  lcd.print("   Your Card");

  val = Serial.read();
  while (val != 0xff)
  {                               //On Successful read, first byte will always be 0xFF
    val = Serial.read();
    delay(1000);
  }

  Serial.read();                  //Reserved
  Serial.read();                  //Length
  Serial.read();                  //Command (indicates tag data)
  data[0] = Serial.read();    //Read data 1
```

```
data[1] = Serial.read();    // we read data 2
data[2] = Serial.read();    // we read data 3
data[3] = Serial.read();    // we read data 4
data[4] = Serial.read();    // we read data 5
Serial.read();              // checksum

//Indentifying RFID Card
boolean s_card = true;
boolean c_card = true;
Serial.print("Card found - Code: ");

for (int i=0; i<5; i++)
{
  if (data[i] < 16) Serial.print("0");
  Serial.print(data[i], HEX);

  //Cross-checking
  if (data[i] != sami[i]) s_card = false;
  if (data[i] != caggiano[i]) c_card = false;
}

Serial.println();

if (s_card)
{
  Serial.println("Hello Sami!");
```

```
 for (int i = 0; i < 12; i++)
 {
  int j_noteDuration = 1000/j_noteDurations[i];
  tone(speaker_pin, j_melody[i],j_noteDuration);
  int j_pauseBetweenNotes = j_noteDuration * 1.30;
  delay(j_pauseBetweenNotes);
  noTone(speaker_pin);
 }

 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print(" Access Granted");
 delay(1500);
 lcd.setCursor(0, 1);
 lcd.print("   Hello Sami   ");
 delay(1500);
}
else if (c_card)
{
 Serial.println("Hello Caggiano!");

 for (int i = 0; i < 12; i++)
 {
  int d_noteDuration = 1000/d_noteDurations[i];
  tone(speaker_pin, d_melody[i],d_noteDuration);
  int d_pauseBetweenNotes = d_noteDuration * 1.30;
  delay(d_pauseBetweenNotes);
```

```
    noTone(speaker_pin);
  }

 lcd.clear();
 lcd.setCursor(0, 0);
 lcd.print(" Access Granted");
 delay(1500);
 lcd.setCursor(0, 1);
 lcd.print(" Hello Caggiano");
 delay(1500);
}
else
{
  Serial.println("Not Recognized!  Get out of Here!");

  for (int i = 0; i < 6; i++)
  {
    int fail_noteDuration = 1000/fail_noteDurations[i];
    tone(speaker_pin, fail_melody[i],fail_noteDuration);
    int fail_pauseBetweenNotes = fail_noteDuration * 1.30;
    delay(fail_pauseBetweenNotes);
    noTone(speaker_pin);
  }

   lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("Access Denied");
```

```
    delay(1500);
    lcd.setCursor(0, 1);
    lcd.print("Get out of Here!");
    delay(1500);
  }
  //Unlock Door for Authorized Users
  if (c_card || s_card)
  {
      doorLock.write(180);
      delay(5000);
      doorLock.write(90);
  }
  Serial.println();
}
```

----------Code Ended----------

I also wrote code for melodies that are being played in this device. It is being incorporated in main code as well. Below is the code for melodies,

----------Code Started----------

```
 * Defining Public Constants
#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
```

```
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
#define NOTE_G4  392
#define NOTE_GS4 415
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
```

```
#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
#define NOTE_C6  1047
#define NOTE_CS6 1109
#define NOTE_D6  1175
#define NOTE_DS6 1245
#define NOTE_E6  1319
#define NOTE_F6  1397
#define NOTE_FS6 1480
#define NOTE_G6  1568
#define NOTE_GS6 1661
#define NOTE_A6  1760
#define NOTE_AS6 1865
#define NOTE_B6  1976
#define NOTE_C7  2093
#define NOTE_CS7 2217
#define NOTE_D7  2349
```

```
#define NOTE_DS7 2489
#define NOTE_E7  2637
#define NOTE_F7  2794
#define NOTE_FS7 2960
#define NOTE_G7  3136
#define NOTE_GS7 3322
#define NOTE_A7  3520
#define NOTE_AS7 3729
#define NOTE_B7  3951
#define NOTE_C8  4186
#define NOTE_CS8 4435
#define NOTE_D8  4699
#define NOTE_DS8 4978
```

----------Code Ended----------

This is a code that is explaining in comments section that what these bits of code are being used for. In order to write on RFID module we need to use serial communication which has some defined standard that I used from the datasheet and explained them after "//" (comment notation for programming languages)

Conclusions

As almost everything described already for this design, I would like to say there are still numerous kinds of enhancements one can implement on this project to make it even more convenient. Such as, creating a smart phone app to give access to guests when person is away from his/her home. Over all this is an amazing device which can definitely increase the usability of RFID module as well as convenient access for doors.

Specially in hotel environment if users have their own personal RFID tag they can just reserve hotel online and get their tag assigned at home so once they reached at the hotel they do not even have to go to lobby to bring their tag. They can simply go to their assigned room number where they can just scan their RIFD tag and enjoy the benefit of this automation.

Future Work

In future there can be a smart phone application enhancement for this project that is, creating a smart phone app to give access to guests when person is away from his/her home will enhance this project numerously.

The others extensions within the app could also be viewing who is passing through the door by adding the camera to Arduino and sending stream directly to mobile device. This can also extends to, if someone is trying to break in the door, alarm will start beeping on mobile device as well. In short there can be tons of enhancements one can implement in this system as this system is very easy to understand for both consumers and developers.