

Milestone 4

Team Members:

Jacob Diaz (*jdiaz88*), Jamiel Impoy (*jimpoy2*)

Contact:

jdiaz88@uic.edu, jimpoy2@uic.edu

Project Name: Arduino Security System

Abstract:

Our project consists of creating a RFID locking system. This lock will be able to attach to a door and allow a user to lock and unlock the door using a RFID card. Users will be able to see a success or failure message on the locks lcd screen. A valid RFID card will trigger a mechanism to move the locking bolt.

Project Details:

- **Overall Description of Project Idea:**

We will create an RFID Locking system that uses a RFID card for unlocking. A servo motor will be activated when a RFID card is successfully validated. The device will be able to attach to a door or large drawer, or a cardboard box simulating a locking door. Users will be able to see whether or not their RFID card was successful or unsuccessful based on the message displayed on the screen. We have obtained an RFID scanner for the arduino and progressed on the code to implement the RFID scanner to read RFID cards to get input data to determine whether the card is valid or not. The code has pseudo code for the parts still left to implement so we know what to do next. We also have parts for the servo motor that will be used as the locking mechanism and will be wiring that up to the arduino.

- **Initial Project Design:**

Our project will consist of a RFID sensor, RFID card, LCD screen, bolt lock, active buzzer, motor driver, and a servo motor. Our Arduino will receive input from the RFID reader when a RFID card is used. We will interpret the card and determine if the card is valid for unlocking the lock. We will display to the user, on a lcd

screen, if the card has or has not successfully triggered the unlocking mechanism. If the card is valid, the Arduino will output instructions for the unlocking mechanism to occur. The locking mechanism will consist of a servo motor and a wooden bolt. When the lock is in the locked state, the wooden bolt will be extended to prevent the user from opening the door. The bolt will then retract when the device is in the unlocked state. The user then will have 30 seconds before the locking system automatically locks itself once more. We will use either a passive buzzer or a speaker in order to produce a noise or vibration when the device is unlocked successfully.

Expected Plan for Communication:

In general, we will use Discord as the main form of communication. This includes video calls, voice calls, and text conversations. When in the development phase, we will use Git to work on the same code base. Since we live relatively close to one another, we will be able to work together physically, or be able to work together through a video call on Discord. We have been using text messaging to talk with one another and send pictures of the parts we have received. We are also using Github to version control the Arduino code

Original Work Being Attempted:

Though a lock has been invented before, we will be creating a small-scale RFID lock with a mechanism we will be building ourselves. The originality in this project will come from this mechanism as well as the output we will give our user when using the lock. Most locks will not have a LCD screen to display messages, as well as a speaker or buzzer to signify if the user has successfully or unsuccessfully unlocked the device.

How the project is used:

The user would first have to have a valid RFID card. The user would then look at the lcd screen and see that it will ask to scan their card. In order to gain access to what is being locked, the user would have to place the card against the RFID scanner. The LCD screen would then display if the card is valid, which the LCD screen would then say "Access Granted", or invalid, which the LCD screen would then say "Access Denied". The servo the user will then hear an audio cue depending on the validity of the card. If the user's card is valid, then the servo motor unlocks and allows the user up to 30 seconds to open the door and close it before the servo motor locks. If the user's card is invalid, the servo will stay locked. After locking, the LCD will now wait for the next card to be read. Summary of steps:

1. Look at prompt on LCD screen to scan RFID card
2. Place an RFID card on the scanner.
3. LCD display will then show if the RFID card is valid or invalid.
4. If valid, servo motor will unlock and the user will open and close the door within 30 seconds before the servo locks

Supporting Materials

Timeline:

- **Week 7** – Finish Milestone 3 (**COMPLETED**)
- **Week 8** – create a list of parts and start to acquire parts (**COMPLETED**)
- **Week 9** – get any parts and make final changes to the project idea. (**COMPLETED**)
- **Week 10** – Create Fritzing Diagrams for hardware (**COMPLETED**)
- **Week 11** – Create Pseudo Software and set up development environment; finish Milestone 4 (**Completed**)
- **Week 12** - Build Week (*Hardware*)
- **Week 13** - Build Week (*Hardware*)
- **Week 14** - Build Week (*Software*)
- **Week 15** - Prepare for design presentation
- **Week 16** – Work on project improvements and final presentation
- **Week 17** – Last Minute Adjustments / Final Project Presentations

List of Materials Expected to be Needed:

Arduino Uno
9g Micro Servo
Piezo Buzzer
1k Ohm Resistor (1)
220 Ohm Resistor (1)
Transistor - NPN BC337
RFID RC522
LCD 16 x 2
Rotary Potentiometer 10k Ohm
Stereo Speaker
9V Battery
USB Cable A to B
Breadboard
Breadboard half-size
Jumper Wires

Male Header

List of References

<https://fritzing.org/>

<https://www.arduino.cc/reference/en/libraries/servo/>

<https://create.arduino.cc/projecthub/SURYATEJA/use-a-buzzer-module-piezo-speaker-using-arduino-uno-89df45>

<https://www.instructables.com/How-to-Lock-a-ServoMotorRelay/>

<https://create.arduino.cc/projecthub/Aritro/security-access-using-rfid-reader-f7c746>

<https://www.arduino.cc/en/Tutorial/LibraryExamples/HelloWorld>

<https://www.arduino.cc/en/Reference/ArduinoSound>

<https://www.circuito.io>

Github Repository

<https://github.com/jacobdiaz/Arduino-Security-System.git>

Building the project (High Level Overview)

Connecting Power and ground to breadboard

- Place the full sized(FS) breadboard next to the half sized(HS) breadboard, and connect the 5v pin and ground pin to the FS breadboard
- Connect power and ground from FS breadboard to the power and ground rails of the HS breadboard

Components on FS BreadBoard

- Place Buzzer, TSBC337 Transistor, RFID Reader, and 16x2 LCD on FS Breadboard
 - TSBC 337 will need a 220 Ohm resistor on it's middle pin
 - 16 x 2 LCD will need a 1k Ohm resistor on it's pin 15 (A)

Components on HS BreadBoard

- Place Rotary 10k Ohm Potentiometer, and Stereo Speaker on HS breadboard.
- Connect potentiometer to LCD screen

Servo Motor and 9V Battery

- Connect Battery to VIN analog pin on Arduino and to ground on HS breadboard, and place near arduino
- Connect Servo motor to Analog pin A3 on arduino and to ground and power on FS breadboard
- Mount small wooden arm to Servo Motor

Connecting and Mounting

- Connect all components from FS breadboard and HS breadboard to arduino.
 - Pins for LCD: 8,7,3,4,5,6
 - Pins for RFID: 9, 10
 - Pins for Buzzer: 2
 - Pins for Speaker: A4
 - Pins for Servo Motor: A3
- Mount Arduino and breadboards to a square plate
- Place cover over arduino breadboards and battery, leaving servo motor and arm protruding.
- Mark area for RFID card reader on cover
- Create a window for the 16x2 LCD screen on the cover.
- Mount plate near a door and mount servo motor and arm near door edge.

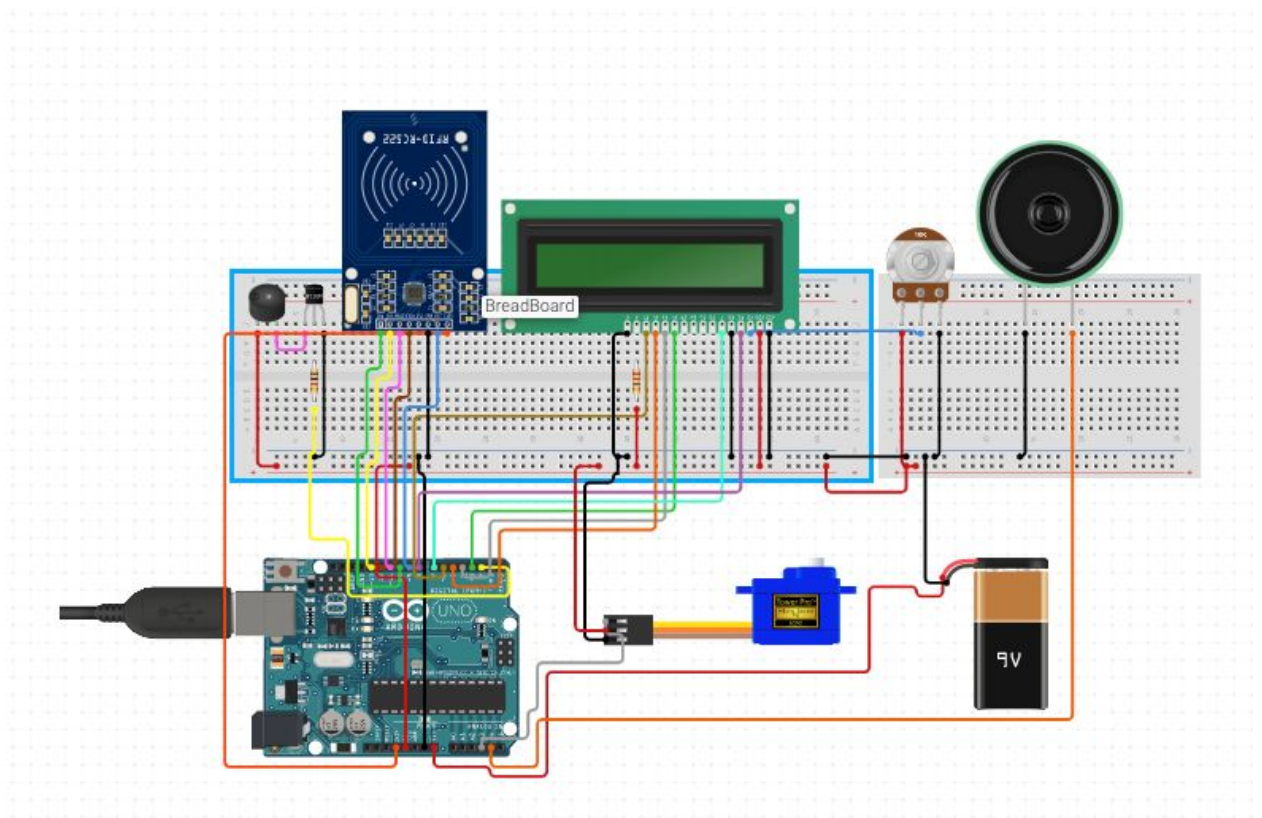
Testing Device

- Test device by swiping RFID card near the area where we marked the RFID card reader.

** Please Refer to hardware diagram for a detailed look at the design **

Hardware

High Fidelity Hardware Design



Software

Below is updated code that has been done so far

AdruinoSecuritySystem.ino

```
Src > ArduinoSecuritySystem.ino > isCardValid()
1 #include "Arduino.h"
2 #include "Buzzer.h"
3 #include "LiquidCrystal.h"
4 #include "RFID.h"
5 #include "Servo.h"
6 #include "SpeakerStereo3W.h"
7
8 #define BUZZER_PIN_SIG 2
9 #define LCD_PIN_RS 8
10 #define LCD_PIN_E 7
11 #define LCD_PIN_DB4 3
12 #define LCD_PIN_DB5 4
13 #define LCD_PIN_DB6 5
14 #define LCD_PIN_DB7 6
15 #define RFID_PIN_RST 9
16 #define RFID_PIN_SDA 10
17 #define SERVO9G_PIN_SIG A3
18 #define STEREOSPEAKER_PIN_POS A4
19
20 // RFID system initializer
21 RFID rfid(RFID_PIN_SDA,RFID_PIN_RST);
22 // LCD initializer
23 LiquidCrystal lcd(LCD_PIN_RS,LCD_PIN_E,LCD_PIN_DB4,LCD_PIN_DB5,LCD_PIN_DB6,LCD_PIN_DB7);
24 // Servo initializer
25 Servo servo9g;
26 // Buzzer
27 Buzzer buzzer(BUZZER_PIN_SIG);
28 // Speaker
29 SpeakerStereo3W StereoSpeaker(STEREOSPEAKER_PIN_POS);
30
31
32 // Global variables
33 const int servo9gRestPosition = 20; //Starting position
34 const int servo9gTargetPosition = 150; //Position when event is detected
35 unsigned int StereoSpeakerHoorayLength = 6; // amount of notes in melody
36 unsigned int StereoSpeakerHoorayMelody[] = {NOTE_C4, NOTE_E4, NOTE_G4, NOTE_C5, NOTE_G4, NOTE_C5}; // list of notes. List length must match HoorayLength!
37 unsigned int StereoSpeakerHoorayNoteDurations[] = {8, 8, 8, 4, 8, 4}; // note durations; 4 = quarter note, 8 = eighth note, etc. L
38 int attempts;
39
40 // States
41 bool locked;
42 bool unlocked;
43 bool unlockedFailed;
44
45 // LCD
46 String sitIdleMessage = "Please scan card";
47 String successMessage = "Access Granted";
48 String failureMessage = "Access Denied";
49
```

```
Src > ArduinoSecuritySystem.ino > isCardValid()
38 int attempts;
39
40 // States
41 bool locked;
42 bool unlocked;
43 bool unlockedFailed;
44
45 // LCD
46 String sitIdleMessage = "Please scan card";
47 String successMessage = "Access Granted";
48 String failureMessage = "Access Denied";
49
50 void printMessage() {
51 // Print message base on state.
52 if (locked && !unlocked) {
53 lcd.print(sitIdleMessage);
54 } else if (!locked && unlocked) {
55 lcd.print(successMessage);
56 } else if (unlockedFailed) {
57 lcd.print(failureMessage);
58 }
59 }
60
61 void unlock() {
62 // unlock mechanism - retract servo motor
63 // Call 30 second timer
64 // lock mechanism
65 myservo.write(10); // unlock
66 }
67 void lock() {
68 // extend servo motor
69 }
70 void readCard() {
71 // isCardValid()
72 // If id card is true... unlock(), printMessage(), playMedia(successmp3)
73 // If not... printMessage() failure and playMedia(failure.mp3), buzz()
74
75 if (!mfrc522.PICC_IsNewCardPresent()) {
76 return;
77 }
78 if (!mfrc522.PICC_ReadCardSerial()) {
79 return;
80 }
81 }
82 bool isCardValid() {
83 // Determine if card is valid
84 }
85 void playMedia(String path) {
86 // Switch case
```

```

72 // If id card is true... unlock(), printMessage(), playMedia(successmp3)
73 // if not... printMessage() failure and playMedia(failure.mp3), buzz()
74
75 if (!mfrc522.PICC_IsNewCardPresent()) {
76     return;
77 }
78 if (!mfrc522.PICC_ReadCardSerial()) {
79     return;
80 }
81 }
82 bool isCardValid() {
83     // Determine if card is valid
84 }
85 void playMedia(String path) {
86     // Switch case
87     // Based on success or failure determine which sound to play
88     // Play the corresponding
89 }
90 void buzz() {
91     // Activate a buzz
92 }
93
94 void setup() {
95     // Setup LCD
96     // Setup servo motor
97     // setup RFID Reader
98     // Setup Buzzer
99     // Setup Speaker
100     Serial.begin(9600);
101     lcd.begin(16,2);
102     rfid.init();
103     servo9g.attach(SERVO9G_PIN_SIG);
104     servo9g.write(servo9gRestPosition);
105     printMessage();
106     Serial.println(sitIdleMessage);
107 }
108
109 void loop() {
110     // if there is a card input readCard()
111 }
112

```