**Code Used:**

#include <pitches.h>

#include <SPI.h>

#include <MFRC522.h>

#include <LiquidCrystal.h>

#include <Servo.h>

#define SS\_PIN 10

#define RST\_PIN 9

#define SERVO\_PIN 5

#define Red\_LED 6

#define Green\_LED 7

#define Buzzer 8

//initialize the library with the numbers of the interface pins

LiquidCrystal lcd(A5, A4, A3, A2, A1, A0);

Servo BoomLock;

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

int melody[] = {

NOTE\_C4, NOTE\_G3, NOTE\_G3, NOTE\_A3, NOTE\_G3, 0, NOTE\_B3, NOTE\_C4

};

// note durations: 4 = quarter note, 8 = eighth note, etc.:

int noteDurations[] = {

4, 8, 8, 4, 4, 4, 4, 4

};

void setup()

{

pinMode(Red\_LED,OUTPUT);

pinMode(Green\_LED,OUTPUT);

pinMode(Buzzer,OUTPUT);

//Servo Connected to pin Digital Pin 5

BoomLock.attach(SERVO\_PIN);

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

lcd.begin(16,2);

SPI.begin(); // Initiate SPI bus

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

Serial.println("WELCOME");

Serial.println();

}

void loop()

{

BoomLock.write(0); //Servo at 0 Position, Door is Closed.

lcd.clear();

lcd.print(" Put Your Card");

// Look for new cards

if ( ! mfrc522.PICC\_IsNewCardPresent())

{

lcd.clear();

lcd.print("Card NOT FOUND!");

lcd.setCursor(0,1);

lcd.print("No Entry");

return;

}

// Select one of the cards

if ( ! mfrc522.PICC\_ReadCardSerial())

{

lcd.clear();

lcd.print("Card NOT FOUND!");

lcd.setCursor(0,1);

lcd.print("No Entry");

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) == "17 4D 3E 63") //change here the UID of the card/cards that you want to give access

{

Serial.println("Authorized access");

Serial.println();

lcd.clear();

lcd.print("Welcome to");

lcd.setCursor(0,1);

lcd.print("College!");

delay(2000);

int i = 0;

while(i < 2)

{

for (int thisNote = 0; thisNote < 8; thisNote++) {

// to calculate the note duration, take one second divided by the note type.

//e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.

int noteDuration = 1000 / noteDurations[thisNote];

tone(8, melody[thisNote], noteDuration);

// to distinguish the notes, set a minimum time between them.

// the note's duration + 30% seems to work well:

int pauseBetweenNotes = noteDuration \* 1.30;

delay(pauseBetweenNotes);

// stop the tone playing:

noTone(8);

}

i = i + 1;

delay(500);

}

delay(1000);

//Now, Open the Barrier with the help of Servo Motor

BoomLock.write(90);

delay(200);

lcd.clear();

lcd.print("Barrier is Open");

lcd.setCursor(0,1);

lcd.print("Now!");

delay(2000);

lcd.clear();

//Give 10 Sec delay to enter into room

//After that door will again close!

for(int i = 10; i > 0; i--)

{

lcd.print("Barrier will close");

lcd.setCursor(0,1);

lcd.print("in ");

lcd.print(i);

lcd.print("HurryUp!");

delay(1000);

lcd.clear();

}

//Now,Barrier is closed and Green LED is Turned-Off.

BoomLock.write(0);

digitalWrite(Green\_LED,LOW);

delay(200);

lcd.clear();

lcd.print("Barrier is Close");

lcd.setCursor(0,1);

lcd.print("Now!");

delay(1000);

}

else {

Serial.println(" Access denied");

lcd.clear();

lcd.print("Card NOT FOUND!");

lcd.setCursor(0,1);

lcd.print("No Entry");

for(int i = 0; i < 7; i++)

{

digitalWrite(Buzzer, HIGH);

digitalWrite(Red\_LED,HIGH);

delay(500);

digitalWrite(Buzzer, LOW);

digitalWrite(Red\_LED,LOW);

delay(500);

}

delay(1000);

}

mfrc522.PICC\_HaltA(); // Stop reading

return 1;

//Put RFID Reader into Halt, until it does not detect any RFID Tag.

} // End of the code