https://www.pes.edu/wp-content/uploads/2019/09/pes_logo.png

**END SEMESTER ASSESSMENT (ESA) B.TECH. (CSE)**

**IV SEMESTER**

**UE20CS252 – MICROPROCESSOR AND COMPUTER ARCHITECTURE LABORATORY**

**PROJECT REPORT**

**ON**

PROJECT TITLE HERE

SUBMITTED BY

**NAME SRN**

1. **CHANDAN KUMAR.S PES2UG20CS804**
2. **UPENDRA.K.BHAT PES2UG20CS920**
3. **VIJAY.J PES2UG20CS815**

**JANUARY – MAY 2022**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**ELECTRONIC CITY CAMPUS,**

**BENGALURU – 560100, KARNATAKA, INDIA**

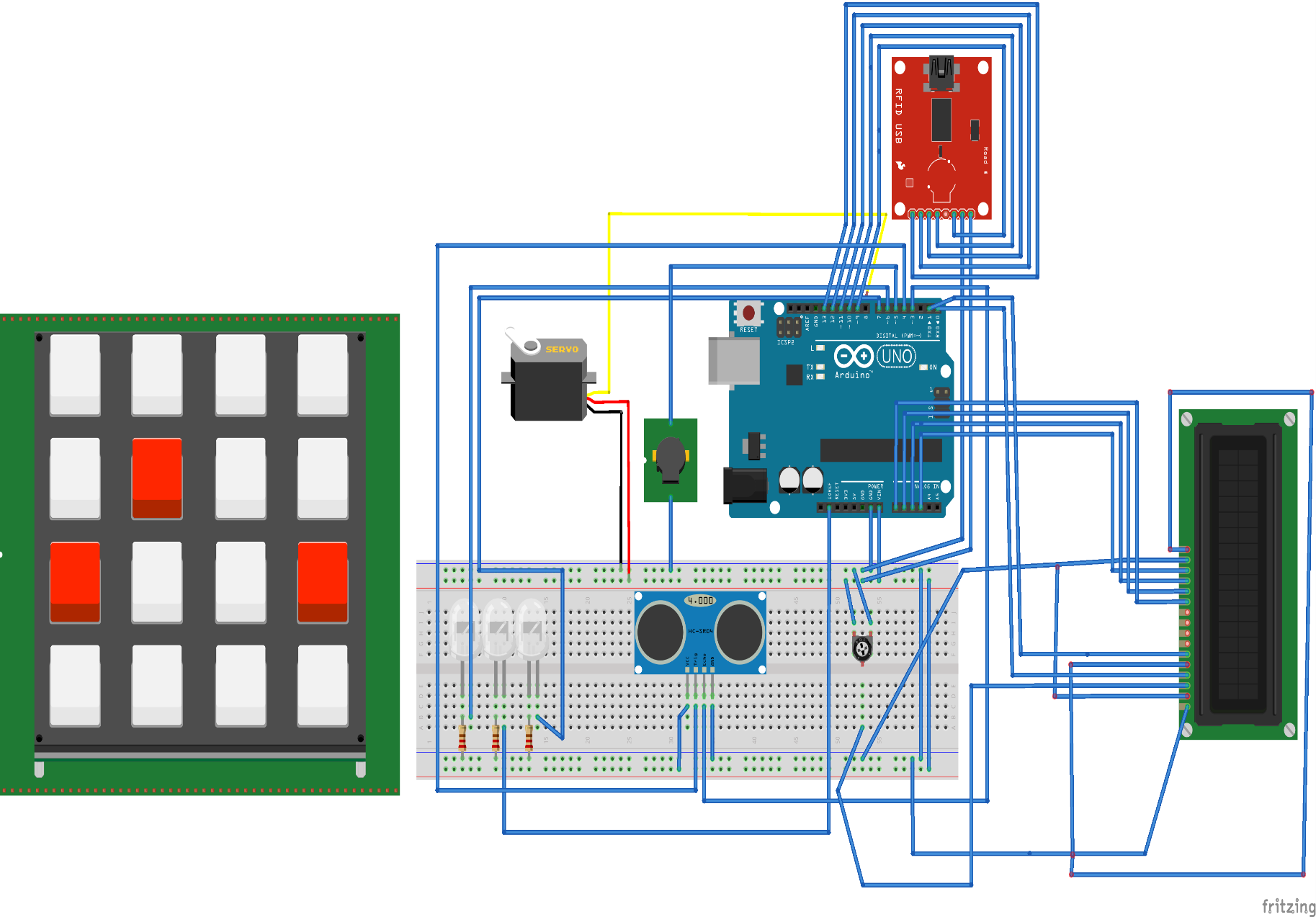
|  |  |  |
| --- | --- | --- |
| TABLE OF CONTENTS | | |
| Sl.No | TOPIC | PAGE No |
|  | ABSTRACT OF THE PROJECT | 1 |
|  | CIRCUIT DIAGRAM | 2 |
|  | ARDUINO CODE | 3-7 |
|  | SCREEN SHOTS OF THE OUTPUT | 8 |

**ABSTRACT OF THE PROJECT:**

* Often times, we need to secure a room at our home or office (perhaps a secret dexter’s laboratory)so that no one can access the room without our permission and ensure protection against theft or loss of our important accessories and assets. There are so many types of security systems present today but behind the scene, for authentication they all relay on fingerprint, retina scanner, iris scanner, face id, tongue scanner, RFID reader, password, pin, patterns, etc.

Whenever the keys are pressed, they are matched with the keys already stored. If the keys that are pressed match the initial password stored in the EEPROM which is ‘1234’, then the lock will open up. If the password does not match, then it will print “access denied” on the LCD. First, connect the 4X4 keypad to the Arduino; connect the first six pins on the 4X4 keypad with the A0 and A5 pins on the Arduino. As security features we are LDR and laser sensor is used so that any intruder cannot enter bypassing the sensors.

**CIRCUIT DIAGRAM:**

****

**ARDUINO CODE:**

// Include required libraries

#include <MFRC522.h>

#include <LiquidCrystal\_I2C.h>

#include <Keypad.h>

#include <Servo.h>

#include <SPI.h>

void pred();

// Create instances

LiquidCrystal\_I2C lcd(0x27, 16, 2);

MFRC522 mfrc522(10, 9); // MFRC522 mfrc522(SS\_PIN, RST\_PIN)

Servo sg90;

constexpr uint8\_t greenLed = 7;

constexpr uint8\_t redLed = 6;

constexpr uint8\_t servoPin = 8;

constexpr uint8\_t buzzerPin = 5;

const int out=2;

const int in=3;

char initial\_password[4] = {'2', '1', '1', '2'}; // Variable to store initial password

String tagUID = "91 71 9D 1B"; // String to store UID of tag. Change it with your tag's UID

char password[4]; // Variable to store users password

boolean RFIDMode = true; // boolean to change modes

char key\_pressed = 0; // Variable to store incoming keys

uint8\_t i = 0; // Variable used for counter

// defining how many rows and columns our keypad have

const byte rows = 4;

const byte columns = 4;

// Keypad pin map

char hexaKeys[rows][columns] = {

{'1', '2', '3', 'A'},

{'4', '5', '6', 'B'},

{'7', '8', '9', 'C'},

{'\*', '0', '#', 'D'}

};

// Initializing pins for keypad

byte row\_pins[rows] = {0, 1};

byte column\_pins[columns] = {16, 17, 18, 19};

Keypad keypad\_key = Keypad( makeKeymap(hexaKeys), row\_pins, column\_pins, rows, columns);

void setup() {

// Arduino Pin configuration

pinMode(buzzerPin, OUTPUT);

pinMode(redLed, OUTPUT);

pinMode(greenLed, OUTPUT);

sg90.attach(servoPin); //Declare pin 8 for servo

sg90.write(0); // Set initial position at 90 degrees

lcd.begin(); // LCD screen

lcd.backlight();

SPI.begin(); // Init SPI bus

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

lcd.clear(); // Clear LCD screen

Serial.begin(9600);

pinMode(in, INPUT);

pinMode(out, OUTPUT);

}

void loop() {

// System will first look for mode

if (RFIDMode == true) {

lcd.setCursor(0, 0);

lcd.print(" Door Lock");

lcd.setCursor(0, 1);

lcd.print(" Scan Your Tag ");

pred();

// Look for new cards

if ( ! mfrc522.PICC\_IsNewCardPresent()) {

return;

}

// Select one of the cards

if ( ! mfrc522.PICC\_ReadCardSerial()) {

return;

}

//Reading from the card

String tag = "";

for (byte j = 0; j < mfrc522.uid.size; j++)

{

tag.concat(String(mfrc522.uid.uidByte[j] < 0x10 ? " 0" : " "));

tag.concat(String(mfrc522.uid.uidByte[j], HEX));

}

tag.toUpperCase();

//Checking the card

if (tag.substring(1) == tagUID)

{

// If UID of tag is matched.

lcd.clear();

lcd.print("Tag Matched");

digitalWrite(greenLed, HIGH);

delay(3000);

digitalWrite(greenLed, LOW);

lcd.clear();

lcd.print("Enter Password:");

lcd.setCursor(0, 1);

RFIDMode = false; // Make RFID mode false

}

else

{

// If UID of tag is not matched.

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Wrong Tag Shown");

lcd.setCursor(0, 1);

lcd.print("Access Denied");

digitalWrite(buzzerPin, HIGH);

digitalWrite(redLed, HIGH);

delay(3000);

digitalWrite(buzzerPin, LOW);

digitalWrite(redLed, LOW);

lcd.clear();

}

}

// If RFID mode is false, it will look for keys from keypad

if (RFIDMode == false) {

key\_pressed = keypad\_key.getKey(); // Storing keys

if (key\_pressed)

{

password[i++] = key\_pressed; // Storing in password variable

lcd.print("\*");

}

if (i == 4) // If 4 keys are completed

{

delay(200);

if (!(strncmp(password, initial\_password, 4))) // If password is matched

{

lcd.clear();

lcd.print("Pass Accepted");

sg90.write(90); // Door Opened

digitalWrite(greenLed, HIGH);

delay(3000);

digitalWrite(greenLed, LOW);

sg90.write(0); // Door Closed

lcd.clear();

i = 0;

RFIDMode = true; // Make RFID mode true

}

else // If password is not matched

{

lcd.clear();

lcd.print("Wrong Password");

digitalWrite(buzzerPin, HIGH);

digitalWrite(redLed, HIGH);

delay(3000);

digitalWrite(buzzerPin, LOW);

digitalWrite(redLed, LOW);

lcd.clear();

i = 0;

RFIDMode = true; // Make RFID mode true

}

}

}

}

void pred(){

long dur;

long dis;

long tocm;

digitalWrite(out,LOW);

delayMicroseconds(2);

digitalWrite(out,HIGH);

delayMicroseconds(10);

digitalWrite(out,LOW);

dur=pulseIn(in,HIGH);

tocm=microsecondsToCentimeters(dur);

//int ldrStatus = analogRead(ldrPin);

if(tocm<20){

Serial.print(tocm);

Serial.print("cm");

//Serial.println(ldrStatus);

digitalWrite(buzzerPin, HIGH);

digitalWrite(redLed, HIGH);

delay(1000);

lcd.clear();

lcd.print("!INTRUDER ALERT!");

digitalWrite(buzzerPin, LOW);

digitalWrite(redLed, LOW);

delay(1000);

lcd.clear();

}

}

long microsecondsToCentimeters(long microseconds)

{

return microseconds / 29 / 2;

}

**SCREEN SHOTS OF THE OUTPUT:**