J S S ACADEMY OF TECHNICAL EDUCATION, BENGALURU

DEPARTMENT OF COMPUTER SCIENCE

Subject : Microcontroller and Embedded Systems (18CS44)

Topic : Finger-print Sensing Door Lock

Submitted by:

Ankit Singh 1JS19CS027

Chirag Verma 1JS19CS046

Deepak Naidu 1JS19CS049

Under the guidance of

Mr. Niranjan C K

Assistant Professor

Department of Computer Science

Content

ABSTRACT

This project on "Biometric

Security" is intended to create an

INTRODUCTION

Biometric systems have overtime served as robust security mechanisms in various domains. Fingerprints are the oldest and most widely used form of biometric identification. The use of fingerprint for identification has been employed in law enforcement for about a century. A much broader application of fingerprint is for personal authentication, for instance to access a computer, a network, an ATM machine, a car or a home.

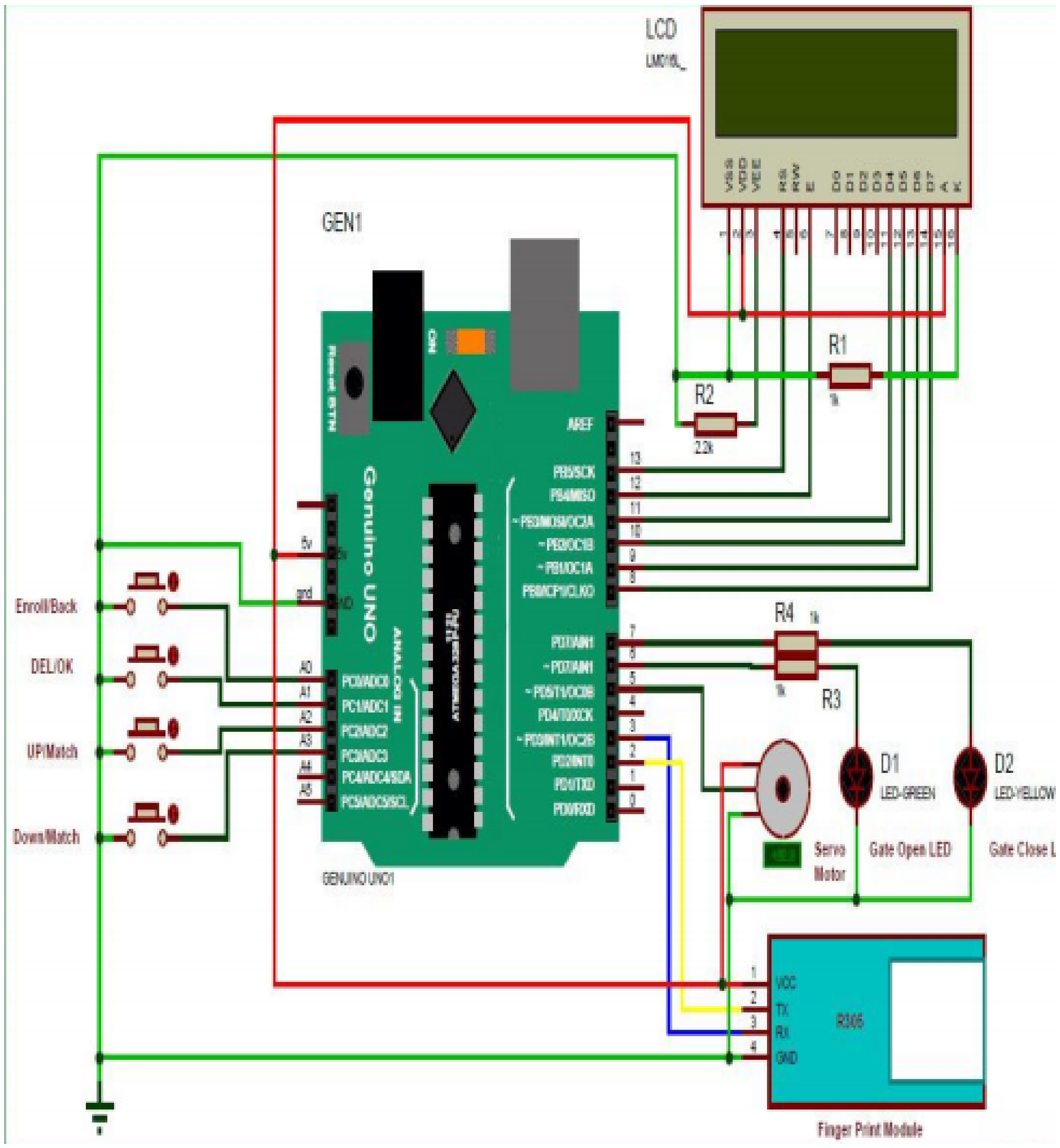
Electronic lock using fingerprint recognition system is a process of verifying the fingerprint image to open the electronic lock.Verification is completed by comparing the data of authorized fingerprint image with incoming fingerprint image. Then the information of incoming fingerprint image will undergo the comparison process to compare with authorized fingerprint image.

INTRODUCTION

(cntd.)

Fingerprint door lock incorporates the proven technology. Fingerprint reader scanning is the most mature and tested type of biometric technology. Recent studies on biometrics have shown that compared to the hand method, fingerprint is more accurate and cost-effective..

CIRCUIT DIAGRAM



COMPONENT REQUIRED

Arduino board

12v Solenoid

Finger print scanner R307

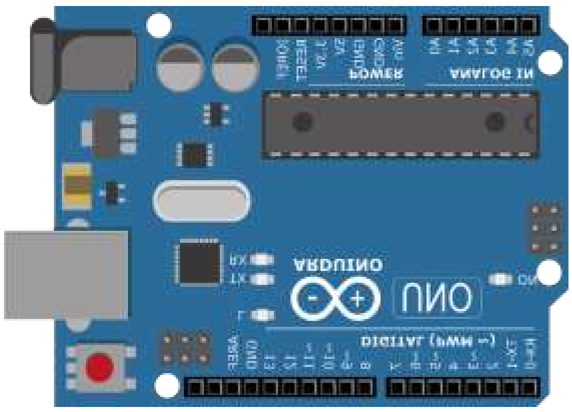
0-5 relay module

Bread board

Male to male and male to female jumped cables

ARDUINO UNO

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started



ARDUINO CODE

```
/* ENROLL */
/*
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); //you can change them if it is not working on 2 or 3

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

uint8_t id;

void setup()
{
Serial.begin(9600);
while (!Serial); // For Yun/Leo/Micro/Zero/...
delay(100);
Serial.println("\n\nFingerprint sensor enrollment");

// set the data rate for the sensor serial port
finger.begin(57600);

if (finger.verifyPassword()) {
Serial.println("Found fingerprint sensor!");
} else {
Serial.println("Did not find fingerprint sensor :(");
while (1) { delay(1); }
}
}

uint8_t readnumber(void) {
uint8_t num = 0;

while (num == 0) {
while (! Serial.available());
num = Serial.parseInt();
}
return num;
}

void loop() // program wil repeat this part (loop here)
{
Serial.println("Ready to enroll a fingerprint!");
Serial.println("Please type in the ID # (from 1 to 127) you want to save this finger
as...");
id = readnumber();
```

```
if (id == 0) {// ID #0 not allowed, try again!
return;
}
Serial.print("Enrolling ID #");
Serial.println(id);

while (! getFingerprintEnroll() );
}

uint8_t getFingerprintEnroll() {

int p = -1;
Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.println(".");
break;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
```

```
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
default:
Serial.println("Unknown error");
return p;
}

Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
Serial.print(".");
break;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
break;
default:
Serial.println("Unknown error");
break;
}
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
```

```
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
default:
Serial.println("Unknown error");
return p;
}

// OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
Serial.println("Fingerprints did not match");
return p;
} else {
Serial.println("Unknown error");
return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
Serial.println("Stored!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_BADLOCATION) {
Serial.println("Could not store in that location");
return p;
} else if (p == FINGERPRINT_FLASHERR) {
Serial.println("Error writing to flash");
return p;
} else {
Serial.println("Unknown error");
return p;
}
}
*/

/* Fingerprint*/
```

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3);

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup()
{
Serial.begin(9600);
while (!Serial); // For Yun/Leo/Micro/Zero/...
delay(100);
Serial.println("fingertest");
pinMode(12, OUTPUT);
pinMode(11, OUTPUT);

// set the data rate for the sensor serial port
finger.begin(57600);

if (finger.verifyPassword()) {
Serial.println("Found fingerprint sensor!");
} else {
Serial.println("Did not find fingerprint sensor :(");
while (1) {
delay(1);
}
}

finger.getTemplateCount();
Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println("
templates");
Serial.println("Waiting for valid finger...");
}

void loop() // run over and over again

{
getFingerprintIDez();
delay(50); //don't ned to run this at full speed.
digitalWrite(12, LOW);
digitalWrite(11, LOW);
}

uint8_t getFingerprintID() {
uint8_t p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image taken");
break;
case FINGERPRINT_NOFINGER:
```

```
Serial.println("No finger detected");
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Imaging error");
return p;
default:
Serial.println("Unknown error");
return p;
}

// OK success!

p = finger.image2Tz();
switch (p) {
case FINGERPRINT_OK:
Serial.println("Image converted");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Image too messy");
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Communication error");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Could not find fingerprint features");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Could not find fingerprint features");
return p;
default:
Serial.println("Unknown error");
return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println("Communication error");
return p;
} else if (p == FINGERPRINT_NOTFOUND) {
Serial.println("Did not find a match");
return p;
} else {
Serial.println("Unknown error");
return p;
```

```
}
{digitalWrite(11, HIGH);
delay(3000);
digitalWrite(11, LOW);
Serial.print("Not Found");
Serial.print("Error");
return finger.fingerID;
}
// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);
return finger.fingerID;
}
// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return -1;
p = finger.image2Tz();
if (p != FINGERPRINT_OK) return -1;
p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -1;
// found a match!
{
digitalWrite(12, HIGH);
delay(3000);
digitalWrite(12, LOW);
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);
} }
```

BIBLIOGRAPHY

https://electronicsworkshops.com/2020/09/12/fingerprint-door-based-lock-system-arduino/

https://www.youtube.com/watch?v=4iklgEtV9to

https://www.nedapsecurity.com/insight/biometric-security/

ACKNOWLEDGMENT

- We would like to thank our MCES professor, Mr. Niranjan C K for providing us with this learning opportunity and helping us get a comprehensible understanding on microcontrollers.

- Finally, yet importantly I would like to thank all open- source developers and programmers for sharing their useful work in the coding community and we as a group aim to help and contribute towards and for the betterment of the community as well.

CONCLUSION & PROJECT TAKE-AWAY

On completion of this project we have been able to understand the basics of microcontrollers and various sensors along with its application and working.

Growing IT culture in metropolitan cities have invoked an interest in the security soltions of the surrounding.

It is too important these daysto be updated with latest security solutions as the threat anf confidentiality of society has taken major folds and we need to make us and our family secure with the latest anti pick modular locks and newer security options.