Title: *Fingerprint Scanner*

I. Summary

A Fingerprint scanner is a system that scan and analyze the fingerprint of the user and verifies the person's identity. It is commonly used in security purposes and other stuff like employee verifications, identity verification for the government, and biometric authentication to approve transactions and grant the user access to some confidential information.

II. Objectives

i. To create an advance system that will allow the user to save and delete a fingerprint into the system.

ii. To provide students basic knowledge on how the fingerprint reader used in the industry.

III. Industry-Based Applications

A Fingerprint scanner is a type of electronic security system that uses fingerprint or biometrics as a key or a password to unlock gadgets, doors, confidential files, etc. Even though it was mostly seen in movies and TV shows but in real life it has been used for decades in a lot of industry. Every human being has a practically unique fingerprint that cannot be altered, which is why they are successfully used in identifying individuals. Most law enforcement agencies have a fingerprint database that they collect and maintain over the years. Many types of occupations such as financial advisor, teachers, security, contractors and other jobs that involve licensing and certification need fingerprinting as one of the requirements of employment.

The first smartphone that incorporate a fingerprint scanner was the Motorola Atrix (iii) that was in the year 2011. Since then, numerous smartphones have incorporated this kind technology feature such as Apple iphone 5s and 7, Apple iPad models. When it comes to PC security, there are a lot of fingerprint scanning options, some of them can be already integrated into certain laptop models (iv). Biometric door locks that uses fingerprint scanners in addition to touchscreen for manual entry are also available these days. Other theme parks, such as Walt Disney World, scan fingerprints upon entry to avoid ticket fraud.

IV. Project Methodology

a) Components

- **i.** Arduino UNO
- **ii.** Fingerprint module (R307) – used to scan fingerprints
- **iii.** Push buttons
- **iv.** 16x2 LCD – used to display messages
- **v.** Bread board or PCB
- **vi.** Wires – to connect the components
- **vii.** LED **–** used as indicator that the sensor is ready to take fingerprint for matching.
- **viii.** Two pieces 1k resistor and One piece 2.2k resistor
- **ix.** Buzzer
- **x.** Real Time Clock (RTC) module – A driver library that allows program to easily set, read or track the time and date.

b) Procedure

1.) Connect the push 4 push buttons to pin A0, A1, A2, and A3 of the Arduino respectively.

2.) Connect the LED is connected to the Digital pin D7 of the Arduino through a 1k resistor.

3.) Connect the Rx pin of the fingerprint module to the serial pin D2 of the Arduino.

4.) Connect the Tx pin of the fingerprint module to the serial pin D3 of the Arduino.

5.) Connect the Vcc of the Real Time Clock (RTC) and the Fingerprint module to the 5V pin of the Arduino Board.

6.) Connect the 16x2 LCD pins RS, EN, D4, D5, D6, and D7 directly to the Digital pins D13, D12, D11, D10, D9, and D8 of the Arduino respectively.

7.) Connect the buzzer to the pin A5 of the Arduino.

Note: If the student got stuck in the process please refer to the diagram below (figure 1).

8.) Use the Arduino Code below to program the fingerprint reader.

c) Codes and Diagrams

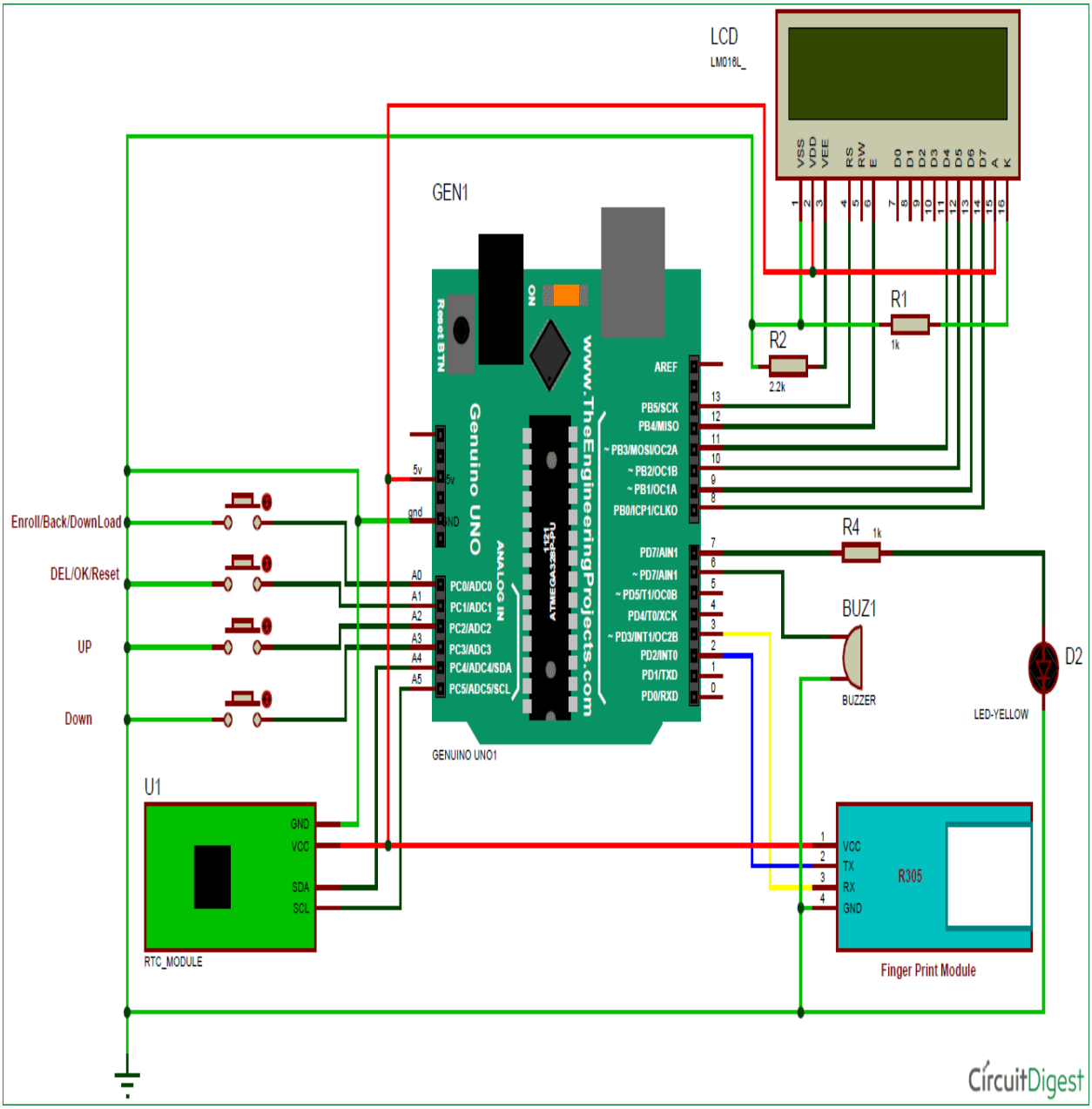

Figure 1.

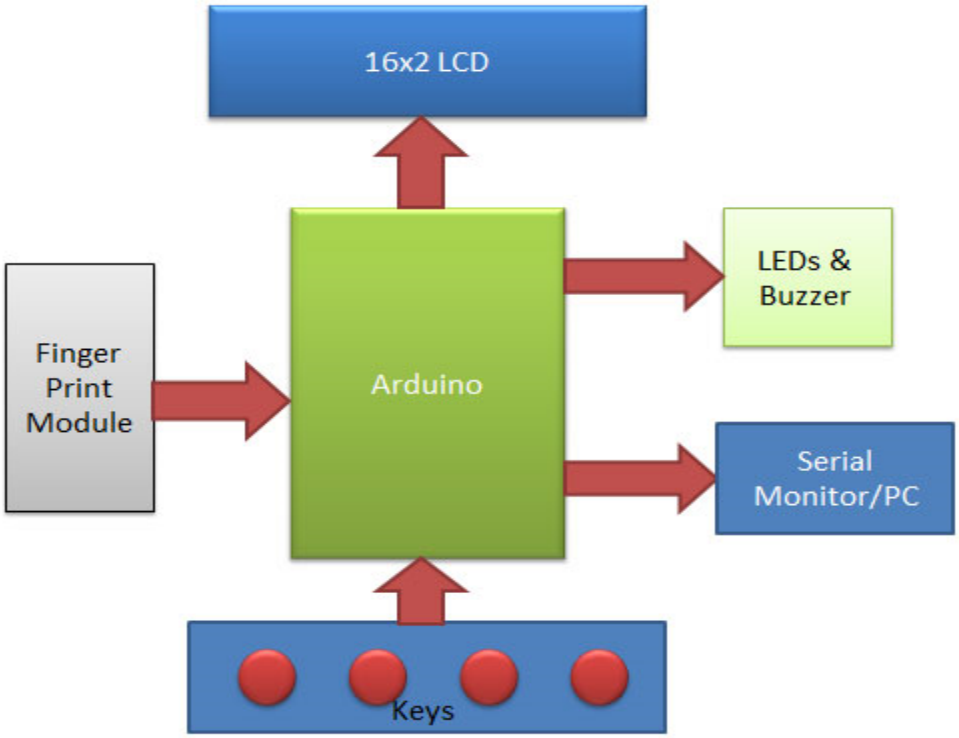

Figure 2.

V. References

i. https://circuitdigest.com/microcontroller-projects/fingerprint-attendance-system-using-arduino-uno

ii. https://youtu.be/IpBEbwMQYn8

iii. https://support.motorola.com/in/en/products/cell-phones/previous-android-models/atrix/documents/MS63688

iv. https://bestlaptopsworld.com/best-laptops-fingerprint-readers/

v. https://www.lifewire.com/understanding-finger-scanners-4150464

VI. Appendix

All data sheets should be stated here.

VII. Cost Analysis

Components	Website	Cost
Arduino UNO	https://store.arduino.cc/usa/arduino-uno-rev3	$22.00
Fingerprint module (r307)	https://www.adafruit.com/product/751	$49.95
Push Buttons	https://www.adafruit.com/product/4183	$2.50
16x2 LCD display	https://www.adafruit.com/product/399	$13.95
Breadboard wire bundle	https://www.adafruit.com/product/153	$4.95
Real Time Clock Module	https://www.adafruit.com/product/3295	$4.95
Buzzer	https://www.adafruit.com/product/1536	$0.95
Total		$99.25

Code

```
#include<EEPROM.h>

#include<LiquidCrystal.h>

LiquidCrystal lcd(13,12,11,10,9,8);

#include <SoftwareSerial.h>

SoftwareSerial fingerPrint(2, 3);


#include <Wire.h>

#include "RTClib.h"

RTC_DS1307 rtc;


#include "Adafruit_Fingerprint.h"

uint8_t id;

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&fingerPrint);


#define enroll 14

#define del 15

#define up 16

#define down 17

#define match 5

#define indFinger 7

#define buzzer 5
```

```
#define records 4  // 5 for 5 user

int user1,user2,user3,user4,user5;

DateTime now;

void setup()
{
   delay(1000);
   lcd.begin(16,2);
   Serial.begin(9600);
   pinMode(enroll, INPUT_PULLUP);
   pinMode(up, INPUT_PULLUP);
   pinMode(down, INPUT_PULLUP);
   pinMode(del, INPUT_PULLUP);
   pinMode(match, INPUT_PULLUP);
   pinMode(buzzer, OUTPUT);
   pinMode(indFinger, OUTPUT);
   digitalWrite(buzzer, LOW);
   if(digitalRead(enroll) == 0)
   {
     digitalWrite(buzzer, HIGH);
     delay(500);
```

```
digitalWrite(buzzer, LOW);
lcd.clear();
lcd.print("Please wait");
lcd.setCursor(0,1);
lcd.print("Downloding Data");

Serial.println("Please wait");
Serial.println("Downloding Data..");
Serial.println();

Serial.print("S.No.         ");
for(int i=0;i<records;i++)
{
      digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
  Serial.print("         User ID");
  Serial.print(i+1);
  Serial.print("                   ");
}
Serial.println();
int eepIndex=0;
for(int i=0;i<30;i++)
```

```
  {
    if(i+1<10)
    Serial.print('0');
    Serial.print(i+1);
    Serial.print("          ");
    eepIndex=(i*7);
    download(eepIndex);
    eepIndex=(i*7)+210;
    download(eepIndex);
    eepIndex=(i*7)+420;
    download(eepIndex);
    eepIndex=(i*7)+630;
    download(eepIndex);
  //  eepIndex=(i*7)+840;   // 5th user
  //  download(eepIndex);
    Serial.println();
  }
}
if(digitalRead(del) == 0)
{
  lcd.clear();
  lcd.print("Please Wait");
  lcd.setCursor(0,1);
```

```
  lcd.print("Reseting.....");
  for(int i=1000;i<1005;i++)
  EEPROM.write(i,0);
  for(int i=0;i<841;i++)
  EEPROM.write(i, 0xff);
  lcd.clear();
  lcd.print("System Reset");
  delay(1000);
}



lcd.clear();
lcd.print("   Attendance   ");
lcd.setCursor(0,1);
lcd.print("    System     ");
delay(2000);
lcd.clear();
lcd.print("Circuit Digest");
lcd.setCursor(0,1);
lcd.print("Saddam Khan");
delay(2000);
     digitalWrite(buzzer, HIGH);
  delay(500);
```

```
   digitalWrite(buzzer, LOW);
for(int i=1000;i<1000+records;i++)
{
  if(EEPROM.read(i) == 0xff)
      EEPROM.write(i,0);
 }

 finger.begin(57600);
 Serial.begin(9600);
 lcd.clear();
 lcd.print("Finding Module");
 lcd.setCursor(0,1);
 delay(1000);
 if (finger.verifyPassword())
 {
   Serial.println("Found fingerprint sensor!");
   lcd.clear();
   lcd.print("Found Module ");
   delay(1000);
 }
 else
 {
 Serial.println("Did not find fingerprint sensor :(");
```

```
    lcd.clear();
    lcd.print("module not Found");
    lcd.setCursor(0,1);
    lcd.print("Check Connections");
    while (1);
    }


     if (! rtc.begin())
       Serial.println("Couldn't find RTC");


    // rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));


    if (! rtc.isrunning())
    {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date & time this sketch was compiled
       rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    // This line sets the RTC with an explicit date & time, for example to set
    // January 21, 2014 at 3am you would call:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
    }
lcd.setCursor(0,0);
 lcd.print("Press Match to ");
```

```
 lcd.setCursor(0,1);
 lcd.print("Start System");
 delay(2000);

 user1=EEPROM.read(1000);
  user2=EEPROM.read(1001);
   user3=EEPROM.read(1002);
   user4=EEPROM.read(1003);
    user5=EEPROM.read(1004);
    lcd.clear();
    digitalWrite(indFinger, HIGH);

}

void loop()
{
    now = rtc.now();
    lcd.setCursor(0,0);
    lcd.print("Time->");
    lcd.print(now.hour(), DEC);
    lcd.print(':');
    lcd.print(now.minute(), DEC);
    lcd.print(':');
```

```
lcd.print(now.second(), DEC);
lcd.print("    ");
lcd.setCursor(0,1);
lcd.print("Date->");
lcd.print(now.day(), DEC);
lcd.print('/');
lcd.print(now.month(), DEC);
lcd.print('/');
lcd.print(now.year(), DEC);
lcd.print("     ");
delay(500);
int result=getFingerprintIDez();
if(result>0)
{
        digitalWrite(indFinger, LOW);
        digitalWrite(buzzer, HIGH);
        delay(100);
        digitalWrite(buzzer, LOW);
        lcd.clear();
        lcd.print("ID:");
        lcd.print(result);
        lcd.setCursor(0,1);
        lcd.print("Please Wait....");
```

```
            delay(1000);
            attendance(result);
            lcd.clear();
            lcd.print("Attendance ");
            lcd.setCursor(0,1);
            lcd.print("Registed");
            delay(1000);
        digitalWrite(indFinger, HIGH);
        return;
 }
 checkKeys();
 delay(300);
}

//     dmyyhms - 7 bytes
void attendance(int id)
{
  int user=0,eepLoc=0;
  if(id == 1)
  {
    eepLoc=0;
    user=user1++;
  }
```

```
else if(id == 2)
{
  eepLoc=210;
  user=user2++;
}
else if(id == 3)
{
  eepLoc=420;
  user=user3++;
}
else if(id == 4)
{
  eepLoc=630;
  user=user4++;
}
/*else if(id == 5)   // fifth user
{
  eepLoc=840;
  user=user5++;
}*/
else
return;
```

```
    int eepIndex=(user*7)+eepLoc;
    EEPROM.write(eepIndex++, now.hour());
    EEPROM.write(eepIndex++, now.minute());
    EEPROM.write(eepIndex++, now.second());
    EEPROM.write(eepIndex++, now.day());
    EEPROM.write(eepIndex++, now.month());
    EEPROM.write(eepIndex++, now.year()>>8 );
    EEPROM.write(eepIndex++, now.year());

    EEPROM.write(1000,user1);
    EEPROM.write(1001,user2);
    EEPROM.write(1002,user3);
    EEPROM.write(1003,user4);
  //  EEPROM.write(4,user5);   // figth user
}

void checkKeys()
{
   if(digitalRead(enroll) == 0)
   {
    lcd.clear();
    lcd.print("Please Wait");
    delay(1000);
```

```
    while(digitalRead(enroll) == 0);
    Enroll();
   }

   else if(digitalRead(del) == 0)
   {
    lcd.clear();
    lcd.print("Please Wait");
    delay(1000);
    delet();
   }
}

void Enroll()
{
   int count=1;
   lcd.clear();
   lcd.print("Enter Finger ID:");

   while(1)
   {
    lcd.setCursor(0,1);
     lcd.print(count);
```

```
if(digitalRead(up) == 0)
{
  count++;
  if(count>records)
  count=1;
  delay(500);
}

else if(digitalRead(down) == 0)
{
  count--;
  if(count<1)
  count=records;
  delay(500);
}
else if(digitalRead(del) == 0)
{
     id=count;
     getFingerprintEnroll();
     for(int i=0;i<records;i++)
     {
       if(EEPROM.read(i) != 0xff)
       {
```

```
            EEPROM.write(i, id);
            break;
          }
        }
        return;
    }

     else if(digitalRead(enroll) == 0)
    {
        return;
    }
 }
}

void delet()
{
  int count=1;
  lcd.clear();
  lcd.print("Enter Finger ID");

  while(1)
  {
   lcd.setCursor(0,1);
```

```
lcd.print(count);
if(digitalRead(up) == 0)
{
  count++;
  if(count>records)
  count=1;
  delay(500);
}

else if(digitalRead(down) == 0)
{
  count--;
  if(count<1)
  count=records;
  delay(500);
}
else if(digitalRead(del) == 0)
{
     id=count;
     deleteFingerprint(id);
     for(int i=0;i<records;i++)
     {
       if(EEPROM.read(i) == id)
```

```
          {
            EEPROM.write(i, 0xff);
            break;
          }
        }
        return;
    }

      else if(digitalRead(enroll) == 0)
    {
        return;
    }
 }
}

uint8_t getFingerprintEnroll()
{
  int p = -1;
  lcd.clear();
  lcd.print("finger ID:");
  lcd.print(id);
  lcd.setCursor(0,1);
  lcd.print("Place Finger");
```

```
delay(2000);
while (p != FINGERPRINT_OK)
{
  p = finger.getImage();
  switch (p)
  {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    lcd.clear();
    lcd.print("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.println("No Finger");
    lcd.clear();
    lcd.print("No Finger");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    lcd.clear();
    lcd.print("Comm Error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
```

```
    lcd.clear();
    lcd.print("Imaging Error");
    break;
  default:
    Serial.println("Unknown error");
     lcd.clear();
    lcd.print("Unknown Error");
    break;
  }
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    lcd.clear();
    lcd.print("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
     lcd.clear();
```

```
    lcd.print("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
        lcd.clear();
    lcd.print("Comm Error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
        lcd.clear();
    lcd.print("Feature Not Found");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
            lcd.clear();
    lcd.print("Feature Not Found");
    return p;
  default:
    Serial.println("Unknown error");
            lcd.clear();
    lcd.print("Unknown Error");
    return p;
}
```

```
Serial.println("Remove finger");
lcd.clear();
lcd.print("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");
 lcd.clear();
    lcd.print("Place Finger");
    lcd.setCursor(0,1);
    lcd.print("   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");
    return;
  }
}

// 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!");
  lcd.clear();
  lcd.print("Stored!");
  delay(2000);
} 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;
  }
}

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)
  {
```

```
  lcd.clear();
  lcd.print("Finger Not Found");
  lcd.setCursor(0,1);
  lcd.print("Try Later");
  delay(2000);
 return -1;
  }
 // found a match!
 Serial.print("Found ID #");
 Serial.print(finger.fingerID);
 return finger.fingerID;
}

uint8_t deleteFingerprint(uint8_t id)
{
 uint8_t p = -1;
 lcd.clear();
 lcd.print("Please wait");
 p = finger.deleteModel(id);
 if (p == FINGERPRINT_OK)
 {
   Serial.println("Deleted!");
   lcd.clear();
```

```
    lcd.print("Figer Deleted");
    lcd.setCursor(0,1);
    lcd.print("Successfully");
    delay(1000);
   }

   else
   {
    Serial.print("Something Wrong");
    lcd.clear();
    lcd.print("Something Wrong");
    lcd.setCursor(0,1);
    lcd.print("Try Again Later");
    delay(2000);
    return p;
   }
}

void download(int eepIndex)
{

        if(EEPROM.read(eepIndex) != 0xff)
        {
```

```
  Serial.print("T->");
  if(EEPROM.read(eepIndex)<10)
  Serial.print('0');
  Serial.print(EEPROM.read(eepIndex++));
  Serial.print(':');
  if(EEPROM.read(eepIndex)<10)
  Serial.print('0');
  Serial.print(EEPROM.read(eepIndex++));
  Serial.print(':');
  if(EEPROM.read(eepIndex)<10)
  Serial.print('0');
  Serial.print(EEPROM.read(eepIndex++));
  Serial.print("   D->");
  if(EEPROM.read(eepIndex)<10)
  Serial.print('0');
  Serial.print(EEPROM.read(eepIndex++));
  Serial.print('/');
  if(EEPROM.read(eepIndex)<10)
  Serial.print('0');
  Serial.print(EEPROM.read(eepIndex++));
  Serial.print('/');
  Serial.print(EEPROM.read(eepIndex++)<<8 | EEPROM.read(eepIndex++));
}
```

```
        else
        {
          Serial.print("---------------------------");
        }

        Serial.print("         ");
}
```