Project Report

FINGERPRINT BASED BIOMETRIC ATTENDANC SYSTEM USING ARDUINO

Done by:

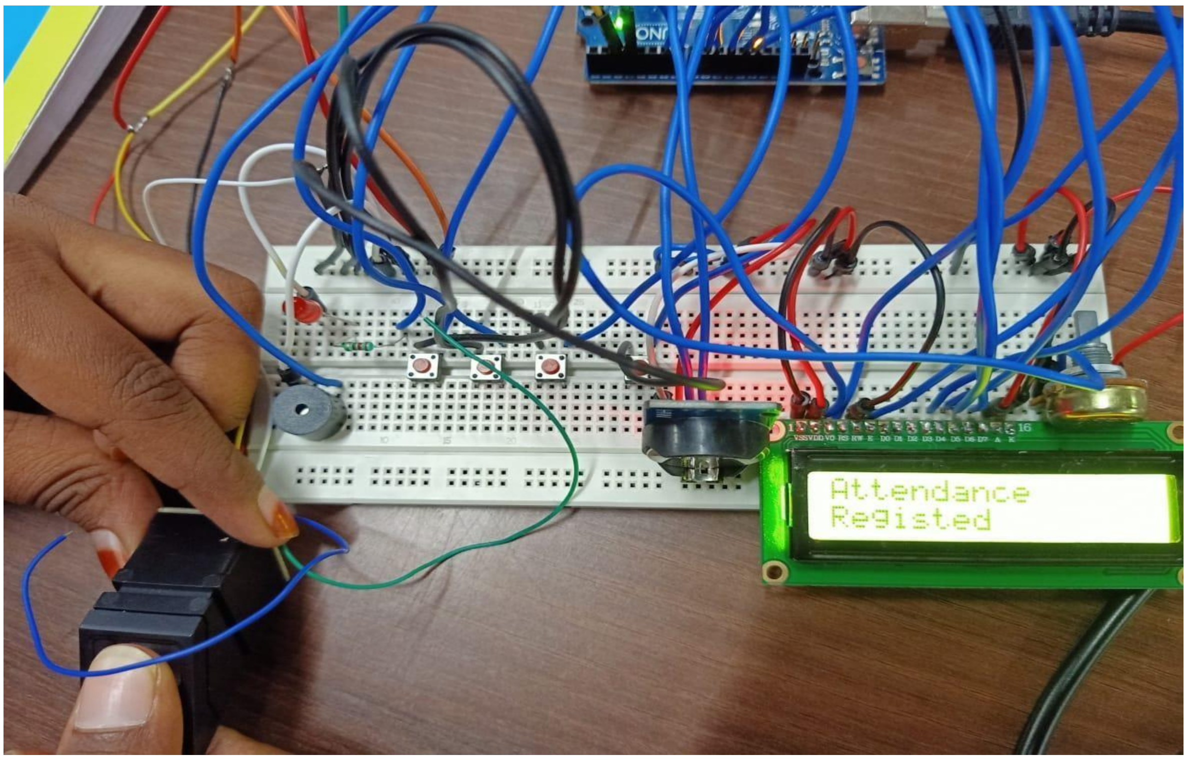
Kookutla Satwika

CONTENTS

- ABSTRACT
- INTRODUCTION
- BACKGROUND
- PROBLEM DEFINITION
- OBJECTIVE
- METHODOLOGY
- PROCEDURE
- THINGS CAN BE USED
- BASIC CIRCUIT DIAGRAM
- 3D DESIGN
- CONCLUSION
- REFERENCES
- CODING

ABSTRACT:

Attendance system is required in many different places such as offices, companies, schools, organisations and institutions etc. There are many attendance systems to take attendance. The main objective of this project is to study and construct the attendance system using fingerprint module. But every place we need to have a good system. This project describes one of the attendance systems.



INTRODUCTION:

In this project we are going to design Fingerprint Sensor Based Biometric Attendance System using Arduino. Simply we will be interfacing fingerprint sensor with Arduino, LCD Display & RTC Module to design the desired project. In this project, we used fingerprint Module and Arduino to take and keep attendance data and records.

Biometric Attendance systems are commonly used systems to mark the presence in offices and schools. This project has a wide application in school, college, business organization, offices where marking of attendance is required accurately with time. By using the fingerprint sensor, the system will become more secure for the users.

In developing countries most of the education institutions and government organisations still use paper based attendance method to keep and save the attendance. This causes time wastage during work time. Managing the attendance data is also very difficult when there is a large group. Therefore we use biometric attendance system for proper attendance management.

Biometric is the engineering technology used for automatic identification of a person based on biological characters such as fingerprint, iris, etc. The fingerprint verification system is commonly used biometric technique. Fingerprint based technique use computer to store and verify fingerprints.

BACKGROUND:

Fingerprints are the oldest form of biometric identification. Modem fingerprint based identification is used in forensic science, and in biometric systems such as civilian identification devices. Despite the widespread use of fingerprints, there is little statistical theory on the uniqueness of fingerprint minutiae.

PROBLEM DEFINITION:

There are some problems which face by fingerprint recognition or identification security. We can catch a cold by touching a biometric system .

- Twins have identical biometric traits .This is the same clones.
- Stolen body parts can be reused.
- Biometric features can be reconstructed from the template.
- Making a fake finger is easy.

OBJECTIVE:

The aim of this system is to implement in combination with these development techniques, statistical experiments can then be performed on the fingerprint data set.

The results from these experiments can be used to help us better understand what is involved in determining the statistical uniqueness of fingerprint minutiae.

The main aim that this system would test whether attendance by fingerprint is enough for identification. It is expected that the work in this system will reach the stage of being able to fully test hypothesis.

The system typically records details such as time of their arrival and departure apart from maintaining relevant information regarding the employees and their department, shifts, location, leaves status etc. The user-friendly devices and software eliminates paperwork.

METHODOLOGY:

The goal of this project is to develop a portable classroom attendance system based on fingerprint biometric. To achieve this objective, microcontroller board Arduino Mega is used as a central processor, fingerprint scanner to scan and retrieve biometric information from student. Real time clock (RTC) module to obtain the current time, date and day for the fingerprint reader and LCD to display the enrolment procedure.

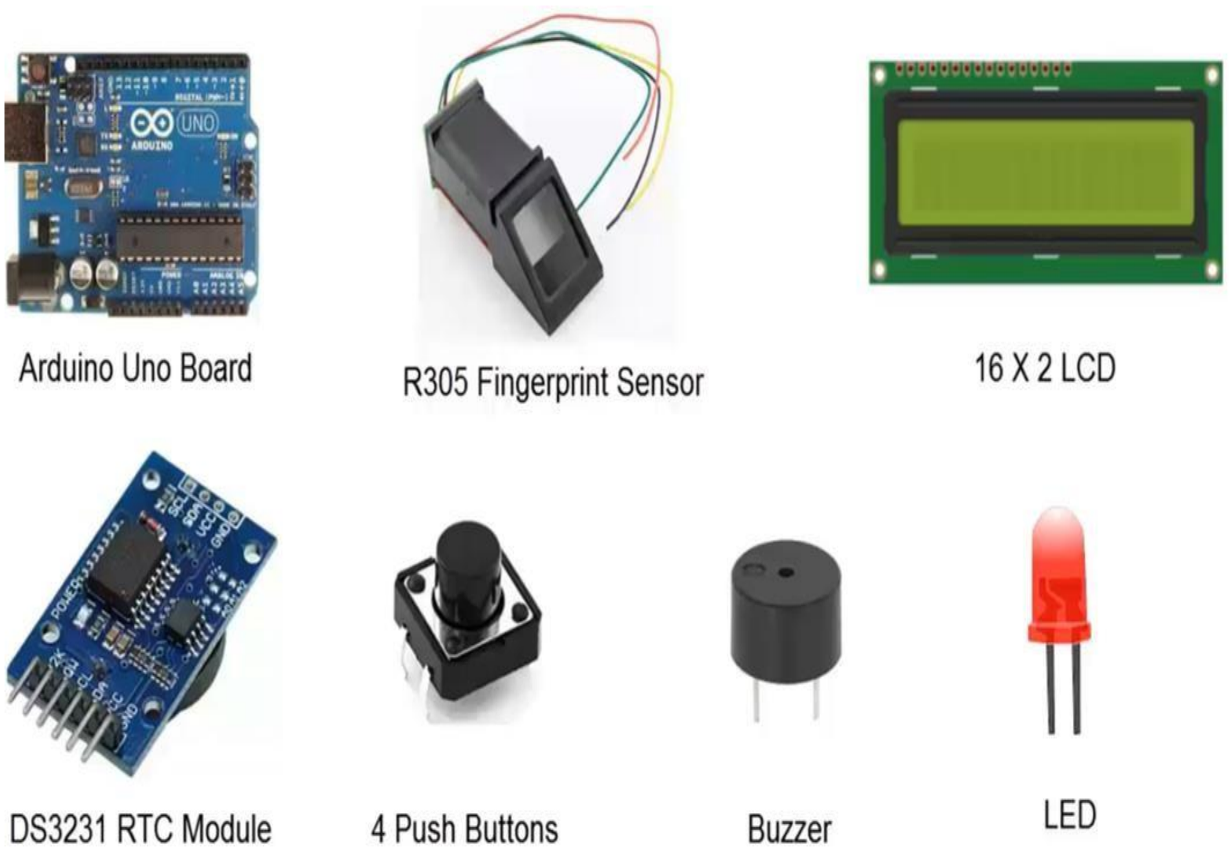
PROCEDURE :

Working of this fingerprint attendance system project is fairly simple. First of all, the user needs to enroll fingerprints of the user with the help of push buttons. To do this, user need to press ENROLL key and then LCD asks for entering ID for the fingerprint to save it in memory by ID name. So now user needs to enter ID by using UP/DOWN keys. After selecting ID, user needs to press OK key (DEL key). Now LCD will ask to place finger over the fingerprint module. Now user needs to place his finger over finger print module and then the module takes finger image. Now the LCD will say to remove finger from fingerprint module, and again ask to place finger again. Now user needs to put his finger again and module takes an image and convert it into templates and stores it by selected ID into the finger print module's memory. Now the user will be registered and he/she can feed attendance by putting their finger over fingerprint module. By the same method, all the users will be registered into the system.

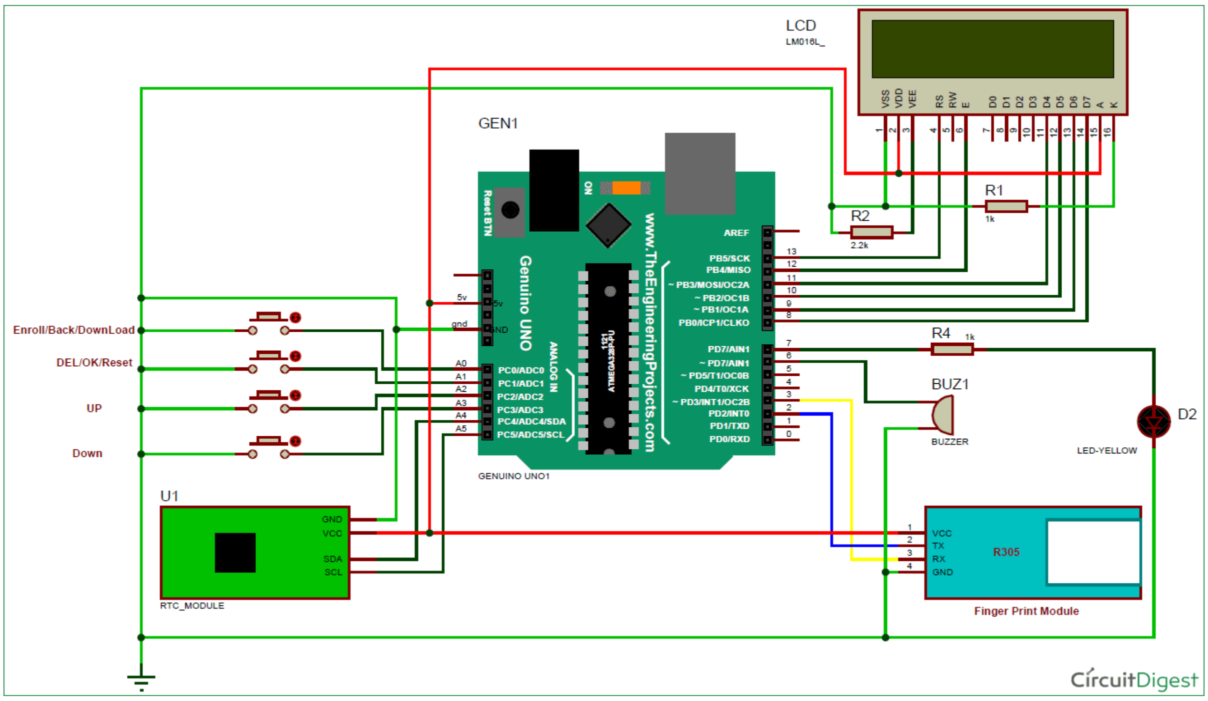
Now if the user wants to remove or delete any of the stored ID or fingerprint, then he/she need to press DEL key. Once delete key is pressed LCD will ask to select ID that need to be deleted. Now user needs to select ID and press OK key. Now LCD will let you know that fingerprint has been deleted successfully.

THINGS CAN BE USED:

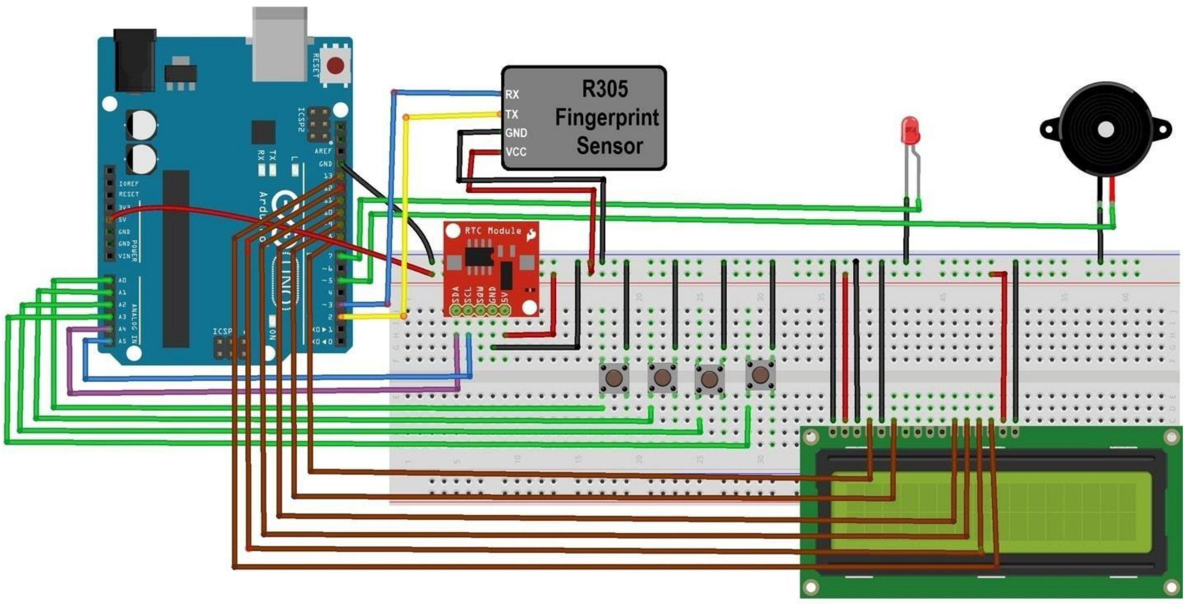
- Arduino -1
- Finger print module -1
- Push Button   - 4
- LEDs -1
- 1K Resistor -2
- 2.2K resistor -1
- Power
- Connecting wires
- Box
- 10.Buzzer-1
- 11.16x2 lcd-1
- 12.RTC Module -1
- 13.BreadBoard – 1



BASIC CIRCUIT DIAGRAM:



3D DESIGN :



CONCLUSION:

Now a days, Information systems and Communication Technologies (ICT) are becoming more and more improved. Biometric technology is also an effective tool to identify and detect fraudulent issues. A fingerprint-based attendance system is presented in this paper. This system will enhance the ability to detect the presence of the students in class or employees in an organization. In terms of efficiency and performance, fingerprint based attendance system is used in many places. This system is user-friendly and reliable because this system displays name, the ID numbers, date and time on excel sheet. This excel sheet can also be saved and attendance can be calculated with Microsoft Excel technique. Otherwise, this attendance system can be implemented valuable time of students and lectures, paper, generating report at required time.to check which person reached the work in time or on time or late time. So, this developed system is very also useful in saving

REFERENCES:

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

http://www.ijsrp.org/research-paper-0718/ijsrp-p7967.pdf

CODING:

```
#include "Adafruit_Fingerprint.h" //fingerprint library header file
#include<LiquidCrystal.h> //lcd header file
LiquidCrystal lcd(8,9,10,11,12,13);
#include <SoftwareSerial.h>
SoftwareSerial fingerPrint(2, 3); //for tx/rx communication between arduino & r305 fingerprint sensor

#include <Wire.h>
#include "RTClib.h" //library file for DS3231 RTC Module
RTC_DS3231 rtc;

uint8_t id;
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&fingerPrint);

#define register_back 14
#define delete_ok 15
#define forward 16
#define reverse 17
#define match 5
#define indFinger 7
#define buzzer 5

#define records 10 // 10 for 10 user

int user1,user2,user3,user4,user5,user6,user7,user8,user9,user10;

DateTime now;

void setup()
{
delay(1000);
lcd.begin(16,2);
Serial.begin(9600);
pinMode(register_back, INPUT_PULLUP);
pinMode(forward, INPUT_PULLUP);
pinMode(reverse, INPUT_PULLUP);
pinMode(delete_ok, INPUT_PULLUP);
pinMode(match, INPUT_PULLUP);
pinMode(buzzer, OUTPUT);
pinMode(indFinger, OUTPUT);
digitalWrite(buzzer, LOW);
```

```
if(digitalRead(register_back) == 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;
download(eepIndex);
eepIndex=(i*7)+1050;
download(eepIndex);
eepIndex=(i*7)+1260;
download(eepIndex);
eepIndex=(i*7)+1470;
download(eepIndex);
eepIndex=(i*7)+1680;
download(eepIndex);
Serial.println();
}
}
if(digitalRead(delete_ok) == 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(" Fingerprint ");
lcd.setCursor(0,1);
lcd.print("Attendance System");
delay(2000);
lcd.clear();

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(2000);
if (finger.verifyPassword())
{
Serial.println("Found fingerprint sensor!");
lcd.clear();
lcd.print(" Module Found");
delay(2000);
}
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.lostPower())
{
Serial.println("RTC is NOT running!");
// following line sets the RTC to the date & time this sketch was compiled
rtc.adjust(DateTime(2018, 6, 7, 11, 0, 0));
// This line sets the RTC with an explicit date & time, for example to set
// June 7, 2018 at 11am you would call:
// rtc.adjust(DateTime(2018, 6, 7, 11, 0, 0));
}
lcd.setCursor(0,0);
lcd.print(" Press Match to ");
lcd.setCursor(0,1);
lcd.print(" Start System");
delay(3000);

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("Registered");
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)
{
eepLoc=0;
user=user5++;
}
else if(id == 6)
{
eepLoc=840;
user=user5++;
}
else if(id == 7)
{
eepLoc=1050;
```

```
user=user7++;
}
else if(id == 8)
{
eepLoc=1260;
user=user8++;
}
else if(id == 9)
{
eepLoc=1470;
user=user9++;
}
else if(id == 10)
{
eepLoc=1680;
user=user8++;
}
/*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(register_back) == 0)
```

```
{
lcd.clear();
lcd.print("Please Wait");
delay(1000);
while(digitalRead(register_back) == 0);
Enroll();
}

else if(digitalRead(delete_ok) == 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(forward) == 0)
{
count++;
if(count>records)
count=1;
delay(500);
}

else if(digitalRead(reverse) == 0)
{
count--;
if(count<1)
count=records;
delay(500);
}
else if(digitalRead(delete_ok) == 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(register_back) == 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(forward) == 0)
{
count++;
if(count>records)
count=1;
delay(500);
}

else if(digitalRead(reverse) == 0)
{
count--;
if(count<1)
count=records;
delay(500);
```

```
}
else if(digitalRead(delete_ok) == 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(register_back) == 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 Found");
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(" Finger 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("Finger 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(" ");
}
```