**21RP01798**

**IPRC MUSANZE**

**ELECTRICAL AND ELECTRONIC ENGINEERING DEPARTEMENT**

**ELECTRICAL AUTOMATION TECHNOLOGY**

**MODULE TITLE: INDUSTRIAL SENSOR TECHNOLOGY**

**PROJECT NAME:**

**FINGER PRINT BASED BIOMETRIC ATTENDENSE SYSTEM**

**Abstract: The term 'biometrics' refers to a measurable characteristic that is unique to an individual such as fingerprints,**

**facial structure, the iris or a person's voice. This paper presents a fingerprint based biometric system that records the**

**attendance of a person by using a hand held fingerprint sensor. The experimental results suggest that many fraudulent**

**issues can be overcome using the fingerprint based attendance system and improves the reliability of the attendance**

**records.**

**Keywords: Biometric, Fingerprint, Attendance, Optical sensor, Arduino.**

**I. INTRODUCTION**

**attendance data and records. By using fingerprint sensor, the system will become more secure for the users. Following sections explains technical details of making a fingerprint based biometric attendance system using Arduino.Attendance is a concept that exists in different places like**

**institutions, organisations, hospitals, etc. during the start**

**and end of the day to mark a person’s presence. In early**

**days and even now in many places attendance is recorded**

**manually in attendance registers by calling out the names.**

**This results in waste of time and human effort. Also there**

**are many fraudulent issues that happen when we use a**

**register. For example, in educational institution, the**

**teacher calls out the names of the student’s one after the**

**other and marks their presence after they answer.**

**The other way that is followed is the teacher passes the**

**attendance sheet around the class for the students to sign**

**besides their names. But these methods have a major**

**drawback where the students tend to answer or sign for**

**their friends who are not present for that day. These**

**fraudulent issues may become more frequent if the class**

**strength is high.**

**II.SOLUTION TO OVERCOME THESE PROBLEMS**

**A solution to overcome these problems is by using a**

**system that will record the attendance automatically. In**

**this direction, this paper presents a fingerprint based**

**biometric system that records the attendance**

**automatically.** **This system consists of a fingerprint sensor**

**which is used to detect the person’s identification. For**

**example, in educational institutions, the student needs to**

**place their finger on the fingerprint sensor to obtain their**

**attendance. The fingerprint captured is recorded in a flash**

**memory and then each time it is checked whether the**

**obtained fingerprint matches with the record in the flash**

**memory after which the student gets the attendance. By**

**making use of this system, we overcome the issues such as**

**proxy so no student can give attendance for their friends**

**who are absent.**

**APPLICATION OF THIS PROJECT**

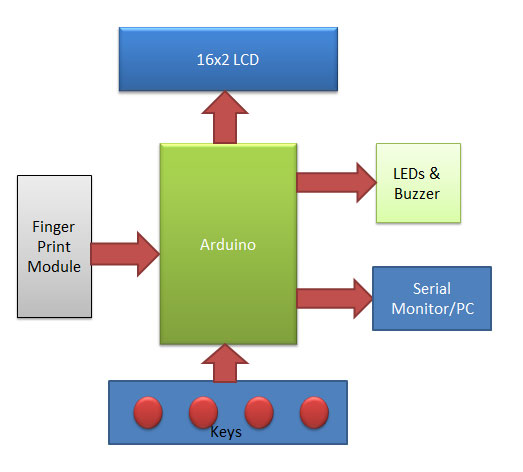
**Attendance systems are commonly used systems to mark the presence in offices and schools. From manually marking the attendance in attendance registers to using high-tech applications and biometric systems, these systems have improved significantly. In our previous projects, we have covered few other electronic attendance system projects using RFID and AVR microcontroller, 8051 and raspberry Pi. In this project, we used fingerprint Module and Arduino to take and keep**

**III.Required Components**

* **Arduino -1**
* **Finger print module -1**
* **Push Button - 4**
* **LEDs -1**
* **1K Resistor -2**
* **2.2K resistor -1**
* **Power**
* **Connecting wires**
* **Box**
* **Buzzer -1**
* **16x2 LCD -1**
* **Bread Board -1**
* **RTC Module -1**

**IV.Project Description:**

**want to proceed further then he/she can press ENROLL key again to go back. This time ENROLL key behave as Back key, i.e. ENROLL key has both enrollment and back function. Besides enroll key is also used to download attendance data over serial monitor. Similarly, DEL/OK key also has the same double function like when user enrolls new finger, then he/she need to select finger ID by using another two key namely UP and DOWN. Now user need to press DEL/OK key (this time this key behave like OK) to procIn this fingerprint attendance system circuit, we used Fingerprint Sensor module to authenticate a true person or employee by taking their finger input in the system. Here we are using 4 push buttons to enroll, Delete, UP/Down. ENROLL and DEL key has triple features. ENROLL key is used for enrollment of a new person into the system. So when the user wants to enroll new finger then he/she need to press ENROLL key then LCD asks for the ID, where user want to be store the finger print image. Now if at this time user does not eed with selected ID. And Del key is used for reset or delete data from EEPROM of Arduino**.



Block diagram for biometric attedance system project using arduino

**FingerPrint module**:

Fingerprint sensor module



**Fingerprint sensor module captures finger’s print image and then converts it into the equivalent template and saves them into its memory as per selected ID by Arduino. All the process is commanded by Arduino like taking an image of finger’s print, convert it into templates and storing as ID etc. You can check some more projects using fingerprint module:**

**Here we have added a Yellow LED which indicates that fingerprint module is ready to take an image of the finger. A buzzer is also used for various indications. Arduino is the main component of this system it is responsible for control of the whole system.**

**V.Working of Fingerprint Based Attendance System**

**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 (same DEL key). Now LCD will let you know that fingerprint has been deleted successfully.**

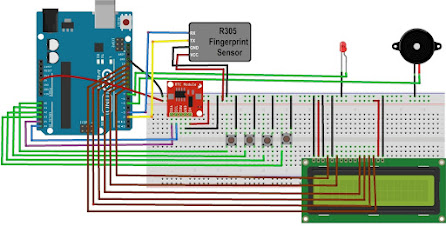
**VI.How Attendance works in this Fingerprint Attendance System Project:**

**Whenever user place his finger over fingerprint module then fingerprint module captures finger image, and search if any ID is associated with this fingerprint in the system. If fingerprint ID is detected then LCD will show Attendance registered and in the same time buzzer will beep once and LED will turn off until the system is ready to take input again.**

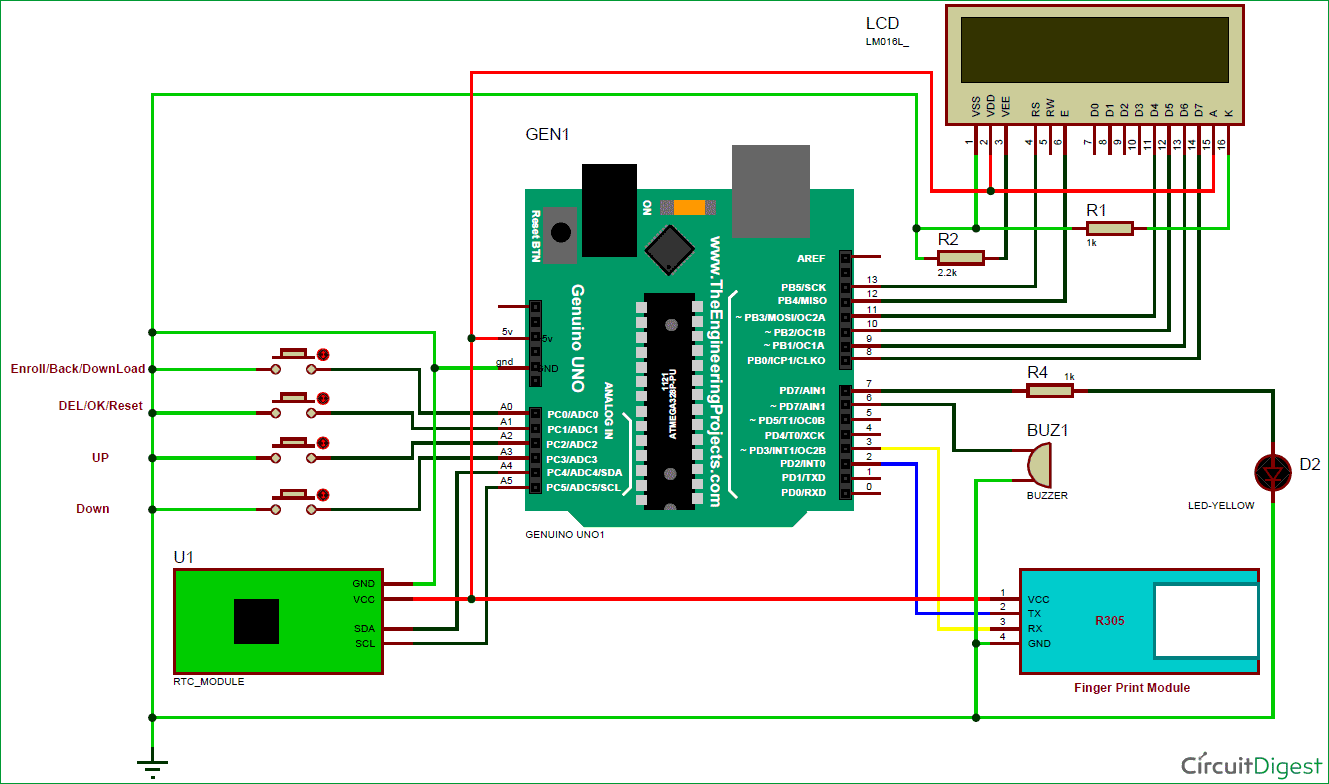
**Along with the fingerprint module, we have also used an RTC module for Time and date. Time and date are running continuously in the system. So Arduino take time and date whenever a true user places his finger over fingerprint and save them in the EEPROM at the allotted slot of memory.**

**Here we have created 5 user space in this system for 30 days. By pressing the RESET button in Arduino and then immediately enroll key will be responsible for downloading attendance data over serial monitor from the Arduino EEPROM Memory .**

**VII.Circuit Diagram and Description for Fingerprint Attendance System Project**

****

**The circuit of this fingerprint based attendance system project, as shown in the above diagram is quite simple. It has Arduino for controlling all the process of the project, push button for enrolling, deleting, selecting IDs and for attendance, a buzzer for alerting, LEDs for indication and LCD to instruct user and showing the resultant messages.**

****

**finger print based attendance system using arduino circuit**

**As shown in the circuit diagram, a push button is directly connected to pin A0(ENROL), A1(DEL), A2(UP), A3(DOWN) of Arduino with respect to the ground And Yellow LED is connected at Digital pin D7 of Arduino with respect to ground through a 1k resistor. Fingerprint module’s Rx and Tx directly connected at Serial pin D2 and D3 (Software Serial) of Arduino. 5v supply is used for powering finger print module taken from Arduino board. A buzzer is also connected at pin A5. A 16x2 LCD is configured in 4-bit mode and its RS, EN, D4, D5, D6, and D7 are directly connected at Digital pin D13, D12, D11, D10,D9, and D8 of Arduino.**

**Code Explanation:**

**The fingerprint attendance system code for arduino is given in the subsequent sections. Although the code is explained well with comments, we are discussing here few important parts of the code. We used fingerprint library for interfacing finger print module with Arduino board.**

**First of all, we include the header file and defines input and output pin and define the macro and declared variables. After this, in setup function, we give direction to defined pin and initiate LCD and finger print module**

**After it, we have to write code for downloading attendance data.**

**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");**

**Afterit, we have to write code for clearing attendance data from EEPROM.**

**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);**

**}**

**After it, we initiate finger print module, showing welcome message over LCD and also initeiated RTC module.**

**After it, in loop function, we have read RTC time and displayed it on LCD**

**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);**

**After it, waiting for the finger print to take input and compare captured image ID with stored IDs. If amatch occurs then proceed with next step. And checking enroll del keys as well**

**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;**

**}**

**Given void checkKeys() function is used for checking Enroll or DEL key is pressed or not and what to do if pressed. If the ENROL key pressed the Enroll() function is called and DEL key press then delete() function is called.**

**void delet() function is used for entering ID to be deleted and calling uint8\_t deleteFingerprint(uint8\_t id) function that will delete finger from records.**

**Given Function is used to taking finger print image and convert them into the template and save as well by selected ID into the finger print module memory.**

**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();**

**..... .....**

**....... ....**

**Given function is used for storing attendance time and date in the allotted slot of EEPROM**

**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)**

**.... ....**

**.....**

**Given function is used to fetching data from EEPROM and send to serial monitor**

**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++));**

**.... ....**

**.....**

**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(" ");**

**}**