PROJECT REPORT ON:

"Fingerprint Based Biometric Attendance System using Arduino"

Submitted by:

ADITHYA SHEELVANTH (3GN18EC001)
AMEETH PARSHETTY (3GN18EC006)

Submitted on: 25/10/2021

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

GURU NANAK DEV ENGINEERING COLLEGE BIDAR,

KARNATAKA-585403

UNDERTAKING

I declare that the work presented in this project titled "**Fingerprint Based Biometric Attendance System using Arduino**", submitted to the All India council of robotics and Automation, for the award of the Internship in **INTERNET OF THINGS**, is my original work. I have not plagiarized or submitted the same work for the award of any other Internship. In case this undertaking is found incorrect, I accept that my Project may be unconditionally withdrawn.

ADITHYA SHEELVANT
AMEETH PARSHETTY

CERTIFICATE

Certified that the work contained in the project titled "**Fingerprint Based Biometric Attendance System using Arduino**", by **ADITHYA SHEELVANT** and **AMEETH PARSHETTY** has been carried out under my supervision and that this work has not been submitted elsewhere for an Internship.

All India Council of Robotics and Automation

Name of the Internship

Delhi-110020

ACKNOWLEDGEMENT

I take upon this opportunity to acknowledge the many people whose prayers and support meant a lot to me

I am deeply indebted to my mentor **SUMIT CHATTERJEE RESEARCH ENGINEER**, **ALAAUDEEN K M** who motivated me along the way

I would like to thank all my teachers who help me in this project

I further thank to my friends

I owe my sincere gratitude towards the God

My heartfelt thanks to parents who supported me a lot

Finally, I would like to wind up by paying my heartfelt thanks to AICRA institute who provided me with this great opportunity

ADITHYA SHEELVANT
AMEETH PARSHETTY

TABLE OF CONTENTS

INTRODUCTION

In this project, we are going to design a 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 the 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.

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 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**.

R305 FINGERPRINT SCANNER SENSOR MODULE

Introduction:

This is a fingerprint sensor module with TTL UART interface for direct connections to microcontroller UART or to PC through MAX232 / USB-Serial adapter. The user can store the fingerprint data in the module and can configure it in 1:1 or 1: N mode for identifying the person.

The Fingerprint module can be directly interfaced with any microcontroller as well as Arduino Board. This optical biometric fingerprint reader with great features and can be embedded into a variety of end products like access control system, attendance system, safety deposit box, car door locking system.

Features:

1. Integrated image collecting and algorithm chip together, All-in-one
2. Fingerprint can conduct secondary development & embedded into a variety of end products
3. Low power consumption, low cost, small size, excellent performance
4. Professional optical technology, precise module manufacturing techniques
5. Good image processing capabilities can successfully capture image up to resolution 500 dpi

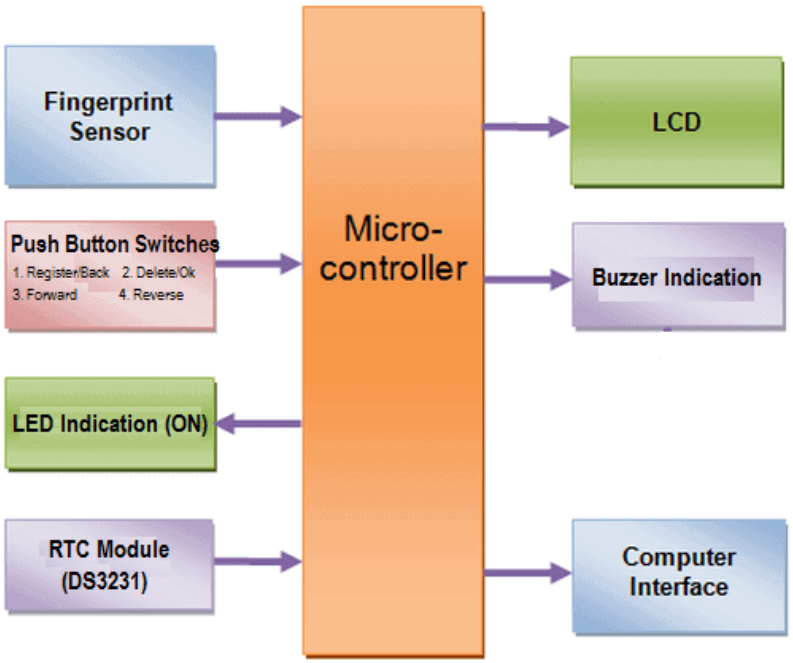
Specifications:

1. Fingerprint sensor type: Optical
2. Sensor Life: 100 million times
3. Static indicators: 15KVBacklight: bright green
4. Interface: USB1.1/UART (TTL logical level)
5. RS232 communication baud rate: 4800BPS~115200BPS changeable
6. Dimension: 553221.5mm
7. Image Capture Surface 15—18(mm)
8. Verification Speed: 0.3 sec
9. Scanning Speed: 0.5 sec
10. Character file size: 256 bytes
11. Template size: 512 bytes
12. Storage capacity: 250
13. Security level: 5 (1,2,3,4,5(highest))
14. False Acceptance Rate (FAR) :0.0001%
15. False Rejection Rate (FRR): 0.1%
16. Resolution 500 DPI
17. Voltage :3.6-6.0 VDC
18. Working current: Typical 90 mA, Peak 150mA
19. Matching Method: 1: N
20. Operating Environment Temperature: -20 to 45° centigrade

HARDWARE DESCRIPTION

SL No.	COMPONENTS	DESCRIPTION	QUANTITY
01	Arduino Board	Arduino UNO R3 Development Board	**1**
02	Fingerprint Sensor	R305/R307 Fingerprint Sensor Module	**1**
03	RTC Module	DS3231/DS1307 Real Time Clock Module	**1**
04	LCD Display	JHD162A 16x2 LCD Display	**1**
05	Potentiometer	10K	**1**
06	Push Buttons	Push-To-ON Reset Tact Switch	**5**
07	Buzzer	5V Active Buzzer	**1**
08	LED	5mm LED Any Colour	**1**
09	Connecting Wires	Jumper Wires	**20**
10	Breadboard		**1**

BLOCK DIAGRAM

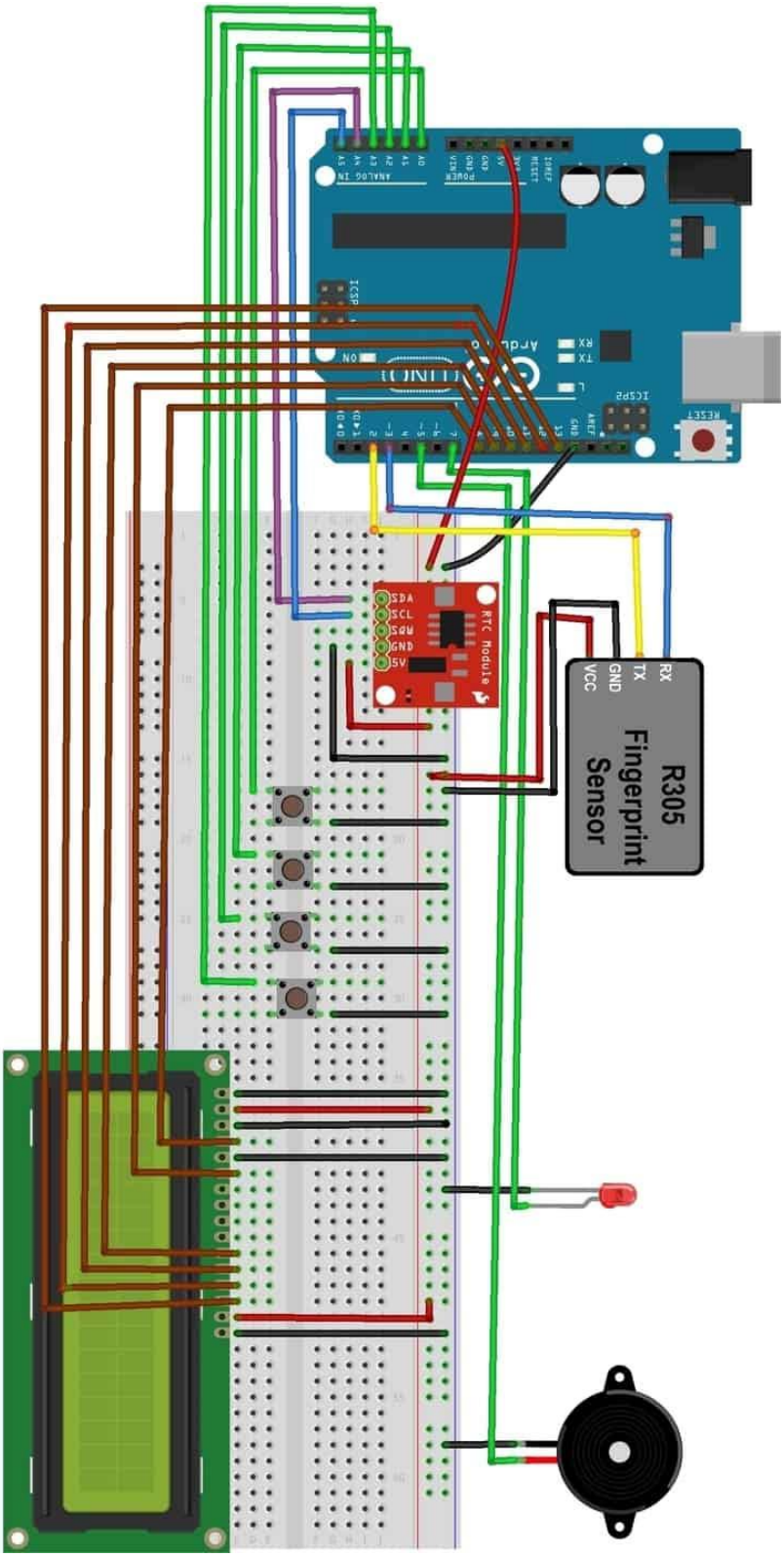


In this Fingerprint Sensor Based Biometric Attendance System using Arduino, we used a Fingerprint Sensor module to authenticate a true person or employee by taking their finger input in the system.

Here we are using 4 push buttons to register new fingerprint or delete stored fingerprint or match stored fingerprint. The 4 push buttons are used as an input unit for these tasks. Similarly, RTC Module DS3231 is used for registering scanning/entering/existing time of the user.

The LCD displays the time record and every function happening via push button. Buzzer indicates different functions and happening whenever an interrupt is detected. The LED is used for power indication.

CIRCUIT DIAGRAM



SOURCE CODE PROGRAM

The source code/Program for Fingerprint Sensor Based Biometric Attendance System using Arduino is

given below.

```
#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("         ");
}
```

WORKING EXPLANATION

The working of the Fingerprint Sensor Based Biometric Attendance System. In this project, we have used a DS3231 RTC Module for time & date display. We used 1 LED for power indication, 1 buzzer for different function indication. We have interfaced 16*2 LCD which displays everything whenever the finger is placed or removed, or registering attendance or downloading data.

We have used 4 push buttons which are used to control the entire system. The functions of each button are:

1. Register/Back Button – Used for enrolling new fingerprint as well as reversing the back process or going back

2. Delete/OK Button – This Button is used for deleting the earlier stored fingerprint system as well as granting access as an OK selection.

3. Forward Button – Used for moving forward while selecting the memory location for storing or deleting fingerprints.

4. Reverse Button – Used for moving backward while selecting memory location for storing or deleting fingerprints.

Enrolling New Fingerprint

To enrol New Fingerprint Click on the Enrol button. Then select the memory location where you want to store your fingerprint using the UP/DOWN button. Then click on OK. Put your finger and remove your finger as the LCD instructs. Put your finger again. So finally your fingerprint gets stored.

Deleting Stored Fingerprint

To delete the fingerprint which is already clicked on DEL Button. Then select the memory location where your fingerprint was stored earlier using the UP/DOWN button. Then click on OK. So finally your fingerprint is deleted.

Downloading Data:

Simply click on Register/Back Button and reset the button together. At this movement, the serial monitor should be opened.

ADVANTAGES OF FINGERPRINT RECOGNITION

- **Security –** security-wise, it is a vast improvement on passwords and identity cards. Fingerprints are much harder to fake, they also change very little over a lifetime, so the data remains current for much longer than photos and passwords.

- **Ease of use** – for the user they are simple and easy to use. No more struggling to remember your last password or being locked out due to leaving your photo ID at home. Your fingerprints are always with you.

- **Non-transferable** – fingerprints are non-transferrable, ruling out the sharing of passwords or 'clocking in' on behalf of another colleague. This allows for more accurate tracking of workforce and provides additional security against the theft of sensitive materials.

- **Accountability** – using fingerprint recognition also provides a higher level of accountability at work. Biometric proof you have been present when a situation or incident has occurred is hard to refute and can be used as evidence if required.

- **Cost effective** – from a technology management perspective, fingerprint recognition is now a cost-effective security solution. Small hand-held scanners are easy to set up and benefit from a high level of accuracy.
-

DISADVANTAGES OF FINGERPRINT RECOGNITION

- **System failures** – scanners are subject to the same technical failures and limitations as all other electronic identification systems such as power outages, errors and environmental factors.

- **Cost** – it is true that fingerprint recognition systems are more cost effective than ever, but for smaller organisations the cost of implementation and maintenance can still be a barrier to implementation. This disadvantage is lessening as devices become more cost effective and affordable.

- **Exclusions** – while fingerprints remain relatively stable over a person's lifetime there are sections of the population that will be excluded from using the system. For example, older people with a history of manual work may struggle to register worn prints into a system or people who have suffered the loss of fingers or hands would be excluded.

CONCLUSION

Here we have developed a Biometric fingerprint based attendance system using Arduino. In this project we have used R307 fingerprint sensor which reads the Fingerprint and stores in the form of digital data. A buzzer is activated and LED blinks then LCD panel shows that data is stored along with username, date and time. 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. The system can be improved by encasing it in a plastic covering. This would make it more compact and easy to use in a classroom setting. The system can be configured to enable lecture wise attendance taking. It can further be improved to automatically calculate attendance percentages of students and intimate the teachers if a student's attendance is below a certain percentage. It can also be modified to fit the corporate environment.

The traditional process of manually taking and maintaining student attendance is highly inefficient and time consuming. The attendance monitoring system based on biometric authentication has a potential to streamline the whole process. An Internet of Things (IoT) based portable biometric attendance system can prove to be of great value to educational institutions in this regard as it proves to be highly efficient and secure. The cost involved in making this system is quite less, when compared to conventional biometric attendance system.

The use of cloud computing to store the attendance records makes all the data easy to access and retrieve as end when required by the teachers. The use of fingerprint scanner ensures the reliability of the attendance record. The system, due to its lack of complexity, proves to be easy to use and user friendly

REFERENCES

- https://ijarcce.com/wp-content/uploads/2016/11/IJARCCE-ICRITCSA-3.pdf
- https://www.skyfilabs.com/project-ideas/biometric-attendance-system-with-iot
- https://www.how2electronics.com/fingerprint-sensor-based-biometric-attendance-system/
- https://circuitdigest.com/microcontroller-projects/fingerprint-attendance-system-usingarduino-uno
- Wang and Jingli , "The Design of Teaching Management System in Universities Based on Biometrics Identification and the Internet of Things Technology", IEEE 10th International Conference on Computer Science & Education (ICCSE), Cambridge University, UK July 22-24, 2015, pp. 979-982.