



# DIGITAL ELECTRONICS- INNOVATIVE PROJECT

(FINGERPRINT BASED BIOMETRIC ATTENDANCE SYSTEM USING ARDUINO)



# HELLO!

We are CSE students in DTU namely  
**Ayush Karn (2K19/Co/454) &&**  
**Sonu Kumar Kushwaha (2K19/CO/383)**  
presenting our Innovative Project Work.

Submitted To:  
**Mrs. Lavi Tanwar**





## ABSTRACT

Attendance systems are commonly used systems to mark the presence in offices and schools. From manually marking the attendance in attendance registers is not so efficient in compare to high-tech applications and biometric systems, these systems have improved significantly in past years by introduction of biometric security systems in small mobile equipment like smartphones. 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 as well as efficient for the database management.

### Concepts of Digital Electronics that are used are:

- 1) Image of the Finger is converted into Computer understandable binary using some Algorithm that we we'll be explaining in Final Review.
- 2) **EEPROM** of Arduino is used to store the data scanned by fingerprint module.
- 3) Comparison of fingerprints is also done using some Algorithm which is also a part of Digital Electronics
- 4) Arduino is used which itself is a part of Digital Electronic



# APPROACH



Considering the Pandemic Situation from very beginning of the 2<sup>nd</sup> year of our engineering, we are not being able to use any Laboratory hardware. So, we planned to go for Hardware implementation in this project. Even simple Bread-Board was new for us to use in our project. We firstly learned about the about the working Mechanism of the components, and then tried to make the complete circuit. Even the circuit Diagram was available with us, but Hardware implementation was quite a different experience than the Simulation. Final experience is shared in Conclusion.



## WHAT IS ARDUINO?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.



## WHY ARDUINO?

It's simple and accessible user experience, Arduino has been used in thousands of different projects and applications. The Arduino software is easy-to-use for beginners, yet flexible enough for advanced users. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

There are many other microcontrollers and microcontroller platforms available for physical computing. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, and many others offer similar functionality. All of these tools take the messy details of microcontroller programming and wrap it up in an easy-to-use package. Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

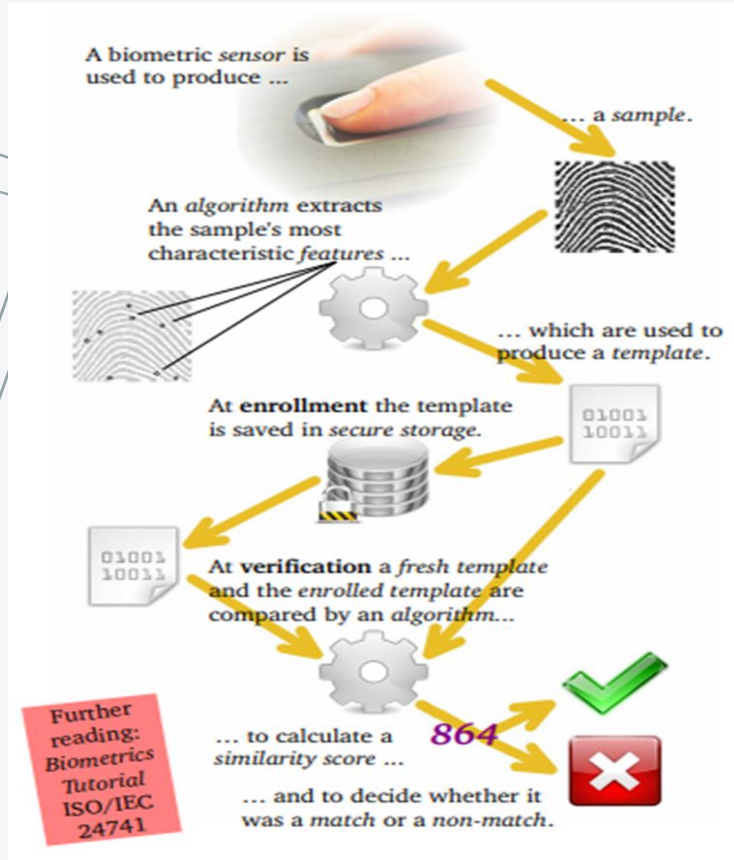
- Inexpensive
- Cross-platform
- Simple, clear programming environment
- Open source and extensible software
- Open source and extensible hardware

## FINGERPRINT MODULE



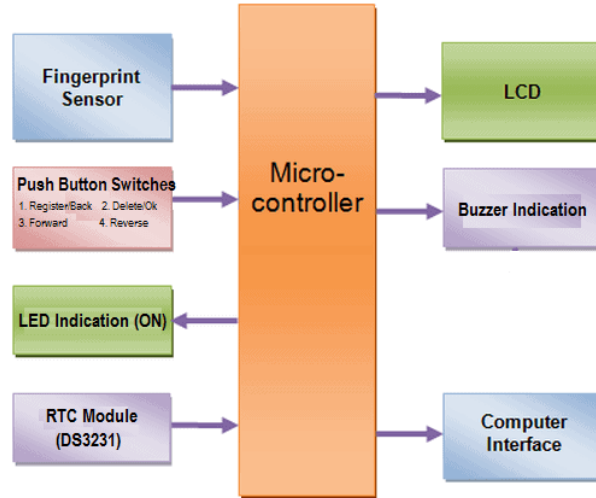
Fingerprint sensor module captures finger's print image and then converts it into the equivalent template and saves them into its memory as per selected ID by Arduino. All the process is commanded by Arduino like taking an image of finger's print, convert it into templates and storing as ID etc.

# FINGER'S PRINT IMAGE TO THE EQUIVALENT TEMPLATE



1. A biometric sensor is used to produce a sample image of fingerprint.
2. An algorithm extracts the most characteristic feature of the sample image.
3. These characteristic features are used to produce templates.
4. These templates are saved in secure place, of EEPROM of the Arduino.
5. At verification, a fresh template and the enrolled templates are compared by an algorithm.
6. The algorithm is used to calculate the similarity score and to decide whether it was a match or a non-match.

# BLOCK-DIAGRAM

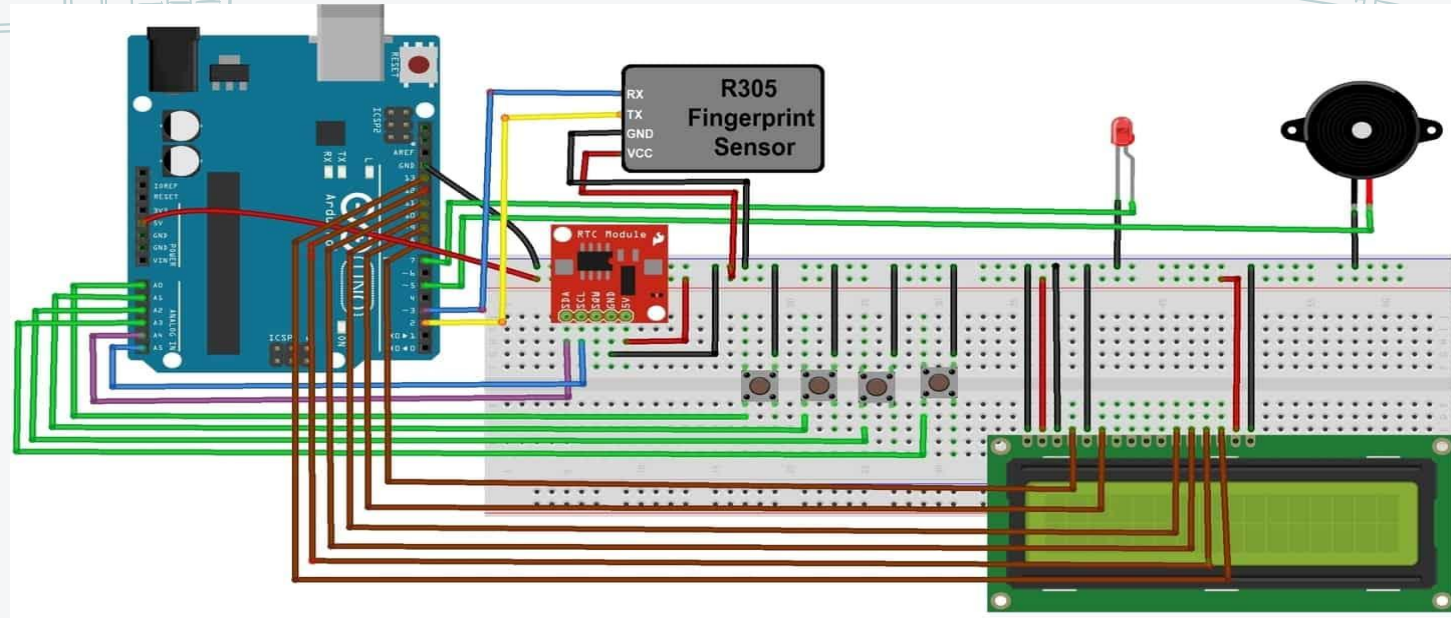


In this Fingerprint Sensor Based Biometric Attendance System using Arduino, we used a Fingerprint Sensor module to authenticate a true person or employee by taking their finger input in the system. Here we are using 4 push buttons to register new fingerprint or delete stored fingerprint or match stored 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.



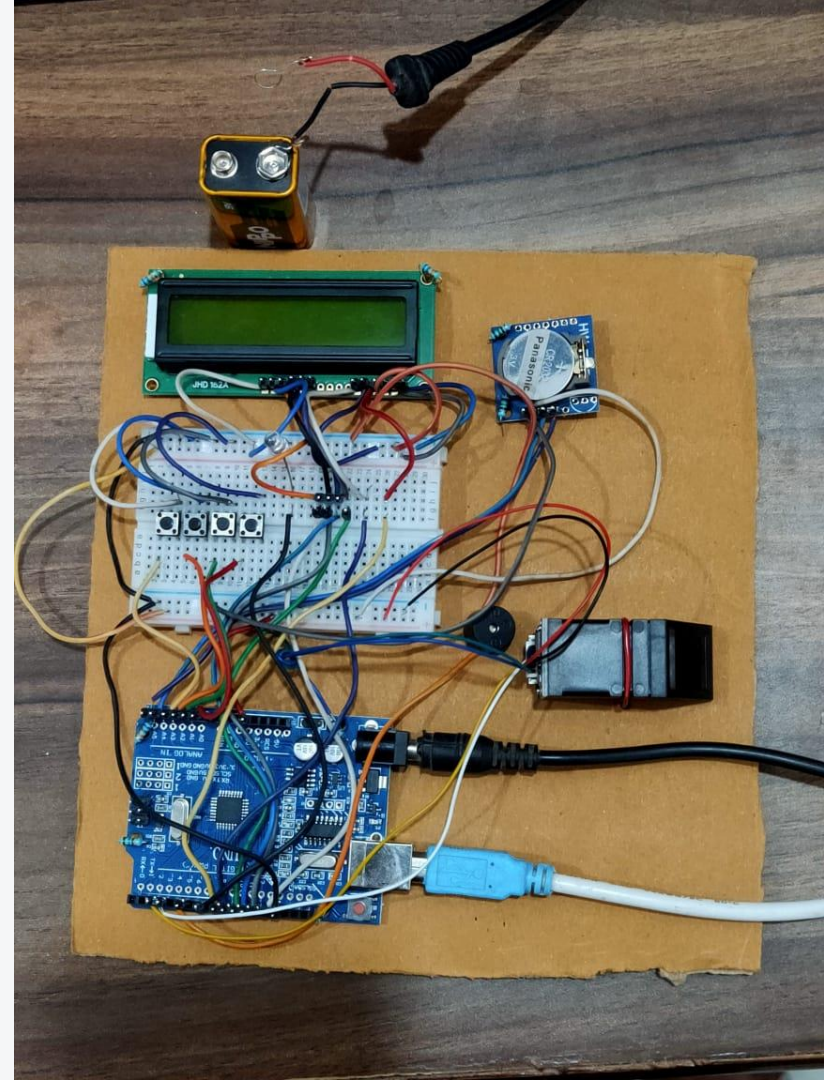
# CIRCUIT DIAGRAM



The circuit of this fingerprint based attendance system 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. 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. Fingerprint module's Rx and Tx directly connected at Serial pin D2 and D3 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.

## REQUIRED COMPONENTS

- Arduino -1
- Finger print module -1
- Push Button - 4
- LEDs -1
- Power
- Connecting wires
- Buzzer -1
- 16x2 LCD -1
- Bread Board -1
- RTC Module -1



## 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 enroll New Fingerprint Click on the Enroll 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.

# SOURCE CODE PROGRAM

## Header Files Used

Adafruit_Fingerprint.h	-----	fingerprint library
RTCLib.h	-----	library file for RTC Module
EEPROM.h	-----	command for storing data
LiquidCrystal.h	-----	lcd header file

The program is coded as to register 4 Fingerprints and can record each users attendance data for 30 times or Days, and every attendance will record time and date so this becomes 7-byte data.

So total memory required is:

$4 * 30 * 7 = 840$  bytes.

```
#define records 4  
int user1,user2,user3,user4;
```

## PARTS OF CODES

```
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(F("Deleted!"));
        lcd.clear();
        lcd.print("Finger Deleted");
        lcd.setCursor(0,1);
        lcd.print("Successfully");
        delay(1000);
    }
}
```

```
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(F("Image taken"));
```

```
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);
    Serial.println();
}
```

```
Serial.print("Found ID #");
Serial.print(finger.fingerID);
return finger.fingerID;
}
```

```
lcd.clear();
lcd.print(" Fingerprint ");
lcd.setCursor(0,1);
lcd.print("Attendance System");
delay(2000);
lcd.clear();
lcd.print("Digital Electronics Project");
lcd.setCursor(0,1);
lcd.print("Sonu and AYush");
delay(2000);
```

```
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);
}
```

```
if (! rtc.begin())
    Serial.println(F("Couldn't find RTC"));
rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
```



# WORKING

To show the working of our Fingerprint Attendance module, we had made two videos.

First one is to show the Registering of Fingerprint in the module. Video will show just one register but we added 3 fingerprint for the demo.

Second video shows the actual attendance being recorded by module. As I put my different finger on the Fingerprint Scanner, arduino registers the attendance.

And finally at last of second video the Database of attendance was downloaded by pressing the RESET key followed by INSERT key (1<sup>st</sup> key).

```
Found fingerprint sensor!
```



# CONCLUSION

Hardware implementation was quite a bit different experience from simulation. To achieve the output we wanted in this project we gave multiple try, because it was not that easy to work with hardware just like simulation.

In the first attempt, the LCD was not able to display any output, may be due to improper connection. Then we tried reconnecting every connection from Beginning. In the second attempt the Display worked, but not quite as we wanted. The RTC input was not proper and the Registration of fingerprint was also not working. I figured out the use of Resistors was quite complicating, so I tried for the 3<sup>rd</sup> time without using any Resistors and Finally in the 3<sup>rd</sup> attempt our finger-print attendance module worked as desired.

## REFERENCES:

- <https://www.arduino.cc/>
- <https://components101.com/>
- <https://www.elprocus.com/>
- <https://www.explainthatstuff.com/>
- <https://www.researchgate.net/>



THANK YOU

THE END

