International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Automated Biometric Attendance Recorder

Apurba Kumar Ghosh[1], Harsha Akash Santra[2]

[1]Associate Professor, Head of the Department of Applied Electronics and Instrumentation Engg. (AEIE), University Institute of Technology, The University of Burdwan, Golapbag North, Burdwan, District: Purba Bardhaman, State: West Bengal, India Pin-713104
Email id: dr.apurbaghosh[at]gmail.com

[2]Pursuing M.Tech. in OOE (Optics and Optoelectronics), Technology Campus, Department of Applied Optics & Photonics, University of Calcutta, JD-2, Sector-III, Salt Lake, State: West Bengal, India, Pin-700106
Corresponding Author Email id: *hasantra11[at]gmail.com*

Abstract: ***This paper exhibits the working of a prototype electronic attendance system, which records attendance by utilizing biometric fingerprint technology. It is a compact, secure, robust, user-friendly, and reliable modern attendance system. It operates at only 5V, which makes this a power efficient device. The users feed their fingerprint through an optical fingerprint sensor, proficient in searching, comparing, collecting and registering utility. Whenever a registered user places their finger on the optical fingerprint sensor, their attendance is registered, along with a time stamp, displayed in hour: minute: second format. The complete set of attendance is available for download from the system. The downloaded attendance set can now be stored for further processing. The LCD displays the instructions related to the biometric pre-registration/authentication, along with the corresponding time stamp. Buzzer creates a 'beep' sound according to the instructions, or, messages.***

Keywords: Biometric, LCD, Prototype, Sensor, Fingerprint

1. Introduction

Managing time is crucial. Attendance management is required by schools, universities, colleges, corporations; in short, by everyone. Throughout the ages, many earlier versions of attendance systems have existed which includes but is not limited to sand clocks, time/punch/stint cards, time sheets, etc. These methods have their own unique shortcomings, such as, faulty machines which could damage a time card thus resulting in loss of wages, duplicating signatures, signing a sheet on behalf of someone else, and so on. Implementation of biometric authentication in the area of attendance recording systems would be an absolute game changer. Thus, in order to make attendance systems more accessible, secure, and economical, compact and robust electronic attendance systems is an essential requirement of the modern day and age.

2. Literature Survey

Emphasis on utilizing attendance has, by tradition, been managed by means of stint sheets, clocks, punch cards, etc. During the latter part of the 1800s, the original time clock documenting operative entry and exit attendance for industrial units was devised. A dense paper card termed 'stint card' was used by the motorized timepiece onto which day and time information was emblazoned. It facilitated the owners possess a tangible notation of the total time put into work, by the workforces. This procedure enabled the members of staff not to be cheated upon by the organization when the time came for their remunerations. It unerringly aided the employer to make sure that the no. of work hours professed by the employees are authentic. The development of stint cards progressed with time. It comprised of denoted distinctive spaces for indicating entry and exit, thus allowing recruits warily position the card precisely. Enormous mechanical contraptions were substituted with littler electrical stint timers, as technology became more erudite. Universities, schools, colleges, private firms, corporations, organizations utilize paper-based attendance system on an unvarying footing. Usage of manually documenting techniques flourished with several enhancements over time. But major shortcomings were detected within; impersonation of sign, or, pilfered attendance copies, human errors, etc. A tiny mistake might ruin a vast amount of opulence, which comprises of secretarial, personnel, and concerned divisions of firms, enterprises, organizations, establishments, etc. Swift progression in technological advancement gave birth to automated attendance compilation via concurrent position systems, enabling cross linkage among the said compilation and performance characteristics. Biometric authentication is a compelling, durable alternative to traditional attendance procedures [1]. The user pre-registers their fingerprint via an optical fingerprint sensor. If the user places their finger on the sensor, the attendance is registered along with the time it was registered at. Thus, the user's presence is corroborated via biometric authentication, realized by biometric fingerprint technology. We have made a prototype which demonstrates an automated electronic biometric system for attendance recording.

3. System Description

This attendance prototype system consists of a R307-TTL UART fingerprint sensor, an Arduino UNO R3, a DS3231 RTC, a 1602 LCD module, 4 tactile push buttons, a buzzer, a breadboard and 9V batteries.

a) R307 TTL-UART Fingerprint Sensor

A bright light is shined on the finger by the optical fingerprint sensor using LEDs. Thus, a picture is taken. The picture undergoes two quality checks. The first quality check ensures that the run of the mill pixel quantity of the print is neither too light, nor too dark. The second quality check comprises of fine resolution inspection of the print, by

observation of variation in light and dark locales, coherent with distinct valleys and ridges present on the finger [2][3]. The print is sent for further processing, provided it passes the two quality checks. It is commonly used in ATMs, airports, electronic door locks, safes, etc. The R307 optical fingerprint sensor, as shown in Fig. 1, operates on the aforementioned principle with newer augmentations such as cutting-edge fingerprint positioning procedure, an expeditious DSP processor, flash microchips of high capacity. It also proffers fingerprint storing, searching, matching, submission, and processing [4][5][6]. It is proficient in amassing up to a 1000 fingerprint templates. The description of the pin layout of the fingerprint sensor is given in Table I.

(a) (b)

Figure 1: (a) front; (b) back of R307 Optical Fingerprint Sensor.

Table I: Elucidation of Pin Layout of R307 Fingerprint Sensor

Pin	Pin Name	Details
1	5V	Regulated 5V DC.
2	GND	Common Ground.
3	TXD	Data output - Connect to MCU RX.
4	RXD	Data Input - Connect to MCU TX.
5	TOUCH	Active Low output when there is a touch on the sensor by finger.
6	3.3V	Use this wire to give 3.3V to the sensor instead of 5V.

b) ARDUINO UNO R3

Arduino UNO R3 is an open-source microcontroller. It is based on the Microchip Atmega328P, created by Arduino. As shown in Fig. 2, it comprises of 14 digital input/output pins. Usage of functions such as pinMode(), digitalRead(), and digitalWrite() via Arduino programming [7] provides access to these pins. The onboard voltage regulator converts the voltage abounding to it to 5 volts. Thus, the pins operate on 5V. They are able to provide or receive an utmost of 40mA current.

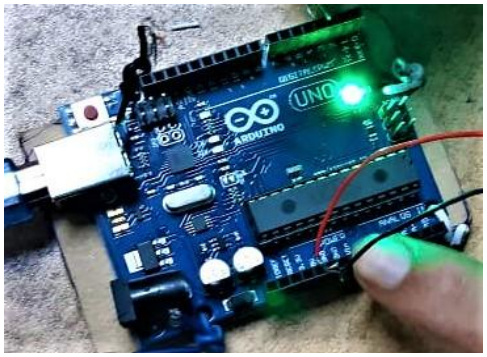


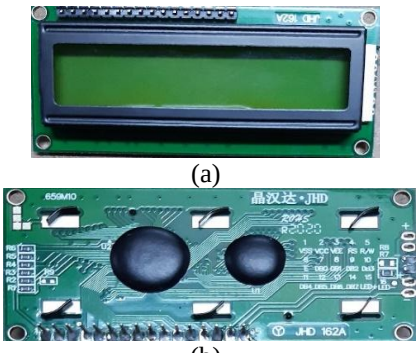
Figure 2: Arduino UNO R3.

c) ARDUINO IDE

Arduino IDE stands for Integrated Development Environment. It is an open-source programming software, invented, engineered, and powered by Arduino. It is an intelligible language, which utilizes a combination of C, C++, and Java. It has devoted libraries for distinct peripherals. It is usually downloaded via the Library Manager, Arduino IDE. The programming written in this environment is well-suited with all the diverse versions of Arduino boards such as, Arduino nano, Arduino Leonardo, Arduino UNO etc. The IDE is affable for creation of open-source developments. Thus, it is favoured by researchers, engineers, students, and people from non-software disciplines. This is one of the most endearing factors of the Arduino IDE in fields of research, development and teaching. The version that we used: ARDUINO 1.8.9.

d) 1602 LCD Module

1602 LCD module, as shown in Fig. 3 has 16 rows and 2 columns. It possesses 32 (16×2) characters in totality. Thus, each character would be composed of 5×8 dots (pixel).The pin configuration is given in Table II.



(a) (b)

Figure 3: (a) front; (b) back of 1602 LCD Module.

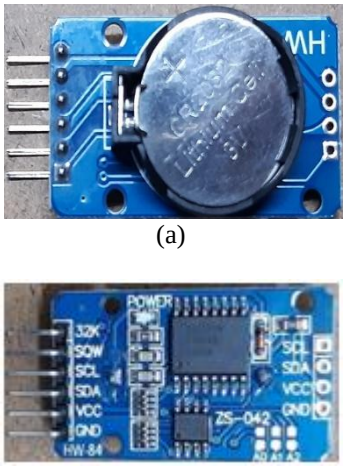
Table II: Pin Layout Elucidation of 1602 LCD Module

Pin	Pin Name	Description
1	Vss (Ground)	Ground pin connected to system ground.
2	Vdd (+5 Volt)	Powers the LCD with +5V (4.7V – 5.3V).
3	VE (Contrast V)	Decides the contrast level of display. Grounded to get maximum contrast.
4	Register Select	Connected to Microcontroller to shift between command/data register
5	Read/Write	Used to read or write data. Normally grounded to write data to LCD.
6	Enable	Connected to Microcontroller Pin and toggled between 1 and 0 for data acknowledgement.
7	Data Pin 0	Data pins 0 to 7 forms an 8-bit data line. They can be connected to Microcontroller to send 8-bit data. These LCD's can also operate on 4-bit mode in such case Data pin 4,5,6 and 7 will be left free.*
8	Data Pin 1	*
9	Data Pin 2	*
10	Data Pin 3	*

11	Data Pin 4	*
12	Data Pin 5	*
13	Data Pin 6	*
14	Data Pin 7	*
15	LED Positive	Backlight LED pin positive terminal.
16	LED Negative	Backlight LED pin negative terminal.

e) DS3231 RTC MODULE

RTC stands for real time clock. It comprises of an integrated chip and offers precise date and time. It runs on a lithium battery which facilitates the RTC to function even if the system suffers a power-cut. Thus, RTC plays a crucial task, when involved in real time systems such as, digital camera, digital clock, places where time stamp is vital [8]. DS3231, as shown in Fig. 4 uses 6 pins to send data to the system or receive data from the system, via I2C interface. The pin layout of DS3231 RTC module is shown in Table III.



(a)

(b)

Figure 4: (a) front; (b) back of the RTC DS3231 Module.

Table III: Pin Layout Elucidation of DS3231 RTC Module

Pin Name	Elucidation
V_{CC}	It is connected to the positive terminal of power supply.
GND	It is connected to the ground terminal.
SDA	It stands for Serial Data pin, comprising of I2C interface.
SCL	It stands for Serial Clock, comprising of I2C Interface.
SQW	It stands for Square Wave output.
32K	It provides a 32K oscillator yield.

4. Working Principle

The users register their fingerprint through an optical fingerprint sensor, proficient in fingerprint registration, collection, search, and comparison. The third and fourth button are used to navigate through the User IDs to select the one that the users have been assigned to. The second push button is used to confirm the selection of the desired ID. After choosing the ID, the device asks the user to put her/his finger on the optical fingerprint sensor. The fingerprint of the user is now stored in the memory of the sensor. The user can delete their fingerprint if they didn't choose the correct ID. Deletion of the fingerprint is also achieved by pressing the second push button. The RTC is simultaneously providing real time output when idle. Whenever a registered user places her/his finger on the optical fingerprint sensor, their attendance is registered, along with a time stamp, displayed in hour: minute: second format. The complete set of attendance can be downloaded from the device by pressing the RESET button present on the Arduino UNO, and, the first tactile push button, together. The downloaded attendance set can now be stored for further processing. A block diagram of the system is given, as shown in Fig. 5.

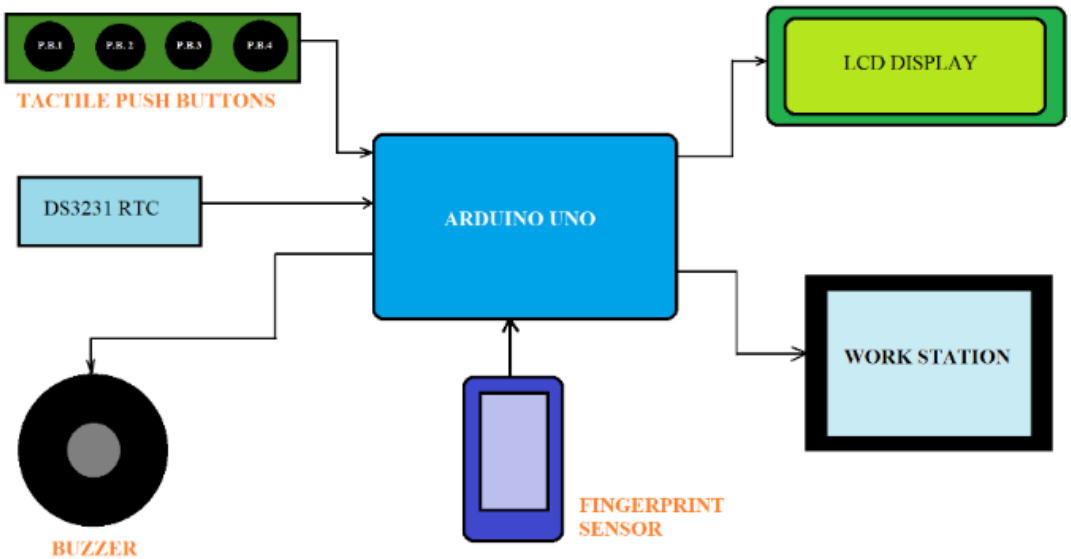


Figure 5: Block Diagram of the Automatic Biometric Attendance Recorder.

5. Methodology

The TX wire and RX wire of finger print sensor is connected to pin 2 and pin 3 of Arduino UNO, with the help of jumper cables (to keep the connection strong). The red wire and the black wire of the fingerprint sensor is connected to the analog 5V pin and the analog GND of the Arduino UNO. The first sketch is uploaded to the UNO, as given in *Appendix I*. It enables the user to circumvent the Atmega chip and links up the fingerprint sensor unswervingly to the USB/Serial chip converter [9]. Now, the interaction between the fingerprint sensor and the UNO has been optimized. The second sketch is uploaded next, which is given in *Appendix II*. This sketch forms the basis of our prototype. It will help interface all the components of the design, when the circuit formation is established. All the pins of 1602 LCD, UNO, DS3231 RTC Module, buzzer, push buttons etc. are properly connected as per their pin configuration, as shown in the following tables.

Table IV: Arduino Uno

Pin	Connection
Positive terminal	Digital Pin 5 (UNO)
Ground terminal	Ground (Solderless Board)

Table V: Buzzer

Pin No.	Connection
Analog 5V	Supply (Solderless Board)
GND	Ground (Solderless Board)

Table VI: R307 Optical Fingerprint Sensor

Pin No.	Connection
1	Ground (Solderless board)
2	Supply (Solderless board)
3	Ground (Solderless board)
4	Digital Pin 8 (UNO)
5	Ground (Solderless board)
6	Digital Pin 9(UNO)
11	Digital Pin 10 (UNO)
12	Digital Pin 11(UNO)
13	Digital Pin 12(UNO)
14	Digital Pin 13(UNO)
15	Supply (Solderless board)
16	Ground (Solderless board)

Table VII: 1602 LCD Module

Pin	Connection
TX	Digital Pin 2 (UNO)
RX	Digital Pin 3 (UNO)
V_{CC}	Supply (Solderless Board)
GND	Ground (Solderless Board)

Table VIII: Tactile Push Buttons

Push Button	Connection	
P.B. 1	Supply	Analog Pin A0 (UNO)
	GND	Ground (Solderless Board)
P.B. 2	Supply	Analog Pin A1 (UNO)
	GND	Ground (Solderless Board)
P.B. 3	Supply	Analog Pin A2 (UNO)
	GND	Ground (Solderless Board)
P.B. 4	Supply	Analog Pin A3 (UNO)
	GND	Ground (Solderless Board)

Table IX: DS3231 RTC Module

Pin Name	Connection
SDA	Analog Pin A4 (UNO)
SCL	Analog Pin A5 (UNO)
GND	Ground (Solderless board)
5V	Supply (Solderless board)

6. Experimental Results

The system is designed, as shown in Fig. 6. Power supply to the UNO is switched ON. The record of the previous attendances (if any)can be displayed by clicking *tools* inArduino IDE, and then selecting *Serial Monitor* from the dropdown menu when the Arduino UNO is connected to a PC/Laptop via a USB cable. To skip this process, the user may set the value of 'database' to 0 in the program, given in *Appendix II*.To delete all the attendances stored, the user can change the value of 'delete_database' to 1 in the program as well. It starts with the LCD module displaying the message "Automated Biometric Attd. Recorder". Now, the user has to push the first button for fingerprint enrollment, or, pre-registration. The third and fourth button are used to navigate through the IDs which are currently empty. The user selects the desired ID by pushing the second button. The system then asks the user to put their finger on the fingerprint sensor. Following this procedure, the fingerprint is enrolled. If the user places the enrolled finger on the sensor, the attendance of the user is registered, along with the time stamp, generated by the RTC. The buzzer creates a 'beep' sound at every stage of functioning of the system, from enrollment of fingerprint, to authentication, attendance registration, or, deletion of fingerprint. When the user places their enrolled finger on the fingerprint sensor, the attendance of the user is registered, along with a time stamp, generated by the RTC module. If an un-enrolled finger is placed on the sensor, it returns the messages "Finger is absent" and "Try over". The output result of the three conditions, that is, (i) enrolled finger is placed, (ii) sensor is unused, and (iii) an unenrolled finger is placed, are given in Table X.

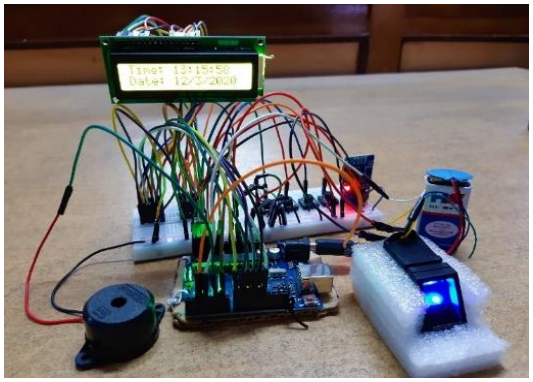


Figure 6: The experimental setup of "Automated Biometric Attendance Recorder"

Table X: Attendance Registration Procedure Using Biometric Authentication

S. No.	Condition	Status	Result
1.	Enrolled finger is placed.	Biometric authentication initiated.	Attendance registered.
2.	Sensor is unused.	System is idle.	Displays date, along with real time on the 1602 LCD module.
3.	Unenrolled finger is placed.	Biometric authentication initiated.	Shows the messages “Finger is absent” and “Try Over” on LCD module.

7. Conclusion

We have developed a secure, robust, user-friendly, cost effective, automated electronic attendance system, by using an Arduino UNO. It would be able to store attendances in real time, and the attendance can be accessed and stored for further processing.

Appendix

Appendix I

```
void setup() {}
void loop() {}
```

Appendix II

```
#include <Adafruit_Fingerprint.h>
#include<LiquidCrystal.h>
#include<EEPROM.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include <RTClib.h>
LiquidCrystal dis_lc(8,9,10,11,12,13);
SoftwareSerial prin_fing(2, 3);
RTC_DS3231 rtc;
uint8_t t;
Adafruit_Fingerprint pfing = Adafruit_Fingerprint(&prin_fing);
#define Preg_b 14
#define rem_c 15
#define front 16
#define back 17
#define comp 5
#define fignid 7
#define buz 5
#define database 0
#define delete_database 1
#define db 10
int us1,us2,us3,us4,us5,us6,us7,us8,us9,us10;
DateTime now;
void setup()
{delay(1000);
dis_lc.begin(16,2);
Serial.begin(9600);
pinMode(Preg_b, INPUT_PULLUP);
pinMode(front, INPUT_PULLUP);
pinMode(back, INPUT_PULLUP);
pinMode(rem_c, INPUT_PULLUP);
pinMode(comp, INPUT_PULLUP);
pinMode(buz, OUTPUT);
pinMode(fignid, OUTPUT);
digitalWrite(buz, LOW);
if(database==0)
{digitalWrite(buz, HIGH);
delay(500);
digitalWrite(buz, LOW);
dis_lc.clear();
dis_lc.print("Processing:= !");
dis_lc.setCursor(0,1);
dis_lc.print("Data Download initiated:=");
Serial.println("Dear User, do wait:=");
Serial.println("Data is being downloaded:=");
Serial.println();
Serial.print("SNo. ");
for(int i=0;i<db;i++)
{digitalWrite(buz, HIGH);
delay(500);
digitalWrite(buz, LOW);
Serial.print("ID of user");
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(delete_database==0){
dis_lc.clear();
dis_lc.print("Currently");
dis_lc.setCursor(0,1);
dis_lc.print("Resetting");
for(int i=1000;i<1005;i++)
EEPROM.write(i,0);
for(int i=0;i<841;i++)
EEPROM.write(i, 0xff);
dis_lc.clear();
dis_lc.print("Reset Done");
delay(1000);}
dis_lc.clear();
dis_lc.print(" Automated Biometric ");
dis_lc.setCursor(0,1);
dis_lc.print("Attd. Recorder");
```

```
delay(2000);
dis_lc.clear();
digitalWrite(buz, HIGH);
delay(500);
digitalWrite(buz, LOW);
for(int i=1000;i<1000+db;i++)
{if(EEPROM.read(i) == 0xff)
EEPROM.write(i,0);}
pfing.begin(57600);
Serial.begin(9600);
dis_lc.clear();
dis_lc.print("Searching:=");
dis_lc.setCursor(0,1);
delay(2000);
if (pfing.verifyPassword())
{Serial.println("R307 Fingerprint Sensor available:=");
digitalWrite(buz,HIGH);
delay(100);
digitalWrite(buz,LOW);
delay(100);
digitalWrite(buz,HIGH);
delay(100);
digitalWrite(buz,LOW);
delay(100);
digitalWrite(buz,HIGH);
delay(100);
digitalWrite(buz,LOW);
delay(100);
dis_lc.clear();
dis_lc.print("OPS available:=");
delay(2000);}
else
{Serial.println("OPS offline:=");
digitalWrite(buz,HIGH);
delay(100);
digitalWrite(buz,LOW);
delay(100);
digitalWrite(buz,HIGH);
delay(100);
digitalWrite(buz,LOW);
delay(100);
dis_lc.clear();
dis_lc.print("Module unavailable");
dis_lc.setCursor(0,1);
dis_lc.print("Please Check");
while (1);}
if (! rtc.begin())
Serial.println("Unable to find the RTC");
rtc.adjust(DateTime(2020, 3, 12, 13, 11, 0));
if (rtc.lostPower())
{Serial.println("RTC is absent");}
dis_lc.setCursor(0,0);
dis_lc.print(" Press PB 1 to ");
dis_lc.setCursor(0,1);
dis_lc.print(" initiate");
delay(3000);
us1=EEPROM.read(1000);
us2=EEPROM.read(1001);
us3=EEPROM.read(1002);
us4=EEPROM.read(1003);
us5=EEPROM.read(1004);
dis_lc.clear();
digitalWrite(fignid, HIGH);}
void loop()
{now = rtc.now();
dis_lc.setCursor(0,0);
dis_lc.print("Time: ");
dis_lc.print(now.hour(), DEC);
dis_lc.print(':');
dis_lc.print(now.minute(), DEC);
dis_lc.print(':');
dis_lc.print(now.second(), DEC);
dis_lc.print(" ");
dis_lc.setCursor(0,1);
dis_lc.print("Date: ");
dis_lc.print(now.day(), DEC);
dis_lc.print('/');
dis_lc.print(now.month(), DEC);
dis_lc.print('/');
dis_lc.print(now.year(), DEC);
dis_lc.print(" ");
delay(500);
int reu=getFingerprintIDez();
if(reu>0)
{digitalWrite(fignid, LOW);
digitalWrite(buz, HIGH);
delay(100);
digitalWrite(buz, LOW);
dis_lc.clear();
dis_lc.print("ID:");
dis_lc.print(reu);
dis_lc.setCursor(0,1);
dis_lc.print("Do wait:=");
delay(1000);
attendance(reu);
dis_lc.clear();
digitalWrite(buz,HIGH);
delay(100);
digitalWrite(buz,LOW);
delay(100);
digitalWrite(buz,HIGH);
delay(100);
digitalWrite(buz,LOW);
delay(100);
dis_lc.print("Attendance ");
dis_lc.setCursor(0,1);
dis_lc.print("Registered");
delay(1000);
digitalWrite(fignid, HIGH);
return;}
checkKeys();
delay(300);}
void attendance(int t)
{int us=0,eepLoc=0;
if(t == 1)
{eepLoc=0;
us=us1++;}
else if(t == 2)
{eepLoc=210;
us=us2++;}
else if(t == 3)
{eepLoc=420;
us=us3++;
}else if(t == 4)
{eepLoc=630;
us=us4++;}
```

```
else if(t == 5)
{eepLoc=0;
us=us5++;}
else if(t == 6)
{eepLoc=840;
us=us5++;}
else if(t == 7)
{eepLoc=1050;
us=us7++;}
else if(t == 8)
{eepLoc=1260;
us=us8++;}
else if(t == 9)
{eepLoc=1470;
us=us9++;}
else if(t == 10)
{eepLoc=1680;
us=us8++;}
else
return;
int eepIndex=(us*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,us1);
EEPROM.write(1001,us2);
EEPROM.write(1002,us3);
EEPROM.write(1003,us4);}
void checkKeys()
{if(digitalRead(Preg_b) == 0)
{dis_lc.clear();
dis_lc.print("Do Wait:=");
delay(1000);
while(digitalRead(Preg_b) == 0);
Enroll();}
else if(digitalRead(rem_c) == 0)
{dis_lc.clear();
dis_lc.print("Do Wait:=");
delay(1000);
del();}}
void Enroll()
{int c=1;
dis_lc.clear();
dis_lc.print("Enter Finger ID:");
while(1)
{dis_lc.setCursor(0,1);
dis_lc.print(c);
if(digitalRead(front) == 0)
{c++;
if(c>db)
c=1;
delay(500);}
else if(digitalRead(back) == 0)
{c--;
if(c<1)
c=db;
delay(500);}
else if(digitalRead(rem_c) == 0)
{t=c;
getFingerprintEnroll();
for(int i=0;i<db;i++)
{if(EEPROM.read(i) != 0xff)
{EEPROM.write(i, t);
break;}}
return;}
else if(digitalRead(Preg_b) == 0)
{return;}}}
void del()
{int c=1;
dis_lc.clear();
dis_lc.print("Enter Finger ID");
while(1)
{dis_lc.setCursor(0,1);
dis_lc.print(c);
if(digitalRead(front) == 0)
{c++;
if(c>db)
c=1;
delay(500);}
else if(digitalRead(back) == 0)
{c--;
if(c<1)
c=db;
delay(500);}
else if(digitalRead(rem_c) == 0)
{t=c;
deleteFingerprint(t);
for(int i=0;i<db;i++)
{if(EEPROM.read(i) == t)
{EEPROM.write(i, 0xff);
break;}}
return;}
else if(digitalRead(Preg_b) == 0)
{return;}}}
uint8_t getFingerprintEnroll()
{int p = -1;
dis_lc.clear();
dis_lc.print("Finger ID:");
dis_lc.print(t);
dis_lc.setCursor(0,1);
dis_lc.print("Position Finger");
delay(2000);
while (p != FINGERPRINT_OK)
{p = pfing.getImage();
switch (p)
{case FINGERPRINT_OK:
Serial.println("Picture Captured:=");
dis_lc.clear();
dis_lc.print("Pic. Captured");
break;
case FINGERPRINT_NOFINGER:
Serial.println("Finger unavailable:=");
dis_lc.clear();
dis_lc.print("Finger unavail.");
break;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Error in dispatch:=");
dis_lc.clear();
dis_lc.print("Error D");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Error in Dispatch=");
```

```
dis_lc.clear();
dis_lc.print("Error D");
break;
default:
Serial.println("Unidentified Issue");
dis_lc.clear();
dis_lc.print("Problem");
break;}}
p = pfing.image2Tz(1);
switch (p)
{case FINGERPRINT_OK:
Serial.println("Picture translated:=");
dis_lc.clear();
dis_lc.print("Pic. Trans.");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Picture is muddled:=");
dis_lc.clear();
dis_lc.print("Pic. muddled");
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Error in dispatch:=");
dis_lc.clear();
dis_lc.print("Error in D");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Unable to find your fingerprint characteristics:=");
dis_lc.clear();
dis_lc.print("Char. not found");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Unable to find your fingerprint characteristics:=");
dis_lc.clear();
dis_lc.print("Char. not found");
return p;
default:
Serial.println("Unidentified problem:=");
dis_lc.clear();
dis_lc.print("Problem");
return p;}

Serial.println("Please reomve your finger:=");
dis_lc.clear();
dis_lc.print("Take off finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER)
{p = pfing.getImage();}
Serial.print("ID "); Serial.println(t);
p = -1;
Serial.println("Place same finger again");
dis_lc.clear();
dis_lc.print("Place Finger");
dis_lc.setCursor(0,1);
dis_lc.print(" Again");
while (p != FINGERPRINT_OK)
{p = pfing.getImage();
switch (p)
{case FINGERPRINT_OK:
Serial.println("Picture Captured:=");
break;
case FINGERPRINT_NOFINGER:
Serial.print(".");
break;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Error in Dispatch:=");
break;
case FINGERPRINT_IMAGEFAIL:
Serial.println("Picture construction issue:=");
break;
default:
Serial.println("Unidentified Issue:=");
return;}}
p = pfing.image2Tz(2);
switch (p)
{case FINGERPRINT_OK:
Serial.println("Pic trans.");
break;
case FINGERPRINT_IMAGEMESS:
Serial.println("Pic. muddled");
return p;
case FINGERPRINT_PACKETRECIEVEERR:
Serial.println("Error in D");
return p;
case FINGERPRINT_FEATUREFAIL:
Serial.println("Unable to find your fingerprint characteristics:=");
return p;
case FINGERPRINT_INVALIDIMAGE:
Serial.println("Unable to find your fingerprint characteristics:=");
return p;
default:
Serial.println("Unidentified issue:=");
return p;}Serial.print("Constructing for :"); Serial.println(t);
p = pfing.createModel();
if (p == FINGERPRINT_OK)
{Serial.println("Your fingerprints are compatible:=");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println("Error in Dispatch:=");
return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
Serial.println("Your fingerprints are incompatible:=");
return p;
} else {Serial.println("Unidentified issue:=");
return p;}
Serial.print("ID"); Serial.println(t);
p = pfing.storeModel(t);
if (p == FINGERPRINT_OK) {
Serial.println("Your fingerprint is now stored:=");
dis_lc.clear();
dis_lc.print("Amassed");
delay(2000);} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
Serial.println("Error in dispatch:=");
return p;}else if (p == FINGERPRINT_BADLOCATION) {
Serial.println("Unable to store in the said location:=");
return p;} else if (p == FINGERPRINT_FLASHERR)
{Serial.println("Flash Error:=");
return p;
}else {
Serial.println("Unidentified issue:=");
return p;}}
int getFingerprintIDez()
```

```
{uint8_t p = pfing.getImage();
if (p != FINGERPRINT_OK)
return -1;
p = pfing.image2Tz();
if (p != FINGERPRINT_OK)
return -1;
p = pfing.fingerFastSearch();
if (p != FINGERPRINT_OK)
{dis_lc.clear();
dis_lc.print("Finger is absent");
dis_lc.setCursor(0,1);
dis_lc.print("Try over");
delay(2000);
return -1;}
Serial.print("Your ID:");
Serial.print(pfing.fingerID);
Serial.println();
return pfing.fingerID;}
uint8_t deleteFingerprint(uint8_t t)
{uint8_t p = -1;
dis_lc.clear();
dis_lc.print("Do wait");
p = pfing.deleteModel(t);
if (p == FINGERPRINT_OK)
{Serial.println("Your fingerprint has been deleted:=");
dis_lc.clear();
dis_lc.print("Deletion");
dis_lc.setCursor(0,1);
dis_lc.print("Successful");
delay(1000);
}else
{Serial.print("Unknown problem has arised");
dis_lc.clear();
dis_lc.print("Unknown problem");
dis_lc.setCursor(0,1);
dis_lc.print("Please try 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++));
Serial.print("    ");}
Else
{Serial.print("**********************************
");}
Serial.print("      ");
```

8. Conflict of Interest

The authors do not have any conflict of interest.

9. Author Contributions

Apurba Kumar Ghosh: Guidance, and revision of the paper; Harsha Akash Santra: Circuit designing and implementation, program writing and original draft writing.

Acknowledgment
The authors would like to thank the Department of Applied Electronics and Instrumentation Engineering, University Institute of Technology, The University of Burdwan for supporting us on this endeavour.

10. Future Scope

Combining different types of biometric recognition processes to make authentication more accurate.

References

[1] Shoewu, O. and O.A. Idowu, "Development of Attendance Management System using Biometrics", *Pacific Journal of Science and Technology*. 13(1):300-307, 2012.
[2] Lunji Qiu, "Fingerprint Sensor Technology"**,** *9th IEEE Conference on Industrial Electronics and Applications, 2014.*
[3] An, B.W., Heo, S., Ji, S. *et al. "*Transparent and flexible fingerprint sensor array with multiplexed detection of tactile pressure and skin temperature", Nat *Commun***9,** 2458(2018).
[4] Kaoru Uchida, "Detection and Recognition Technologies-Fingerprint Identification", *NEC Journal of Advanced Technology*, Vol. 2, No. 1(Jan, 2005).
[5] Martin Aastrup Olsen, Vladimír Šmida, Christoph Busch, "Finger image quality assessment features – definitions and Evaluation", *IET Biometrics* (Volume: 5, Issue: 2, 6 2016).
[6] Subban and Dattatreya P. Mankame, "A Study of Biometric Approach Using Fingerprint Recognition", *Lecture Notes on Software Engineering*, Vol. 1, No. 2, May 2013.
[7] Y. A. Badamasi, "The working principle of an Arduino", 2014 *11th International Conference on Electronics, Computer and Computation (ICECCO)*, Abuja, 2014, pp. 1-4.
[8] Gay W.W.*"Real-Time Clock. In: Mastering the Raspberry Pi."* Apress, Berkeley, CA, 2014.
[9] Limor Fried, *"Adafruit Optical Fingerprint Sensor"*, adafruit learning system, Adafruit Industries, (Dec, 2019).

Author Profile

Dr. Apurba Kumar Ghoshis presently working as Associate professor in University Institute of Technology, Burdwan University. He did his M.E. and Ph.D. in ETCE from Jadavpur University and B.TECH. in IEE from Jadavpur University. His research area includes advanced control algorithm in the area of process control, fibre optic sensors, industrial automation, etc.

Harsha Akash Santra was born in West Bengal, India. In the year 2020, he completed his Bachelor of Engineering in Applied Electronics & Instrumentation Engineering, from University Institute of Technology, The University of Burdwan, India. He is currently pursuing his master degree in Optics & Optoelectronics from University of Calcutta, India.