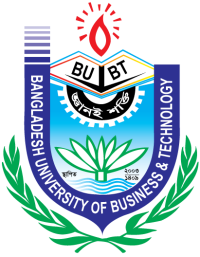
***Project Report***

***On***

***Arduino-uno-based-fingerprint-attendance-system***



***Bangladesh University of Business and Technology (BUBT)***



**LEARN YOURSELF**

***A project***

***Submitted to the Department of Computer Science and engineering***

***Bangladesh University of Business & Technology (BUBT), Dhaka***

***In partial fulfillment of requirements. For the degree***

***Of***

***BACHELOR OF SCIENC***

***IN***

***COMPUTER SCIENCE AND ENGINEERING***

***SUBMITTED BY:***

Md. Rakib Rahian(ID:17182203010)

Md Razaul Haque (ID:17182203035)

Parvin akter (ID: 17182203006)

Arnest Bandhan Biswas(ID: 17182203022)

Shahida Akter Moon(ID:17182203050)

***SUBMITTED TO:***

***(Prof. M. Ameer Ali)***

***Assistant Professor***

***Department of CSE***

***Bangladesh University of Business and Technology (BUBT)***

***Date of Submission:***

**ACKNOWLEGEMENT**

I would like to express my special thanks of gratitude to our teacher Prof. M. Ameer Ali, Department of Computer Science and Engineering, Bangladesh University of Business and Technology Who gave me the golden opportunity to do this project on the topic **Adriano-Uno-based-fingerprint-attendance-system.**

Which also helped me in doing a lot of Research and we came to know about so many new things. We are really thankful to him.

Secondly, I would like to thanks to all of my friends, group member and those who helped, inspired and gave us mental support at different stages in different stages in different moment in my project.

**------------------------ --------------------------------**

Name: Md. Rakib Rahian Name: Md Razaul Haque

ID: 17182203010 ID: 17182203035

Intake: 29th Intake: 29th

Name: Arnest Bandhan Biswas Name: parvin akter

ID: 17182203022 ID: 17182203006

Intake: 29th Intake: 29th

-------------------------------

Name: Shahida Akter Moon

ID: 17182203050

Intake: 29th

**TABLE OF CONTENTS**

**Chapter Topic**

**Abstract……………………………………………………………….06**

**Chapter 1- Introduction……………………………...8**

* 1. Introduction……………………………………………..8
  2. Objective ………………………………………………7
  3. Motivation ……………………………………………..7
  4. R307/R305 Fingerprint sensor module. ………………..9
  5. Working principle of fingerprint sensor. ……………….9-10
  6. Overview and benefits of the project. …………………..10

**Chapter 2- Theory ……………………………………….11**

2.1 R307 Fingerprint Sensor………………………………11

2.2 Operation Principle……………………………………11

2.3 Technical Parameters………………………………….12

2.4 Features………………………………………………..12

2.5 Working of fingerprint sensor and push buttons………12

2.6 Implementation……………………………………….12-13

2.7 Pin Configuration. ……………………………………13-14

**Chapter 3- Hardware Modeling ……………......15**

3.1 Prototype Modeling……………………………………….15

3.2 Main features of the prototype……………………………15

3.3 Overview of the Project. …………………………………..15

3.4 Hardware Connection……………………………………...16

3.4.1 Prototype hardware connection. ………………………16

3.4.2 Detailed Hardware Description……………………….17-18

**Chapter 4- Logic & Operations ………....19**

4.1 Introduction………………………………………………19

4.2 Flow Diagram. ………………………………………….19

4.3 Principle & Operations…………………………………19-20

4.4 Advantages of the project………………………………...20

4.5 Disadvantages of the project. ……………………………..20

4.6 Coding. ……………………………………………22-31

4.8 Circuit Diagram………………………………………..31

**Chapter 5- Conclusion…………………………22**

5.1 Conclusion………………………………………………..32

5.2 Results…………………………………………………….32

**References……………………………………………………………………..……32**

**ABSTRAC**

In industrial and domestic applications attendance registering is important at each and every moment. Many face a lot of problems due to lack of proper attendance monitoring system. In this project we use Fingerprint Sensor (R307) which senses the Fingerprint of a particular person; a buzzer and Led gets activated whenever a person places his finger on the sensor. Then the fingerprint is stored in cloud with id no. Many people can store their fingerprints. Then next time any person puts their finger on the sensor it checks there are any matching fingerprints or not. If his fingerprint matches with any of the stored fingerprints then the LCD display shows which person it is and the time & date of checking.

In this model, all the fingerprints are stored each and every time someone places his finger. User can connect the system wirelessly with the cloud and monitor the process. When the app is running on the computer, data sent by R307 fingerprint module is received and stored on the cloud and displayed in serial monitor and 16\*2 LCD display module.

This study has mainly focused to develop IOT based biometric attendance system that is able to keep record of attendance and count the data for daily purpose. In this project we are going to design Fingerprint Sensor Based Biometric Attendance System using Adriano. Simply we will be interfacing fingerprint sensor with Adriano, LCD Display & RTC Module to design the desired project. In this project, we are using fingerprint Module and Adriano to take and keep attendance data and records. Attendance systems are commonly used systems to mark the presence in offices and schools the unique nature of fingerprint makes it ideal for use in attendance management systems. Besides being secure, Fingerprint based attendance system will also be environment friendly. Fingerprint matching is widely used in forensics for a long time.

**Objective**

The aim of this system is to implement in C.net set of reliable techniques for fingerprint image enhancement and minutiae extraction. The performance of these techniques will be evaluated on a fingerprint data set. In combination with these development techniques, statistical experiments can then be performed on the fingerprint data set. The results from these experiments can be used to help us better understand what is involved in determining the statistical uniqueness of fingerprint minutiae.

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

**Motivation**

The motivation to develop this project is to solve some problems that are currently occurring in every colleges/universities. This project purpose is to improve the current paper-based traditional attendance management system that is still in use by many colleges and universities. From the observation, most of the problems found are normally caused by the use of traditional attendance system in these colleges and universities. Therefore, a fingerprint-based student attendance management system will be developed in order to solve these problems. The system is believed will be needed in order to improve the ways the colleges/universities in managing their student’s attendance.

**CHAPTER- 1**

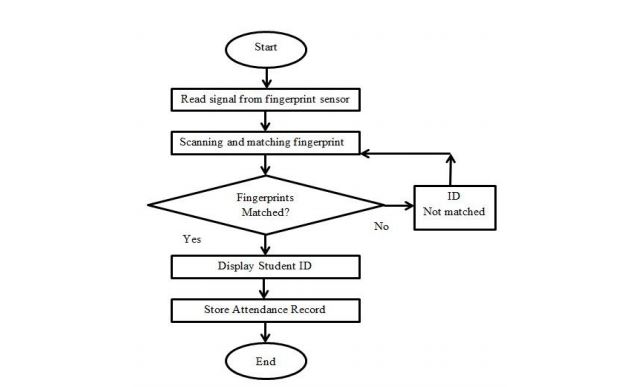
**INTRODUCTION**

**1.1 Introduction**

In the World of Technology, Biometrics plays an effective role in identifying Human beings. Through this project, you will develop a unique system that can identify students for attendance purpose using their fingerprints. In this project, we are going to design a Fingerprint Sensor Based Biometric Attendance System using Adriano. 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.

You will need an Adriano Uno board for interfacing microcontroller with the Finger Print Scanner R307/R305. So with the help of Finger Print Scanner R307/R305, we will store the finger prints of all the students and once they are stored, the Finger Print Scanner will compare the present finger print on the scanner and previously stored finger prints. If any finger print is matched, the microcontroller will print the concern data stored for the particular finger print on the LCD Display. In addition to this, we can add Wi-Fi module, to upload the data into remote cloud, so as to access the entire unit from the sole system of it from anywhere in the world.

Attendance plays a major role in educational institutions. The most common means of taking attendance in the classroom is by calling out the roll numbers of students or asking the students to manually sign the attendance sheet, which is passed around during the lecture. The process of manually taking and maintaining the attendance records becomes highly cumbersome.

****

**Figure 1: Working of Fingerprint based Attendance System**

**Pre-requisites** We will be writing two Arduino scripts for this program. One for the ESP8266-01 Module and the other is for Adriano UNO.. Hence we will write two codes, one for Arduino in which it will communicate with the FPS and send the obtained values via software serial to ESP8266. The other code will be written for ESP8266 which will enable the module to be connected to the Things board server and then will receive the values from Arduino through serial communication to update them on Things board Dashboard.

**1.2 R307/R305 Fingerprint sensor module**

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 Adriano 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:**

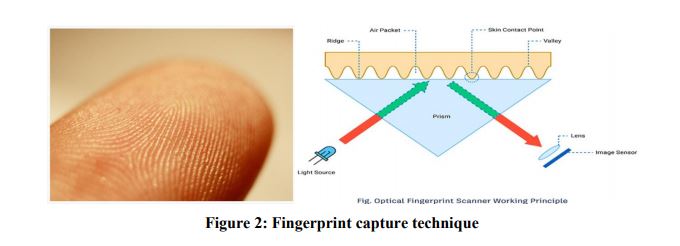
1. Integrated image collecting and algorithm chip together, All-in-one
2. Fingerprint can conduct secondary development & embedded into a variety of end products

3. Low power consumption, low cost, small size, excellent performance

1. Professional optical technology, precise module manufacturing techniques
2. Good image processing capabilities can successfully capture image up to resolution 500 dpi Fingerprints are one of the many unique biometric signatures which we can use to identify people very accurately. But just by holding someone's hand and staring at their fingers can't be practical [grins]; we're not good at it. But computers are good at recognizing and matching 8 patterns very fast and accurately. Before we can process a fingerprint pattern with a computer, we must "capture" it. R307 Fingerprint Module consists of optical fingerprint sensor, high-speed DSP processor, high-performance fingerprint alignment algorithm, high-capacity FLASH chips and other hardware and software composition, stable performance, simple structure, with fingerprint entry, image processing, fingerprint matching, search and template storage and other functions.

**1.3 Working principle of fingerprint sensor**

The skin on the palms of our hands has a special pattern called friction ridges that help us grab things effectively without slipping. These patterns consist of ridges and valleys arranged in certain configurations and are unique for each individual. Our finger tips also have them as you can see from the above image. When a finger comes in contact with a surface, the ridges make strong contact with the surface. When we strongly grab something, the moisture, oil, dirt and dead skin cells on our finger can attach to the surface of the material, leaving an impression we call a fingerprint. Various forensic methods involving the use chemicals are used to extract such fingerprints from crime scenes and are called latent fingerprints. But an optical fingerprint scanner works a bit differently. Can support us by registering on our website if you are prepared to donate your blood when required. As a proud member of this community and as a responsible human. You can help a person in need. So, donate blood digitally.



An optical fingerprint scanner works based on the principle of Total Internal Reflection (TIR). In an optical fingerprint scanner, a glass prism is used to facilitate TIR. Light from an LED (usually blue color) is allowed to enter through one face of the prism at a certain angle for the TIR to occur. The reflected light exits the prism through the other face where a lens and an image sensor or the camera or reflector inside it (essentially camera) are placed

When there's no finger on the prism, the light will be completely reflected off from the surface, producing a plain image in the image sensor. When TIR occurs, a small amount of light leaked to the external medium and it is called the Evanescent Wave. Materials with different refractive indexes (RI) interact with the evanescent wave differently. When we touch a glass surface, only the ridges make good contact with it. The valleys remain separated from the surface by air packets. Our skin and air have different RIs and thus affect the evanescent field differently. This effect is called Frustrated Total Internal Reflection (FTIR). This effect alters the intensities of the internally reflected light and is detected by the image sensor. The image sensor data is processed to produce a high contrast image which will be the digital version of the fingerprint.

**1.4 Overview and benefits of the project**

Remote Control Technology’s line of dependable, durable wireless remote switching systems can and will make money for us and our business. Wireless remote control benefits include:

**No legal issues**

Obtaining access to or traversing properties with hard lines is extremely difficult

**No copper wire to steal**

As the price of copper increases, so does the possibility that your wire will be stolen. Using a wireless remote system means no wire for thieves to steal.

**Extended range**

Unlike much of the equipment on the market, Remote Control Technology’s wireless remote equipment has long-range communication capabilities — up to 5 miles.

**Eliminate the need for wire and conduit**

Wire and conduit are expensive and high maintenance. Typical wear-and-tear, digging, rodent damage, theft, etc., are all examples of problems that can damage wire. RCT’s wireless remote systems put an end to these drawbacks of wired technology.

**Higher profits**

Wireless remote switching systems eliminate the costly, labor-intensive process of trenching and laying wire. As a result, the contractor can enjoy an increased profitability of 200 percent or more in this facet of the job.

**No FCC licensing required**

RCT equipment does not require FCC licensing, whereas much of the other equipment on the market does. This is a significant benefit, as the FCC licensing process alone may take up to 8 weeks.

Less maintenance and servicing

In many states a contractor is obligated by law to maintain pumping systems for up to a year after its installation. RCT switching systems eliminate a majority of these maintenance and servicing issues by automating the job. Fewer service calls mean higher profits.

**Reliability and compatibility**

All of the components that a contractor puts into a project must interface with one another and have the utmost reliability. RCT wireless remote equipment has proven to be highly compatible with standard equipment used in most industries, as well as offering unparalleled reliability in use with programmable logic controllers (PLCs), various switches and relays, etc.

**Chapter 2**

**Theory**

* 1. **R307 Fingerprint Sensor**

Fingerprints are one of the many unique biometric signatures which we can use to identify people very accurately. But just by holding someone's hand and staring at their fingers can't be practical [grins]; we're not good at it. But computers are good at recognizing and matching patterns very fast and accurately. Before we can process a fingerprint pattern with a computer, we.must."Capture".it.

There exists many methods to digitize fingerprints; from forensic methods to ultrasound scanning. In this tutorial, we will learn how an Optical Fingerprint Scanner works and how we can interface the R307 fingerprint scanner module to Arduino. R307 is an optical fingerprint scanner module from R30X series produced by a Chinese vendor called Hangzhou Grow Technology Company Limited. Other sensors in the series are R300, R301T, R302, R303, R303T, R305, R306, R308, and R311, some of which are capacitive sensors. Despite having different sensing techniques and form-factors, they all share the same interface and command set. Therefore it is easy to adapt the library that you find here for other models as well.

**2.3 Technical Parameters**

• Supply voltage: DC 4.2 ~ 6.0V

• Supply current: Working current: 50mA (typical) Peak current: 80mA

• Interface: UART and USB

• Fingerprint image input time:

• Window area: 14x18 mm

• Matching method: Comparison method (1: 1)

• Search method (1: N)

• Characteristic file: 256 bytes

• Template file: 512 bytes

• Storage capacity: 1000 pieces

• Security Level: Five (from low to high: 1,2,3,4,5)

• Fake rate (FAR):

**2.4 Features**

• Perfect function: independent fingerprint collection, fingerprint registration, fingerprint comparison (1: 1) and fingerprint search (1: N) function.

• Small size: small size, no external DSP chip algorithm, has been integrated, easy to install, less fault.

• Ultra-low power consumption: low power consumption of the product as a whole, suitable for low-power requirements of the occasion.

• Anti-static ability: a strong anti-static ability, anti-static index reached 15KV above.

• Application development is simple: developers can provide control instructions, self-fingerprint application product development, without the need for professional knowledge of fingerprinting.

• Adjustable security level: suitable for different applications, security levels can be set by the user to adjust.

• Finger touch sensing signal output, low effective, sensing circuit standby current is very low, less than 5A.

**2.5 Interface Description**

The R307 fingerprint module has two interface TTL UART and USB2.0, USB2.0 interface can be connected to the computer; RS232 interface is a TTL level, the default baud rate is 57600 , can be changed, refer to a communication protocol ; can And microcontroller, such as ARM, DSP and other serial devices with a connection, 3.3V 5V microcontroller can be connected directly. Needs to connect the computer level conversion, level conversion note , embodiments such as a MAX232circuit.

**2.6 Working of fingerprint sensor and push buttons**

We use **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 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 proceed with selected ID. And Del key is used for reset or delete data from EEPROM of Arduino

**2.7 Pin configuration:**

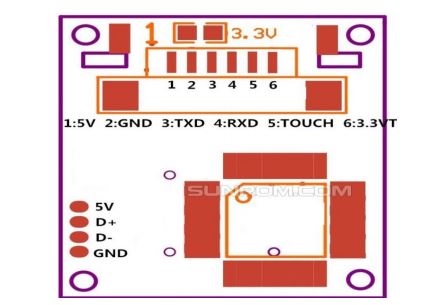
****

Figure 3: Pin Configuration

Table 02: PINOUTS:

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

The pin 6 (Touch Sense Power) is the supply voltage for the finger detection circuit. When a finger is present on the scanner, the output of pin 5 (Touch Sense) will be high. This signal can be used to initiate the scanning of the finger manually. Otherwise the scanner will wait for some time.to.detect.the.finger

The R307 has both USB and UART interfaces. With the USB, you can directly connect the scanner to a computer and communicate with it. A virtual COM port will be created when you connect the scanner to a Windows PC. If you want to interface the scanner with a microcontroller, you can use the UART interface which supports baud rates up to 115200 bps.

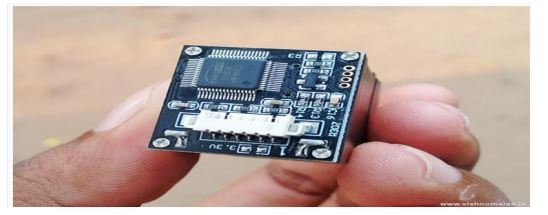


Figure 4: R307 Fingerprint Scanner PCB Side

The main controller on the PCB is AS606 from a company called Synochip. I don't know how Synochip is related to the company Hangzhou Grow. Whatever that is, the AS606 is a microcontroller with Cordis 5+ RISC cores and has everything needed for a performance controller.including.a DSP.

The R307 Manual is the only document to our rescue, and it is ambiguous at many places. The R30X Series Manual has some more information. The below schematic is included in the manual.

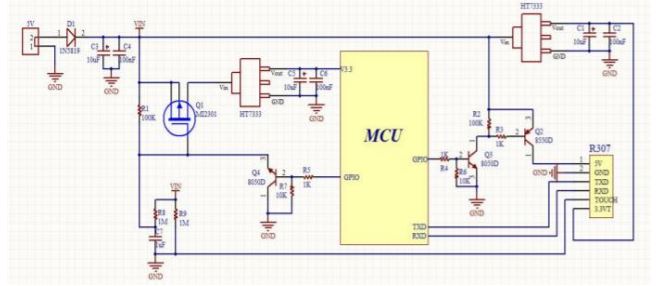


Figure 5:R307 Schematic Diagram.

**Chapter 3**

**Hardware Modeling**

**3.1 Prototype Modeling**

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.

**3.2 Main features of the prototype**

The features of the developed prototype are:

● LCD display (showing the username and time & date)

● Unto 256 fingerprints can be stored and checked when needed

● Fingerprint is stored in cloud digitally

● after fingerprint checking data is displayed in a serial monitor from Things board account

● Fingerprint data can be stored and deleted as many times as one wants

● Buzzer and LCD indicates fingerprint is stored and checked

● the date and time is shown when fingerprint is stored and checked along with username

● Cost Effective (Rs 2000/- approx.)

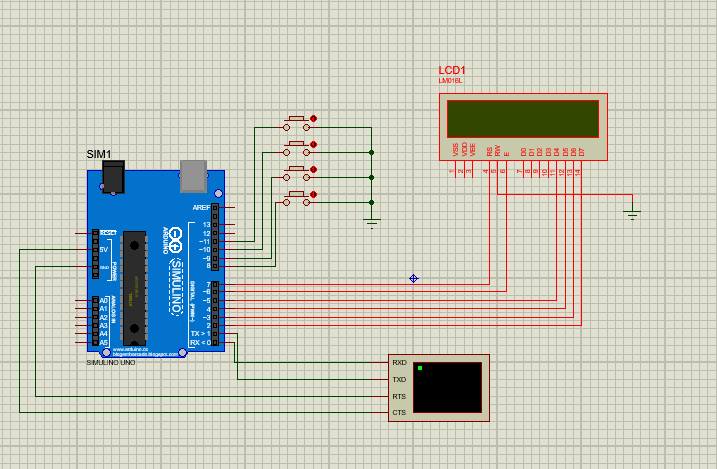
**3.3 Overview of the Project**

Working of this fingerprint attendance system project is fairly simple. First of all, the user needs to enrol 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 42 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 converts 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 needs 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.

* 1. **Hardware Connection**

**3.4.1 Prototype hardware connection:**



**Figure 6: Overall Circuit Diagram for Setup**

**HARDWARE IMPLEMENTATION:**

**• ARDUINO UNO:** Arduino UNO is used here to control the operations involved in taking attendance. The four operations that are to be performed are to enroll, verify, delete and reset. Arduino is chosen here as it is easy to use, code, handle and has many modules which add on features to Arduino board.

**• FINGERPRINT MODULE – R307:** R307 is an optical fingerprint scanner which is an upgraded version of R305. R307 has its own database which can store 1000 templates. Security level for R307 is from 1-5. This module has less false error rate, fast searching process, high speed processor, uses minutiae based algorithm to work with scanned fingerprints.

**• 16X2 LCD DISPLAY:** LCD Display is used here to provide messages to the user to have a better interaction with the device. LCD Display has green light in background with characters displayed on them in black. Characters are displayed in 7X5 matrix.

**• Wi-Fi Module:** ESP8266 Wi-Fi module is generally used to establish the wireless communication between the devices. But this module is not capable of 5-3V logic shifting and will require an external logic level converter

• Push Button.

• LED Lights.

• 1K Resistor.

• 2.2K resistor.

• Power.

• Connecting wires.

• Box.

• Buzzer.

• Bread Board.

• RTC Module. (DS3231)

**3.4.2 Detailed Hardware Description**

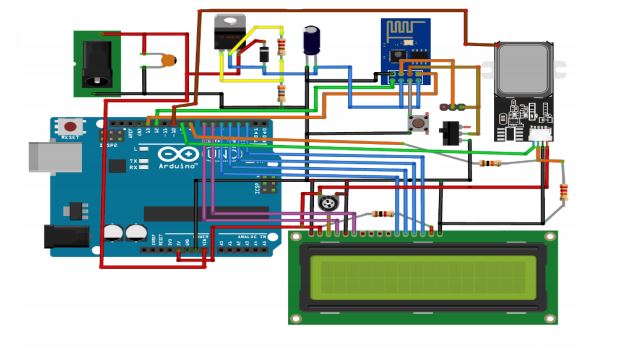


Figure 7: Hardware Setup Diagram

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. 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.

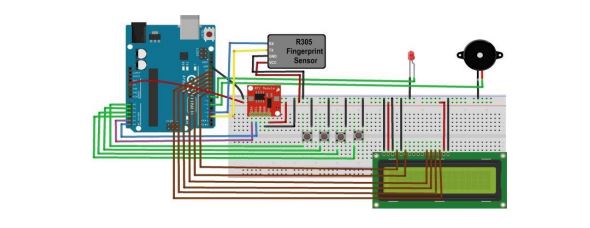


Figure 8: Circuit Diagram of the Setup

**SOFTWARE IMPLEMENTATION:**

1. **Adriano IDE :** You will be needing Arduino IDE software to write and upload the programming logic onto the Arduino Uno board
2. **Think board:** Also, you need to create an account in the Think boad IoT platform to integrate the system onto the cloud and store the data online. (WORKS AS A SERIAL MONITOR)
3. **ADAFRUIT FINGERPRINT SENSOR LIBRARY** is used for downloading the data to the serial monitor of the Arduino IDE and for refined capture of the fingerprints.

**Chapter 4**

**Logic & Operations**

**4.1 INTRODUCTION**

After assembling the system, what remains is to observe its operation and efficiency of the system. The total system is divided in several sub systems, like

● R 307 interfacing

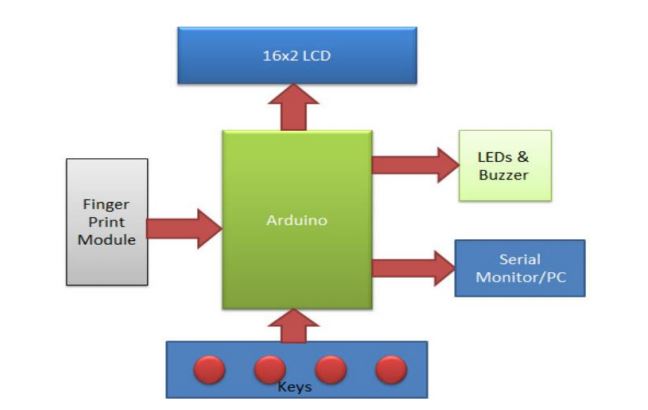
● Serial Monitor interfacing

● connect to Things board

● LCD interfacing

The operation of the whole circuit is depending on every sections performance

**4.2. Flow Diagram**

****

**Figure 9: Flowchart of the System**

**4.3 Principle & Operations**

Few years back if you were to tell someone that the Geyser and bedroom lights in your home are connected to internet, they would be baffled and might even criticize it as over-engineered products. But today with the advent of IoT, Smart cities etc the idea no longer sounds strange, we have devices around us that have become smarter by being able to communicate with the internet.

In this project our aim is to leverage this IoT into the boring attendance system to make it smart and more effective. Most conventional attendance systems available today store the information over a micro SD card and have to be connected to software via a computer to access the information. Here, we will build a biometric attendance system using Arduino that scans for finger print and on successful identification of the person it will log the information to a cloud platform like ThingsBoard by using the ESP8266 Wi-Fi module. This information can then be displayed in the dashboard of ThingsBoard making it available for the required authorities to view and analysis information over the internet without having any direct physical access to the hardware. However the conventional Attendance system without involving IoT can also be built by following the link and Finger print sensor can be further used for many other biometric applications like Voting Machine, Security system etc.

* 1. **Advantages of the project**

● Very accurate fingerprint reading & storing

● Cost Effective

● can be installed in small spaces

● Fingerprint is stored via cloud

● can store up to 1000 fingerprints

● LCD display for username, date and time of the operation.

● Alarm signal for attention of the observer.

**4.5 Disadvantages of the project**

● Limited number of fingerprints are stored

● As the whole thing is connected to internet so any problem related to internet can cause disruption to the whole system

● As fingerprints are stored primarily in EEPROM so that can create some problems

* 1. **Coding**

#include <LiquidCrystal.h>

#include <Adafruit\_Fingerprint.h>

#include <SoftwareSerial.h>

#include <EEPROM.h>

const int buttonPin8 = 8; // the number of the pushbutton pin

const int buttonPin9 = 9; // the number of the pushbutton pin

const int buttonPin10 = 10; // the number of the pushbutton pin

const int buttonPin11 = 11; // the number of the pushbutton pin

//const int ledPin = 13; // the number of the LED pin

LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // initialize the library with the numbers of the interface pins

int buttonState8 = 0;

int buttonState9 = 0;

int buttonState10 = 0;

int buttonState11 = 0;

uint8\_t id =1;

uint8\_t id\_used =0;

int key= 0;

uint8\_t getFingerprintEnroll();

//SoftwareSerial mySerial(11, 12);

Adafruit\_Fingerprint finger = Adafruit\_Fingerprint(&Serial);

void setup() {

pinMode(buttonPin8, INPUT\_PULLUP);

pinMode(buttonPin9, INPUT\_PULLUP);

pinMode(buttonPin10, INPUT\_PULLUP);

pinMode(buttonPin11, INPUT\_PULLUP);

lcd.begin(16, 2);

lcd.print(" FINGERPRINT");

lcd.setCursor(0, 1);

lcd.print("ATTENDANCE DEMO");

delay(500);

finger.begin(57600);

/\* if (!finger.verifyPassword()) {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Did not find ");

lcd.setCursor(0, 1);

lcd.print("fingerprint sensor :(");

while (1);

} \*/

}

void loop() {

showmenu();

while((key = getkey()) == 0);

if (key == 1) {

id = getid();

if(id == 0)

{

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("No space");

lcd.setCursor(0, 1);

lcd.print("Del templetes");

delay(1000);

}else{

getFingerprintEnroll();

}

delay(1500);

} else if (key == 2) {

getFingerprintID();

delay(1500);

} else if (key == 3) {

showDelmenu();

delay(1500);

} else if (key == 4) {

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Credits:");

delay(500);

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("Razaul Haque");

lcd.setCursor(0, 1);

lcd.print("Id-17182203035");

delay(1000);

}else {

//lcd.clear();

}

}

void showmenu()

{

lcd.clear();

lcd.setCursor(0, 0);

lcd.print("1:Enroll 2:Scan ");

lcd.setCursor(0, 1);

lcd.print("3:Del ");

}

void showDelmenu()

{

lcd.clear();

lcd.setCursor(0, 0);

id\_used = getUsedIdnext();

lcd.print("ID <");lcd.print(id\_used);lcd.print(">");

lcd.print(" 1:Del");

lcd.setCursor(0, 1);

lcd.print("2:Up 3:Dn 4:Back");

while(getkey()!=4)

{

if(getkey()==2)

{

while(getkey()==2);

id\_used = getUsedIdnext();

lcd.setCursor(0, 0);

lcd.print("ID <");lcd.print(id\_used);lcd.print("> 1-Del");

}else if(getkey()==3){

while(getkey()==3);

id\_used = getUsedIdprev();

lcd.setCursor(0, 0);

lcd.print("ID <");lcd.print(id\_used);lcd.print("> 1-Del");

}else if(getkey()==1){

while(getkey()==1);

deleteFingerprint(id\_used);

unreserveId(id\_used);

id\_used = getUsedIdprev();

lcd.setCursor(0, 0);

lcd.print("ID <");lcd.print(id\_used);lcd.print("> 1-Del");

}

}

}

int getkey()

{

if( digitalRead(buttonPin8)){

while( digitalRead(buttonPin8));

return 1;

}else if(digitalRead(buttonPin9)){

while( digitalRead(buttonPin9));

return 2;

}else if(digitalRead(buttonPin10)){

while( digitalRead(buttonPin10));

return 3;

}else if(digitalRead(buttonPin11)){

while( digitalRead(buttonPin11));

return 4;

}else{return 0;}

}

int getid(){

for(id = 1 ;id <=160;id++)

{

if(EEPROM.read(id)==0)

{

return id;

}

}

if(id>160)

{

return 0;

}

}

int getUsedIdnext()

{

for(;id\_used<=160;)

{

id\_used++;

if(EEPROM.read(id\_used)==1)

{

return id\_used;

}

}

if(id\_used>160)

{

return 0;

}

return id\_used;

}

int getUsedIdprev()

{

if(id\_used>1)

for(;id\_used>0;)

{

id\_used--;

if(EEPROM.read(id\_used)==1)

{

return id\_used;

}

}

if(id\_used>160)

{

return 0;

}

return id\_used;

}

void reserveId(int id){

EEPROM.write(id,1);

}

void unreserveId(int id){

EEPROM.write(id,0);

}

uint8\_t getFingerprintEnroll() {

int p = -1;

lcd.print("Waiting for finger");

while (p != FINGERPRINT\_OK) {

p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

lcd.clear();

lcd.print("Image taken");

break;

case FINGERPRINT\_NOFINGER:

lcd.clear();

lcd.print(".");

break;

case FINGERPRINT\_PACKETRECIEVEERR:

lcd.clear();

lcd.print("Communication error");

break;

case FINGERPRINT\_IMAGEFAIL:

lcd.clear();

lcd.print("Imaging error");

break;

default:

lcd.clear();

lcd.print("Unknown error");

break;

}

}

// OK success!

p = finger.image2Tz(1);

switch (p) {

case FINGERPRINT\_OK:

lcd.clear();

lcd.print("Image converted");

break;

case FINGERPRINT\_IMAGEMESS:

lcd.clear();

lcd.print("Image too messy");

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

lcd.clear();

lcd.print("Communication error");

return p;

case FINGERPRINT\_FEATUREFAIL:

lcd.clear();

lcd.print("Could not find fingerprint features");

return p;

case FINGERPRINT\_INVALIDIMAGE:

lcd.clear();

lcd.print("Could not find fingerprint features");

return p;

default:

lcd.clear();

lcd.print("Unknown error");

return p;

}

lcd.clear();

lcd.print("Remove finger");

delay(2000);

p = 0;

while (p != FINGERPRINT\_NOFINGER) {

p = finger.getImage();

}

lcd.clear();

lcd.print("ID ");lcd.print(id);

p = -1;

lcd.clear();

lcd.print("Place same finger again");

while (p != FINGERPRINT\_OK) {

p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

lcd.clear();

lcd.print("Image taken");

break;

case FINGERPRINT\_NOFINGER:

lcd.clear();

lcd.print(".");

break;

case FINGERPRINT\_PACKETRECIEVEERR:

lcd.clear();

lcd.print("Communication error");

break;

case FINGERPRINT\_IMAGEFAIL:

lcd.clear();

lcd.print("Imaging error");

break;

default:

lcd.clear();

lcd.print("Unknown error");

break;

}

}

// OK success!

p = finger.image2Tz(2);

switch (p) {

case FINGERPRINT\_OK:

lcd.clear();

lcd.print("Image converted");

break;

case FINGERPRINT\_IMAGEMESS:

lcd.clear();

lcd.print("Image too messy");

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

lcd.clear();

lcd.print("Communication error");

return p;

case FINGERPRINT\_FEATUREFAIL:

lcd.clear();

lcd.print("Could not find fingerprint features");

return p;

case FINGERPRINT\_INVALIDIMAGE:

lcd.clear();

lcd.print("Could not find fingerprint features");

return p;

default:

lcd.clear();

lcd.print("Unknown error");

return p;

}

// OK converted!

lcd.clear();

lcd.print("Creating model #");lcd.print(id);

p = finger.createModel();

if (p == FINGERPRINT\_OK) {

lcd.clear();

lcd.print("Prints matched!");

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

lcd.clear();

lcd.print("Communication error");

return p;

} else if (p == FINGERPRINT\_ENROLLMISMATCH) {

lcd.clear();

lcd.print("Fingerprints did not match");

return p;

} else {

lcd.clear();

lcd.print("Unknown error");

return p;

}

lcd.clear();

lcd.print("ID ");lcd.print(id);

p = finger.storeModel(id);

if (p == FINGERPRINT\_OK) {

lcd.clear();

lcd.print("Stored at id ");lcd.print(id);

reserveId(id);

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

lcd.clear();

lcd.print("Communication error");

return p;

} else if (p == FINGERPRINT\_BADLOCATION) {

lcd.clear();

lcd.print("Could not store in that location");

return p;

} else if (p == FINGERPRINT\_FLASHERR) {

lcd.clear();

lcd.print("Error writing to flash");

return p;

} else {

lcd.clear();

lcd.print("Unknown error");

return p;

}

}

uint8\_t getFingerprintID() {

uint8\_t p = 0;

lcd.clear();

lcd.print("Place Finger");

delay(1500);

p = finger.getImage();

switch (p) {

case FINGERPRINT\_OK:

lcd.clear();

lcd.print("Image taken");

break;

case FINGERPRINT\_NOFINGER:

lcd.clear();

lcd.print("No finger detected");

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

lcd.clear();

lcd.print("Communication error");

return p;

case FINGERPRINT\_IMAGEFAIL:

lcd.clear();

lcd.print("Imaging error");

return p;

default:

lcd.clear();

lcd.print("Unknown error");

return p;

}

// OK success!

p = finger.image2Tz();

switch (p) {

case FINGERPRINT\_OK:

lcd.clear();

lcd.print("Image converted");

break;

case FINGERPRINT\_IMAGEMESS:

lcd.clear();

lcd.print("Image too messy");

return p;

case FINGERPRINT\_PACKETRECIEVEERR:

lcd.clear();

lcd.print("Communication error");

return p;

case FINGERPRINT\_FEATUREFAIL:

lcd.clear();

lcd.print("Could not find fingerprint features");

return p;

case FINGERPRINT\_INVALIDIMAGE:

lcd.clear();

lcd.print("Could not find fingerprint features");

return p;

default:

lcd.clear();

lcd.print("Unknown error");

return p;

}

// OK converted!

p = finger.fingerFastSearch();

if (p == FINGERPRINT\_OK) {

lcd.clear();

lcd.print("Found a print match!");

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

lcd.clear();

lcd.print("Communication error");

return p;

} else if (p == FINGERPRINT\_NOTFOUND) {

lcd.clear();

lcd.print("Did not find a match");

return p;

} else {

lcd.clear();

lcd.print("Unknown error");

return p;

}

// found a match!

lcd.clear();

lcd.print("Found ID #"); lcd.print(finger.fingerID);

}

uint8\_t deleteFingerprint(uint8\_t id) {

uint8\_t p = -1;

p = finger.deleteModel(id);

if (p == FINGERPRINT\_OK) {

Serial.println("Deleted!");

} else if (p == FINGERPRINT\_PACKETRECIEVEERR) {

Serial.println("Communication error");

return p;

} else if (p == FINGERPRINT\_BADLOCATION) {

Serial.println("Could not delete in that location");

return p;

} else if (p == FINGERPRINT\_FLASHERR) {

Serial.println("Error writing to flash");

return p;

} else {

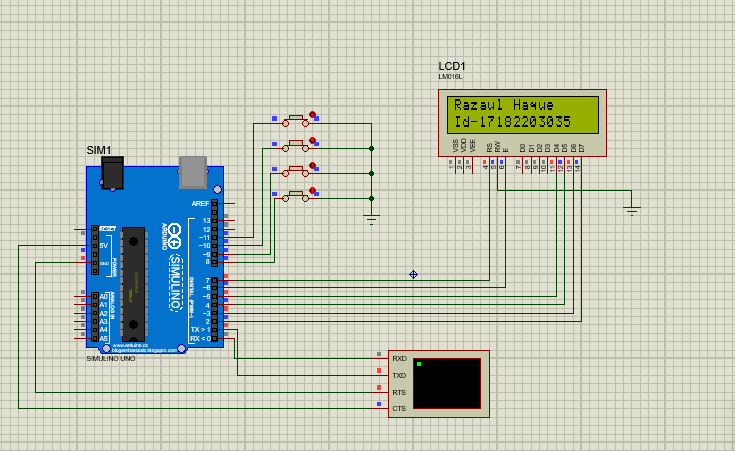
Serial.print("Unknown error: 0x"); Serial.println(p, HEX);

return p;

}

}

**4.7 Circuit Diagram**

****

**Figure 11: Circuit Diagram of Setup**

**CHAPTER-5**

**CONCLUSIONS**

**5.1 CONCLUSION**

Here we have developed a Biometric fingerprint based attendance system using Arduino.

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.

**5.2 RESULTS**

The experimental model was made following the circuit diagram and the desired results were obtained. Every time someone places his finger on the sensor the sensor reads the data and stores it in the cloud. Next time someone wants to check the fingerprint he/she places the finger on the sensor. The sensor reads the data and searches and cross-checks the data with stored fingerprints. If it matches with any of them then it displays the username, date and time. If not then says fingerprint doesn’t match .That’s how the whole system works.

**REFERENCES**

• https://ijarcce.com/wp-content/uploads/2016/11/IJARCCE-ICRITCSA-3.pdf

• www.irjet.net/uploads/2016/05/ IRJET-V3I5237.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.