

Ahsanullah University of Science & Technology

Department of Computer Science & Engineering



CSE 3216

Microcontroller Based System Design Lab

Project Title : Fingerprint Student Attendance System

Submission Date : 15/04/2019

Submitted By:

A.H.M Annur	16.01.04.087
Sumaiya Rahman	16.01.04.088
Mohimenul Islam Razib	16.01.04.096
Abir Hossain Xian	16.01.04.100

Introduction: In our country every school, collage, university, corporate office has an attendance calling system. For now, attendance calling for school, collage, university takes around 10-15 mins. Our system is a automated attendance system with finger print input system that will make the attendance process lot more faster and easier in our county. We are looking forward to make a industrial level device that can be used anywhere in any environment.

Components:

Hardware Components:

- **Arduino UNO REV3** – used as a microcontroller.
- **20x4 LCD** – used to see all data and manage stuffs
- **R305 Fingerprint module** – use to input fingerprints
- **ESP8266 Wi-Fi Module** – gives microcontroller access to Wi-Fi network. In our project it is used to connect with student's mobiles and capture MAC address generated in that device.
- **Mini circuit Boards** – use to solve the common connections / extra connection between Arduino & other hardware components.
- **Male/Female Jumper Wires** – use to give connection between two points/ports.
- **Resistances** – use with the LCD.
- **Keypad** – use to make the controller

Software Components:

- Arduino IDE – for burning code on Arduino UNO REV3.
- Proteus –to design the hardware component model.

Features:

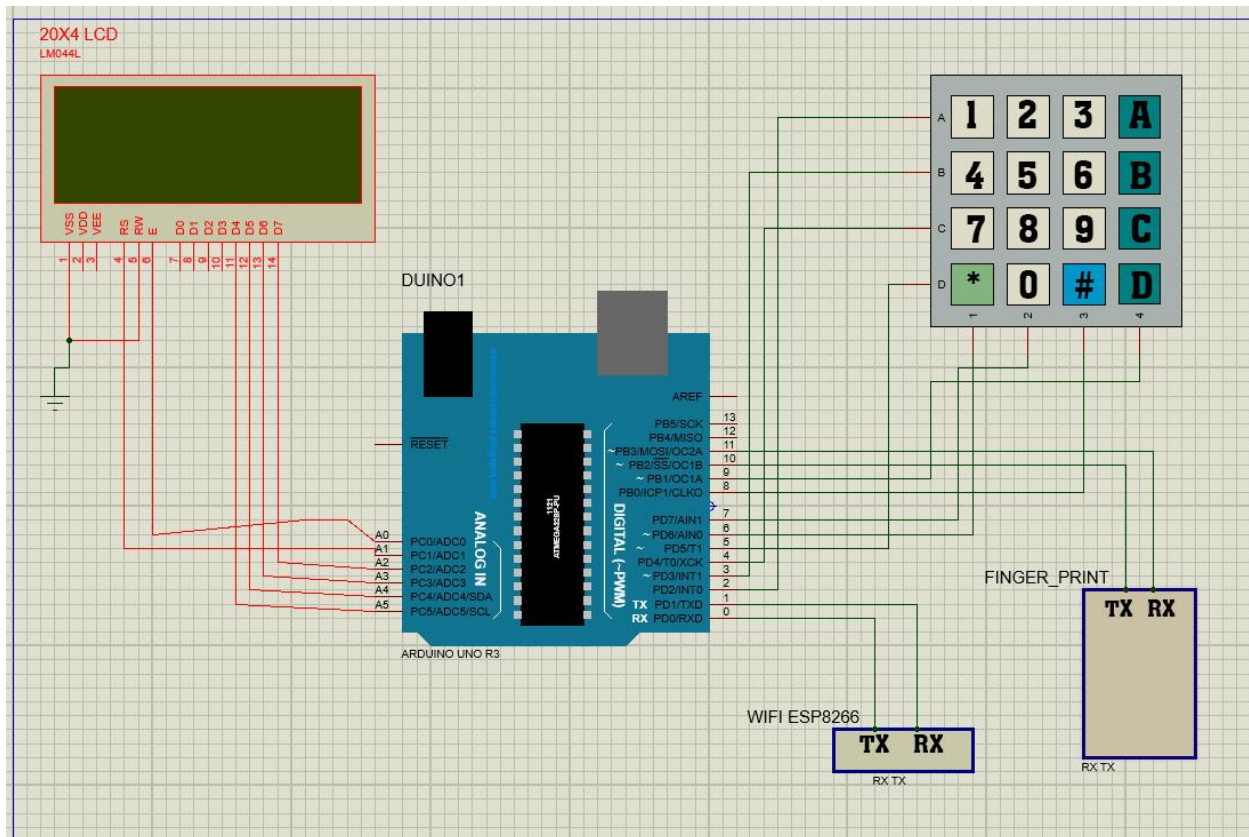
- The device will be portable.
- Visualization of student attendance with the LCD screen
- Basically, initially the device will get fingerprint from the students for registration.
- Then every time a student enters a class, he/she will put his/her finger to give the attendance.
- Our device will receive the fingerprint and match it with the registered fingerprints and give attendance of that class to the student.
- A teacher is the main operator of this device.
- Teacher will start the attendance count by giving his course id after that student can enter their finger to get the attendance in a very fast way.
- Our device is anti-proxy system, there is near zero percent chance that in a class 2 people will have the same finger print. So, no one can give any proxy.
- Our UI will be very simple.

Working Principle:

- First there will be a main menu. Options are given below-
 - Take Attendance
 - Teacher will enter course number to start the attendance system
 - Attendance system will auto magically wait for each student's finger
 - After getting attendance teacher will stop the system
 - Add Students
 - For adding student, first need to enter Student id then system will ask for their finger.
 - After giving the fingerprint system will register the student

- See Attendance
 - Here Teacher can have a short look on the presented ID's
 - Also, teacher can clear the attendance if he wants to.
- Send data to server
 - Here System will send all the presented data to the server.
 - Teacher can see the data from the server after this step

Circuit Diagram:



Figures of the project:



Constrains:

Connecting LCD with the Arduino was little hard.
Also operating WiFi module with command needed a lot of study about it.

Dos and Don'ts:

Key Features	Does	Doesn't
LCD Display	20x4 LCD	Graphic LCD
Push Button		Failed to add
KeyPad	Added successfully	
Fingerprint module	Added successfully	
Wi-Fi Module	Added successfully	

Conclusion: In the end we can say that it's a prototype of a very large project. If we can pull it off by a cheaper cost it will be very helpful to all of our educational institution of our country.

Appendix:

Arduino Code:

```
#include <Wire.h>
#include <LiquidCrystal.h>
#include <Keypad.h>
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
```

```

int displayMode = 0;
int tempID = 0;
String courseNo = "";
String ATTEND_STR = "ID : ";

//WIFI
#define wifi Serial
String wifiSSID = "MSD_STUDENT";
String wifiPASS = "123456789";
String HOST = "192.168.0.105";
String PORT = "5000";
String rxd, getData;
bool connectFlag = false;

//finger print
SoftwareSerial mySerial(10, 11); //RX,TX
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
int id ,getId;
bool fig = false;

//LiquidCrystal lcd(RS, E, D4, D5, D6, D7)
LiquidCrystal lcd(A1, A0, A5, A4, A3, A2);

//KEYPAD
const byte rows = 4;
const byte cols = 4;

char keys[rows][cols] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

byte rowPin[] = {2,3,4,5};
byte colPins[] = {6,7,8,9};

Keypad kpd = Keypad(makeKeymap(keys),rowPin, colPins, rows, cols);

```

```

void setup()
{

    //WiFi
    wifi.begin(9600);

    //LCD
    lcd.begin(20, 4);
    //lcd.print("XIAN");

    //FINGER PRINT
    finger.begin(57600);

    if(finger.verifyPassword()){
        //Serial.println("Found fingerprint sensor!");
    }else{
        //Serial.println("Did not find fingerprint sensor :(");
    }

}

void loop()
{
    //while(connectFlag == false) wifi_init();
    //if(figer) getId = matchFinger();

    if(displayMode == 0){
        printMainMenu();
    }else if(displayMode == 1){
        printAddStudent();
    }else if(displayMode == 2){
        printTakeAttend();
    }else if(displayMode == 3){
        printTakingAttendance();
    }else if(displayMode == 4){
        printAttendedStudents();
    }else if(displayMode == 5){
        sendDataToServer();
    }
}

```



```
}
```

```
void printAttendedStudents(){  
    lcd.setCursor(0,0);  
    lcd.print(ATTEND_STR);  
    lcd.setCursor(0,3);  
    lcd.print("B: BACK  C: CLEAR");  
  
    char x = kpd.getKey();  
    if(x){  
        if(x == 'B'){  
            displayMode = 0;  
            lcd.clear();  
        }else if(x == 'C'){  
            ATTEND_STR = (String)"ID : ";  
            lcd.clear();  
            lcd.setCursor(0,2);  
            lcd.print("ATTENDANCE CLEARED!");  
            sendData("CLEAR");  
            delay(1500);  
            displayMode = 0;  
            lcd.clear();  
        }  
    }  
}
```

```
void sendDataToServer(){  
    lcd.setCursor(0,1);  
    lcd.print("SENDING DATA.");  
    delay(200);  
    lcd.print(".");  
    delay(200);  
    lcd.print(".");  
    delay(200);  
    lcd.print(".");  
    delay(200);  
    lcd.print(".");  
    delay(200);  
    lcd.print(".");  
    delay(200);  
    sendData("PRINT");  
}
```

```

delay(500);
lcd.setCursor(0,3);
lcd.print("<<<<<<<DONE>>>>>>>");
delay(1500);
displayMode = 0;
lcd.clear();

}

void printMainMenu(){
  //lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("A: TAKE ATTENDANCE");
  lcd.setCursor(0,1);
  lcd.print("B: ADD STUDENT");
  lcd.setCursor(0,2);
  lcd.print("C: SEE ATTENDANCE");
  lcd.setCursor(0,3);
  lcd.print("D: SEND DATA");

  char x = kpd.getKey();
  if(x){
    if(x == 'A'){

      displayMode = 2;
      lcd.clear();

    }else if(x == 'B'){

      displayMode = 1;
      lcd.clear();

    }else if(x == 'C'){

      displayMode = 4; //See Attended Students
      lcd.clear();

    }else if(x == 'D'){
      displayMode = 5; //Send Data to server
      lcd.clear();
    }
  }
}

```

```

void printTakeAttend(){
    lcd.setCursor(0,0);
    lcd.print("ENTER COURSE : ");
    //lcd.setCursor(0,10);

    lcd.setCursor(0,3);
    lcd.print("A: DONE");
    lcd.setCursor(11,3);
    lcd.print("B: CANCEL");
    lcd.setCursor(0,1);

    char x = kpd.getKey();
    if(x){
        if(x == 'A'){

            displayMode = 3; //Start Attendance
            lcd.clear();

        }else if(x == 'B'){
            displayMode = 0;
            lcd.clear();
        }else{
            courseNo += x;
            lcd.print(courseNo);
        }

    }

}

void printTakingAttendance(){
    lcd.setCursor(0,0);
    lcd.print("ATTENDANCE STARTED!!");
    lcd.setCursor(0,1);
    lcd.print("ENTER YOUR FINGER");
    lcd.setCursor(0,3);
    lcd.print("A: STOP");

    char x = kpd.getKey();
    if(x){
        if(x == 'A'){
            lcd.clear();
            lcd.setCursor(0,2);

```

```

    lcd.print("ATTENDANCE FINISHED");
    courseNo = "";
    delay(1000);
    displayMode = 0;
    lcd.clear();
}
}else{
    getId = matchFinger();

    if(getId != -1){
        lcd.setCursor(0,2);
        lcd.print("      ");
        lcd.setCursor(0,2);
        lcd.print("PRESENT ID : "+(String)getId);
        sendData(courseNo+"",(String)getId);
        ATTEND_STR += " , "+(String)getId;
        delay(2000);

        lcd.setCursor(0,2);
        lcd.print("      ");
        lcd.setCursor(0,2);
        lcd.print("NEXT STUDENT");

    }else{

    }
    delay(250);
}

}

void printAddStudent(){
    lcd.setCursor(0,0);
    lcd.print("Enter ID : ");
    //lcd.setCursor(0,10);

    lcd.setCursor(0,3);
    lcd.print("A: ADD");
    lcd.setCursor(11,3);
    lcd.print("B: CANCEL");
    lcd.setCursor(11,0);

    char x = kpd.getKey();

```

```

if(x){
    if(x == 'A'){
        id = tempID;
        tempID = 0;
        lcd.setCursor(0,2);
        lcd.print("ADDING ID : "+(String)id);
        delay(1000);
        lcd.clear();

        addStudent();

    }else if(x == 'B'){
        displayMode = 0;
        lcd.clear();
    }else{
        tempID *= 10;
        tempID += (int)x -48;

        lcd.print(tempID);
    }
}

}

}

void addStudent(){
    lcd.setCursor(0,0);
    lcd.print("ENTER FINGER FOR");
    lcd.setCursor(0,1);
    lcd.print("ID = "+(String)id);
    bool enroll = fingerEnroll(id);
    if(enroll == true){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("STUDENT REGISTERED!");
        lcd.setCursor(0,2);
        lcd.print("STUDENT ID = "+(String)id);
        delay(1500);
        displayMode = 0;
        lcd.clear();
    }
}
}

```

```

bool fingerEnroll(int id) {
    int p = -1;
    //Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
    }

    p = finger.image2Tz(1);
    if (p != FINGERPRINT_OK) return false;

    p = -1;
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
    }

    p = finger.image2Tz(2);
    if (p != FINGERPRINT_OK) return false;

    //Serial.print("Creating model for #"); Serial.println(id);

    p = finger.createModel();
    if (p != FINGERPRINT_OK) return false;

    //Serial.print("ID "); Serial.println(id);
    p = finger.storeModel(id);
    if (p != FINGERPRINT_OK) return false;
    return true;
}

int matchFinger(){
    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) return -1;

    //Serial.print("Found ID #"); Serial.println(finger.fingerID);
    //lcd.print("#"+finger.fingerID);

```

```

    return finger.fingerID;
}

void wifi_init(){
    //----- configuration
    digitalWrite(LED_BUILTIN, HIGH);
    //wifi.listen();
    wifi.println((String)"AT+CWJAP=\"" + wifiSSID + "\",\"" + wifiPASS + "\"");
    delay(100);
    if(wifi.available() > 0){
        rxd = wifi.readString();
        if(rxd.indexOf("GOT IP") != -1) {
            connectFlag = true;
            wifi.println((String)"AT+CIPSTART=\"TCP\",\"" + HOST + "\",\"+PORT +\"");
            delay(50);
            digitalWrite(LED_BUILTIN, LOW);
        }
        rxd = "";
    }
}

void sendData(String msg){
    digitalWrite(LED_BUILTIN, HIGH);
    int digit = msg.length() + 2;

    wifi.println((String)"AT+CIPSTART=\"TCP\",\"" + HOST + "\",\"+PORT +\"");
    delay(50);
    wifi.println((String)"AT+CIPSEND=" + digit);
    delay(50);
    wifi.println(msg);
    delay(50);
    digitalWrite(LED_BUILTIN, LOW);
}

```

Server-Side Code:

```

ArrayList<String> STUDENTS = new ArrayList();

    try {
        while(true){
            ss = new ServerSocket(5000);
            s = ss.accept();

```

```

//System.out.println("Server is running");

isr = new InputStreamReader(s.getInputStream());
br = new BufferedReader(isr);
message = br.readLine();

//System.out.println(message);

if(message.equals("PRINT")){
    System.out.println("PRESENT DATA: "+STUDENTS.size());
    for(int i=0;i<STUDENTS.size();i++){
        System.out.println(STUDENTS.get(i));
    }
}else if(message.equals("CLEAR")){
    STUDENTS.clear();
    System.out.println("PRESENT CLEARED:");

}else{
    String[] dataArray = message.split(",");
    STUDENTS.add("STUDENT ID\t"+dataArray[1]+" \tPRESENT ON
COURSE\t"+dataArray[0]);

}

isr.close();
br.close();
ss.close();
s.close();

}
} catch (IOException ex) {
    Logger.getLogger(MyServer.class.getName()).log(Level.SEVERE, null, ex);
}

```