FINGERPRINT BASED ATTENDANCE SYSTEM

Adithya S M

S4 : ECE Alpha

Department of Electronics and Communication Engineering
Rajagiri School of Engineering and Technology

RSET
RAJAGIRI SCHOOL OF
ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

October 10, 2021

Abstract

To design and implement a figerprint based attendence system using Arduino and figerprint system.The device uses different modules such as arduino,Adafruit optical fingerprint sensor,LCD module.This system would be very promising and as the world is moving closely towards automation and digitization this system can have immense potential to pull of a large market share.

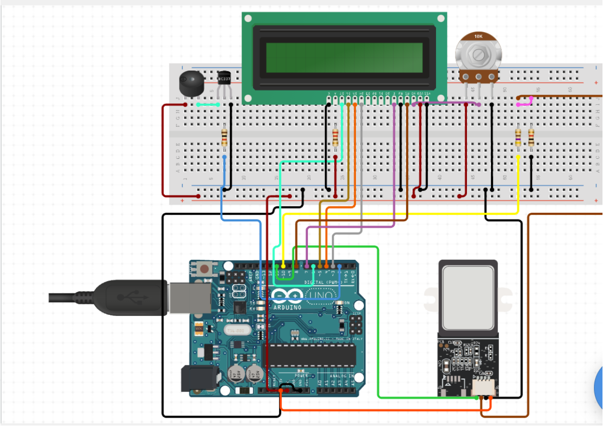
Contents

List of Figures

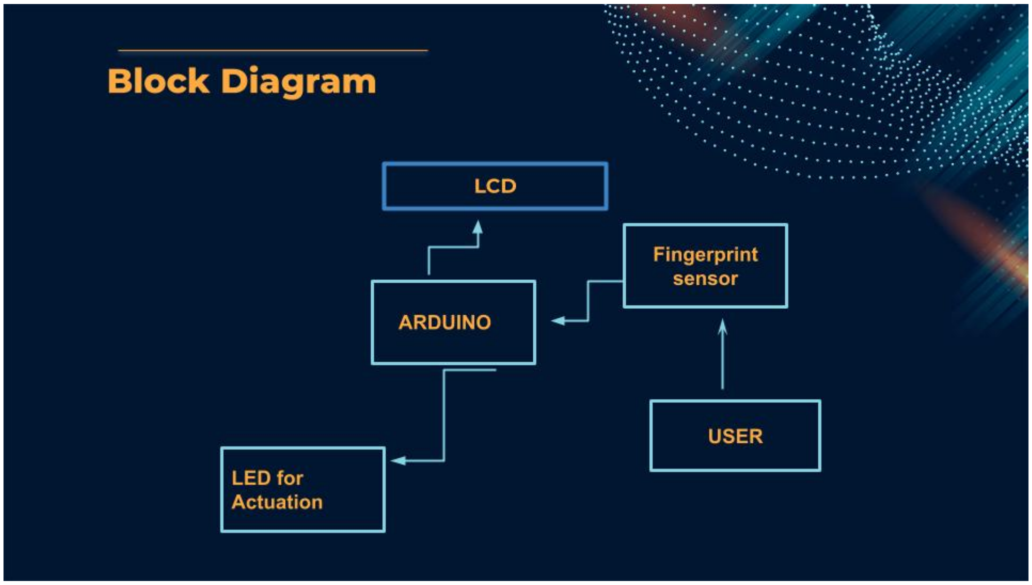
Chapter 1

Introduction

Fingerprint attendance system is a revolutionary method to modernize the existing roll calls. This devices enables customers and users to go paperfree and decreases the time consumption. The device is more feasible and the form factor helps to overcome its present competetors in the market.



1.1 Block Diagram



[1].

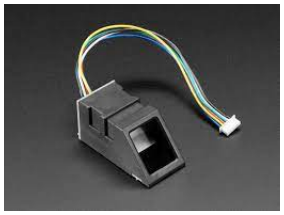
1.1.1 Arduino

Arduino is an open-source hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices.The MCU is a ATMEGA328P based Development board Known for its Versatility and simplicity. itemize

- Microcontroller: ATmega328P.
- Operating Voltage: 5V.
- Input Voltage (recommended): 7-12V.
- Inout Voltage (limit): 6-20V.
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins: 6.
- Analog Input Pins: 6.
- DC Current per I/O Pin: 20 mA.

Figure 1.1: Arduino

1.1.2 Adafruit Fingerprint Sensor



FingerPrint Sensor

The Adafruit optical fingerprint sensor is a popular sensor compatible with MCU such as Arduino,Raspberry pi etc.The Sensor works in the same range of baud rate as arduino itemize

- Supply voltage: 3.6 - 6.0VDC.
- Operating current: 120mA max.
- Peak current: 150mA max.
- Fingerprint imaging time: ¡1.0 seconds.
- Window area: 14mm x 18mm.
- Signature file: 256 bytes.
- Template file: 512 bytes.
- Storage capacity: 162 templates.

1.1.3 Liquid Crystal Display

LCD 16x2 is a 16-pin device that has 2 rows that can accommodate 16 characters each. LCD 16x2 can be used in 4-bit mode or 8-bit mode. It is also possible to create custom characters. It has 8 data lines and 3 control lines that can be used for control purposes. itemize

- Operating voltage :5 V
- Screen resolution :2-lines x 16 characters
- Character resolution :5 x 8 pixels
- Module dimensions :80 x 36 x 12 mm
- Viewing area dimension :64.5 x 16.4 mm

1.2 Arduino Compatible C

The Programming language used here is C which is compatible in c.The Libraries provided simplifies in programs. [3].

Chapter 2

Code and expalnation

The program is done on Arduino Ide.The project utilises the assistance from different inbuilt libraries they are: itemize

- include¡EEPROM.h¿
- include¡LiquidCrystal.h¿
- LiquidCrystal lcd(13,12,11,10,9,8);
- include ¡SoftwareSerial.h¿
- include ¡Wire.h¿
- include ”RTClib.h”

- include ”Adafruit$_{F}ingerprint.h$” The program works on different modes like accepting new entries,deletion,checking the attendances etc. all actions are conveyed through the LCD monitor.
Sketch uses 283297 bytes (27Global variables use 30108 bytes (36

2.0.1 Code used

listings [style=CStyle] include¡EEPROM.h¿ include¡LiquidCrystal.h¿ LiquidCrystal lcd(13,12,11,10,9,8); include ¡SoftwareSerial.h¿ SoftwareSerial fingerPrint(2, 3);
include ¡Wire.h¿ include ”RTClib.h” RTC$_{D}S1307rtc$;
include ”Adafruit$_{F}ingerprint.h” uint8_{t}id$;
Adafruit$_{F}ingerprintfinger = Adafruit_{F}ingerprint(fingerPrint)$;
define enroll 14
define del 15
define up 16
define down 17
define match 5
define indFinger 7
define buzzer 5
define records 4 // 5 for 5 user

```
int user1,user2,user3,user4,user5;
DateTime now;
void setup()
delay(1000);
lcd.begin(16,2);
Serial.begin(9600);
pinMode(enroll, INPUT_PULLUP);
pinMode(up, INPUT_PULLUP);
pinMode(down, INPUT_PULLUP);
pinMode(del, INPUT_PULLUP);
pinMode(match, INPUT_PULLUP);
pinMode(buzzer, OUTPUT);
pinMode(indFinger, OUTPUT);
digitalWrite(buzzer, LOW);
if(digitalRead(enroll) == 0)
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
lcd.clear();
lcd.print(”Please wait”);
lcd.setCursor(0,1);
lcd.print(”Downloding Data”);
Serial.println(”Please wait”);
Serial.println(”Downloding Data..”);
Serial.println();
Serial.print(”S.No. ”);
for(int i=0;i¡records;i++)
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW); Serial.print(” User ID”); Serial.print(i+1);
Serial.print(” ”); Serial.println(); int eepIndex=0; for(int i=0;i¡30;i++)
if(i+1¡10) Serial.print(’0’); Serial.print(i+1); Serial.print(” ”); eepIndex=(i*7);
download(eepIndex); eepIndex=(i*7)+210; download(eepIndex);
eepIndex=(i*7)+420; download(eepIndex); eepIndex=(i*7)+630;
download(eepIndex); // eepIndex=(i*7)+840; // 5th user //
download(eepIndex); Serial.println(); if(digitalRead(del) == 0) lcd.clear();
lcd.print(”Please Wait”); lcd.setCursor(0,1); lcd.print(”Reseting.....”); for(int
i=1000;i¡1005;i++) EEPROM.write(i,0); for(int i=0;i¡841;i++)
EEPROM.write(i, 0xff); lcd.clear(); lcd.print(”System Reset”); delay(1000);
lcd.clear(); lcd.print(” Attendance ”); lcd.setCursor(0,1); lcd.print(” System
”); delay(2000); lcd.clear(); lcd.print(”CAM project”); lcd.setCursor(0,1);
lcd.print(”Adithya S M”); delay(2000); digitalWrite(buzzer, HIGH);
delay(500); digitalWrite(buzzer, LOW); for(int i=1000;i¡1000+records;i++)
if(EEPROM.read(i) == 0xff) EEPROM.write(i,0);
finger.begin(57600); Serial.begin(9600); lcd.clear(); lcd.print(”Finding
Module”); lcd.setCursor(0,1); delay(1000); if (finger.verifyPassword())
```

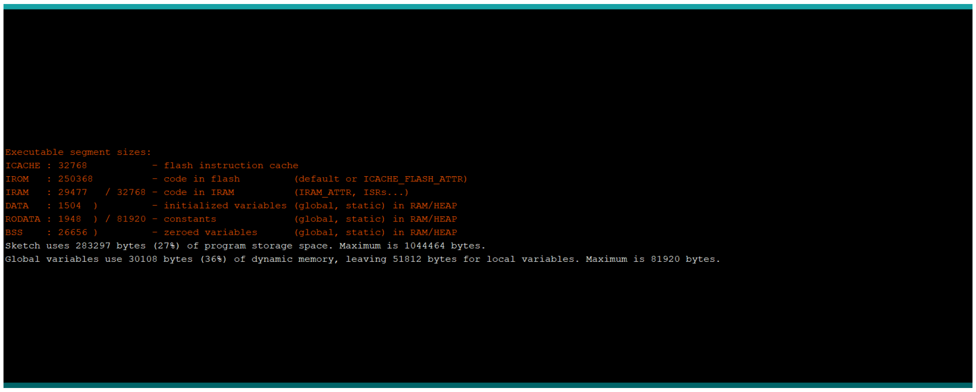
Serial.println("Found fingerprint sensor!"); lcd.clear(); lcd.print("Found
Module "); delay(1000); else Serial.println("Did not find fingerprint sensor
:("); lcd.clear(); lcd.print("module not Found"); lcd.setCursor(0,1);
lcd.print("Check Connections"); while (1);
if (! rtc.begin()) Serial.println("Couldn't find RTC");
// rtc.adjust(DateTime(F($_{DATE}$),F($_{TIME}$)));

if (! rtc.isrunning()) Serial.println("RTC is NOT running!"); // following line sets the RTC to the date time this sketch was compiled
rtc.adjust(DateTime(F($_{DATE}$),F($_{TIME}$)));$//ThislinesetstheRTCwithanexplicitdatetime,forexampletoset//January21,2014at3amyouwou$
user1=EEPROM.read(1000); user2=EEPROM.read(1001);
user3=EEPROM.read(1002); user4=EEPROM.read(1003);
user5=EEPROM.read(1004); lcd.clear(); digitalWrite(indFinger, HIGH);
void loop() now = rtc.now(); lcd.setCursor(0,0); lcd.print("Time-¿");
lcd.print(now.hour(), DEC); lcd.print(':'); lcd.print(now.minute(), DEC);
lcd.print(':'); lcd.print(now.second(), DEC); lcd.print(" "); lcd.setCursor(0,1);
lcd.print("Date-¿"); lcd.print(now.day(), DEC); lcd.print('/');
lcd.print(now.month(), DEC); lcd.print('/'); lcd.print(now.year(), DEC);
lcd.print(" "); delay(500); int result=getFingerprintIDez(); if(result¿0)
digitalWrite(indFinger, LOW); digitalWrite(buzzer, HIGH); delay(100);
digitalWrite(buzzer, LOW); lcd.clear(); lcd.print("ID:"); lcd.print(result);
lcd.setCursor(0,1); lcd.print("Please Wait...."); delay(1000);
attendance(result); lcd.clear(); lcd.print("Attendance "); lcd.setCursor(0,1);
lcd.print("Registed"); delay(1000); digitalWrite(indFinger, HIGH); return;
checkKeys(); delay(300);
// dmyyhms - 7 bytes void attendance(int id) int user=0,eepLoc=0; if(id ==
1) eepLoc=0; user=user1++; else if(id == 2) eepLoc=210; user=user2++;
else if(id == 3) eepLoc=420; user=user3++; else if(id == 4) eepLoc=630;
user=user4++;
else return;
int eepIndex=(user*7)+eepLoc; EEPROM.write(eepIndex++, now.hour());
EEPROM.write(eepIndex++, now.minute()); EEPROM.write(eepIndex++,
now.second()); EEPROM.write(eepIndex++, now.day());
EEPROM.write(eepIndex++, now.month()); EEPROM.write(eepIndex++,
now.year()¿¿8); EEPROM.write(eepIndex++, now.year());
EEPROM.write(1000,user1); EEPROM.write(1001,user2);
EEPROM.write(1002,user3); EEPROM.write(1003,user4); //
EEPROM.write(4,user5); // figth user
void checkKeys() if(digitalRead(enroll) == 0) lcd.clear(); lcd.print("Please
Wait"); delay(1000); while(digitalRead(enroll) == 0); Enroll();
else if(digitalRead(del) == 0) lcd.clear(); lcd.print("Please Wait");
delay(1000); delet();
void Enroll() int count=1; lcd.clear(); lcd.print("Enter Finger ID:");
while(1) lcd.setCursor(0,1); lcd.print(count); if(digitalRead(up) == 0)
count++; if(count¿records) count=1; delay(500);

else if(digitalRead(down) == 0) count–; if(count¡1) count=records;
delay(500); else if(digitalRead(del) == 0) id=count; getFingerprintEnroll();
for(int i=0;i¡records;i++) if(EEPROM.read(i) != 0xff) EEPROM.write(i, id);
break; return;
else if(digitalRead(enroll) == 0) return;
void delet() int count=1; lcd.clear(); lcd.print(”Enter Finger ID”);
while(1) lcd.setCursor(0,1); lcd.print(count); if(digitalRead(up) == 0)
count++; if(count¿records) count=1; delay(500);
else if(digitalRead(down) == 0) count–; if(count¡1) count=records;
delay(500); else if(digitalRead(del) == 0) id=count; deleteFingerprint(id);
for(int i=0;i¡records;i++) if(EEPROM.read(i) == id) EEPROM.write(i,
0xff); break; return;
else if(digitalRead(enroll) == 0) return;
uint8$_tgetFingerprintEnroll()intp = -1; lcd.clear(); lcd.print(”fingerID : ”); lcd.print(id); lcd.setCursor(0, 1$
// OK success!
p = finger.image2Tz(1); switch (p) case FINGERPRINT$_OK :$
$Serial.println(”Imageconverted”); lcd.clear(); lcd.print(”Imageconverted”); break; caseFINGERPRINT_IM$
$Serial.println(”Imagetoomessy”); lcd.clear(); lcd.print(”Imagetoomessy”); returnp; caseFINGERPRINT_P$
$Serial.println(”Communicationerror”); lcd.clear(); lcd.print(”CommError”); returnp; caseFINGERPRIN$
$Serial.println(”Couldnotfindfingerprintfeatures”); lcd.clear(); lcd.print(”FeatureNotFound”); returnp; ca$
$Serial.println(”Couldnotfindfingerprintfeatures”); lcd.clear(); lcd.print(”FeatureNotFound”); returnp; de$
$Serial.println(”Unknownerror”); lcd.clear(); lcd.print(”UnknownError”); returnp;$
Serial.println(”Remove
finger”); lcd.clear(); lcd.print(”Remove Finger”); delay(2000); p = 0; while (p !=
FINGERPRINT$_NOFINGER)p = finger.getImage();Serial.print(”ID”); Serial.println(id); p =$
$-1; Serial.println(”Placesamefingeragain”); lcd.clear(); lcd.print(”PlaceFinger”); lcd.setCursor(0, 1); lcd.p$
$FINGERPRINT_OK)p = finger.getImage(); switch(p)caseFINGERPRINT_OK : Serial.println(”Imageta$
// OK success!
p = finger.image2Tz(2); switch (p) case FINGERPRINT$_OK :$
$Serial.println(”Imageconverted”); break; caseFINGERPRINT_IMAGEMESS :$
$Serial.println(”Imagetoomessy”); returnp; caseFINGERPRINT_PACKETRECIEVEERR :$
$Serial.println(”Communicationerror”); returnp; caseFINGERPRINT_FEATUREFAIL :$
$Serial.println(”Couldnotfindfingerprintfeatures”); returnp; caseFINGERPRINT_INVALIDIMAGE :$
$Serial.println(”Couldnotfindfingerprintfeatures”); returnp; default :$
$Serial.println(”Unknownerror”); returnp;$
// OK converted! Serial.print(”Creating model for ”); Serial.println(id);
p = finger.createModel(); if (p ==
FINGERPRINT$_OK)Serial.println(”Printsmatched!”);elseif(p ==$
$FINGERPRINT_PACKETRECIEVEERR)Serial.println(”Communicationerror”); returnp;elseif(p ==$
$FINGERPRINT_ENROLLMISMATCH)Serial.println(”Fingerprintsdidnotmatch”); returnp;elseSerial.$
Serial.print(”ID ”); Serial.println(id); p = finger.storeModel(id); if (p ==
FINGERPRINT$_OK)Serial.println(”Stored!”); lcd.clear(); lcd.print(”Stored!”); delay(2000);elseif(p ==$
$FINGERPRINT_PACKETRECIEVEERR)Serial.println(”Communicationerror”); returnp;elseif(p ==$
$FINGERPRINT_BADLOCATION)Serial.println(”Couldnotstoreinthatlocation”); returnp;elseif(p ==$
$FINGERPRINT_FLASHERR)Serial.println(”Errorwritingtoflash”); returnp;elseSerial.println(”Unkno$
int getFingerprintIDez() uint8$_tp = finger.getImage();$

if (p != FINGERPRINT$_OK)return-1;$
p = finger.image2Tz(); if (p != FINGERPRINT$_OK)return-1;$
p = finger.fingerFastSearch(); if (p !=
FINGERPRINT$_OK)lcd.clear();lcd.print("FingerNotFound");lcd.setCursor(0,1);lcd.print("TryLater");de$
uint8$_tdeleteFingerprint(uint8_tid)uint8_tp=-1;lcd.clear();lcd.print("Pleasewait");p=finger.deleteModel$
else Serial.print("Something Wrong"); lcd.clear(); lcd.print("Something
Wrong"); lcd.setCursor(0,1); lcd.print("Try Again Later"); delay(2000); return
p;
void download(int eepIndex)
if(EEPROM.read(eepIndex) != 0xff) Serial.print("T-¿");
if(EEPROM.read(eepIndex)¡10) Serial.print('0');
Serial.print(EEPROM.read(eepIndex++)); Serial.print(':');
if(EEPROM.read(eepIndex)¡10) Serial.print('0');
Serial.print(EEPROM.read(eepIndex++)); Serial.print(':');
if(EEPROM.read(eepIndex)¡10) Serial.print('0');
Serial.print(EEPROM.read(eepIndex++)); Serial.print(" D-¿");
if(EEPROM.read(eepIndex)¡10) Serial.print('0');
Serial.print(EEPROM.read(eepIndex++)); Serial.print('/');
if(EEPROM.read(eepIndex)¡10) Serial.print('0');
Serial.print(EEPROM.read(eepIndex++)); Serial.print('/');
Serial.print(EEPROM.read(eepIndex++)¡¡8 — EEPROM.read(eepIndex++));
else Serial.print("——————————");
Serial.print(" ");

2.0.2 Simulation output



```
Executable segment sizes:
ICACHE : 32768           - flash instruction cache
IROM   : 250368          - code in flash         (default or ICACHE_FLASH_ATTR)
IRAM   : 29477   / 32768 - code in IRAM          (IRAM_ATTR, ISRs...)
DATA   : 1504  )         - initialized variables (global, static) in RAM/HEAP
RODATA : 1948  ) / 81920 - constants             (global, static) in RAM/HEAP
BSS    : 26656 )         - zeroed variables      (global, static) in RAM/HEAP
Sketch uses 283297 bytes (27%) of program storage space. Maximum is 1044464 bytes.
Global variables use 30108 bytes (36%) of dynamic memory, leaving 51812 bytes for local variables. Maximum is 81920 bytes.
```

Bibliography

[1] https://learn.adafruit.com/adafruit-optical-fingerprint-sensor

[2] https://whatis.techtarget.com/definition/LCD-liquid-crystal-display: :text=LCD

[3] https://www.arduino.cc/reference/en/libraries/adafruit-fingerprint-sensor-library/