BIOMETRIC PARKING SYSTEM

Final Year Project

Bachelor of Science (Electronics)

Submitted by
Bilal Ahmed

Department of Electronics

Faculty of Natural Sciences
Quaid-i-Azam University Islamabad

Declaration

I hereby declare that the Project titled "Biometric Parking System" submitted to the department of Electronics, Faculty of Natural Sciences, Quaid-i-Azam University Islamabad for the award of degree of Bachelor of Science in Electronics (BS-Electronics). The project work is carried out by our group under the supervision of Dr. Mussarat Abbass.

Bilal

Bilal Ahmed
Date: December 29, 2018

Approval Certificate

This is certified that the project titled **"Biometric Parking System"** submitted by Bilal Ahmed to the Department of Electronics, Faculty of Natural Sciences, Quaid-i-Azam University Islamabad, Pakistan, is accepted in its present form as it is satisfying the requirement for the degree of Bachelor of Science in Electronics (BS Electronics).

Dr. Mussarat Abbass (Superisor)

Assistant Professor

Department of Electronics

Quaid-i-Azam University Islamabad, Pakistan.

Dr. Hassan Mahmood

Chairman,

Department of Electronics

Quaid-i-Azam University Islamabad, Pakistan.

Acknowledgements

We are very obliged to our Supervisor Dr. Mussarat Abbass, for his special attention and guidance in the right direction. Without him, it would be literally impossible to complete the project.

We are also grateful to Prof. Dr. Hassan Mahmood, Chairman, Department of Electronics under whose guidance we were able to achieve this task.

We would also like to extend our deepest thanks to all academic colleagues, who were helpful directly or indirectly in achieving our task.

Table of Contents

Abbreviations

ADC	Analog to Digital Converter
AREF	Analog Reference
MISO	Mater In, Slave Out
MOSI	Mater Out, Slave In
PWM	Pulse Width Modulation
RTC	Real Time Clock
TTL	Transistor-Transistor Logic
Rx and **Tx**	Receive and Transmit
SCK	Serial Clock
SDA	Serial Data
SPI	Serial Peripheral Interface
SS	Slave Select

Abstract

Security is one of the major need of any human being, we have focused our project for security of cars in parking area. Our project is related to the concept of biometric security system for car parking. As the finger prints for each individual are specific hence there are minimum chances of security compromise. This system also enables the user to keep attendance record of each individual parking the car. This system also display the information of the individual using parking area which is displayed on the monitor. We have developed the system by using fingerprint sensor, Arduino UNO and RTC module. Programming for Arduino is done by use of software called Arduino IDE. The program establishes connection between the Arduino and base system.

CHAPTER 1

Introduction

1.1 Introduction

Security is one of the major need of any person. Everyone demands security for their life and their belongings. Car security in Pakistan is important as there are so many cars get stolen every day. For many organizations there is no proper place to park the cars. People have to park their cars on road sides or any place available to them. They are unsure whether the parked car will be there or not when they will get back. To meet the transportation needs the number of cars is increasing day by day but no measurements have been taken to provide proper and secure parking area.

1.2 Problem Statement

For secure and reserve parking area for every organization, a proper security system should be installed there. There are many types of electronic security systems by which we can make the parking secure. Following are some of the popular parking systems:

- RFID (Radio Frequency ID)
- Biometric System
- AVR Microcontroller
- Retina Detection

By using one of these systems, parking can be made easy and safe. We focused on biometric parking for security. Biometric parking would be very helpful for the parking of cars. It will be secure and efficient for users, as fingerprints are specific for each individual with minimum chances of security compromise.

1.3 Motivation

This type of parking provides safe and secure parking. User just need one time registration to get benefit from this facility. If he/she get registered then they can use parking easily. User just require to place his/her finger on the fingerprint sensor and the gates will open, in this way

coming in and getting out of parking area will be easier and safe. Apart from facility of biometric parking it can also serve the purpose of record keeping as it will provide the record of every person's identification and timing of entering and leaving the duty/job place.

1.4 Biometric Parking -The New Trend

With the passage of time, new technology is replacing the old technology and bringing betterment and advancement in human life-style. Everything is becoming smart and easy in use for routine life. Similarly, the use of biometric parking in daily life is bringing ease and efficiency. Introduction of biometric parking system can reduce the need of manpower handling parking affairs as this smart technology requires no guard. Lifestyle nowadays is heavily inclined to automation, where people like convenience, safety, time conservation and advancement at same time.

1.5 Importance

This biometric parking system not only serve the basic purpose of security and record keeping but is also helpful in keeping real-time attendance record which is specifically helpful for supervisors and managers. This application provide convenience to all the users including employees, customers, guests, visitors and administration etc.

1.5.1 Operational Efficiency

To bring functional and operational efficiency the system should be managed efficiently. This biometric parking system should be in proper working condition and should be checked periodically.

1.5.2 Fingerprint Sensor Access

Fingerprint sensor should properly work. It should detect and respond the registered ID's and allow them immediate entry.

1.5.3 Efficient Management

The data of all users is stored in computer and is managed by a single efficient program store to the system that makes the efficient parking. The system stores the data of all registered members

of the organization. There is reserved parking point for each person, so that person can enter anytime.

1.6 Aims & Objectives

The aim of the project is to design a prototype for maintaining the parking of any organization. The number of people may be increased anytime. The system is able to keep whole data correctly which is stored in the form of registered ID's.

The main objectives of the biometric parking system are

- Maintain the parking and ensuring easy and fast entry.
- Providing car security.
- Maintain the data of employees like identification and attendance.

1.6.1 Maintaining the Parking

To maintain the parking means that there is one reserved parking slot for each employ registered in the biometric parking system. This ensure that every registered person can get the parking at any time without any hassle.

1.6.2 Managing the Data of All Employees

As the user places his/her finger on the fingerprint sensor, the system records the time and identify the individual by using serial monitor which has record of all registered users. This serve the purpose of record keeping and is helpful in keeping real-time attendance record which is specifically supportive to supervisors and managers.

1.7 Thesis Plan

This report contains a comprehensive study of the undergoing project discussing each and every step till completion. Chapter 2 presents an overview of the introduced system and an insight, how our project is divided into various phases. It also describes how every stage is integrated with next stage to give a working of the system. Implementation of the system based on information

overviewed is present in chapter 3. Working principle of the project is explained in chapter 4. Chapter 5 presents the conclusion of our work and suggestions for the future work.

CHAPTER 2

System Overview

2.1 Overview

The biometric parking system is mainly divided into two parts, one is the entrance to the gate by biometric scanning and providing security while the other is to maintain record of the registered user with this system.

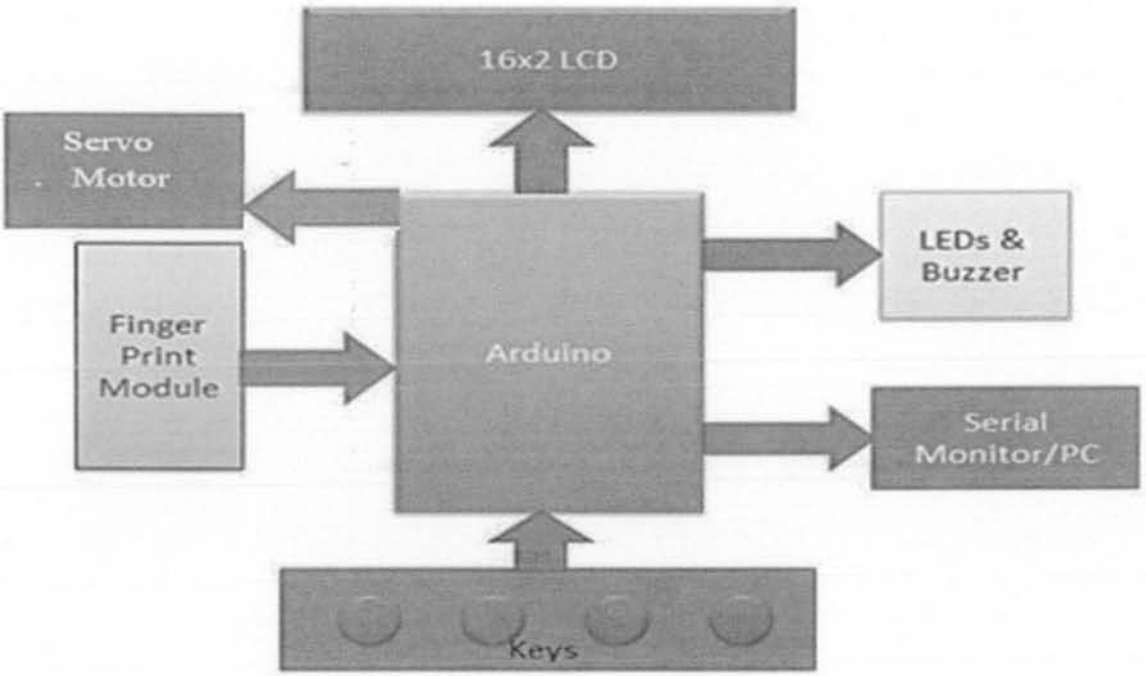

Fig 2. 1 Block Diagram of Basic Components of the Project

2.2 Biometric Parking System

Biometric parking system have ability to convert the normal manual guard based parking to the secure and technology based smart parking in which no man power is wasted. This system has evolved the conventional parking issue into the electronic based parking.

2.3 Required Components

- Arduino
- Finger print module
- Push Button - 4 (not required if LCD keypad shield is used)
- LED's
- Connecting wires
- Buzzer
- 16x2 LCD
- Bread Board
- RTC Module

2.4 Arduino (UNO)

Arduino UNO is an ATmega328P (data sheet) based microcontroller card. It has 14 digital input/output pins (6 pins can be utilized as PWM output) .There are 6 analog input pins, 16 MHz quartz precious stone, one USB association or connection, one power input, one ICSP header and one reset button. It contains everything expected to help the microcontroller.

2.4.1 Pins Configuration

It has 14 digital input/output pins that can be utilized as input or output pins in Arduino programming by pinMode (), digitalRead () and digitalWrite () functions. Each pin works at 5V and can give or take up to 40mA of current and has an inside force obstruction of 20-50 kilo ohms. Aside from these 14 pins, few pins have particular capacities as recorded underneath.

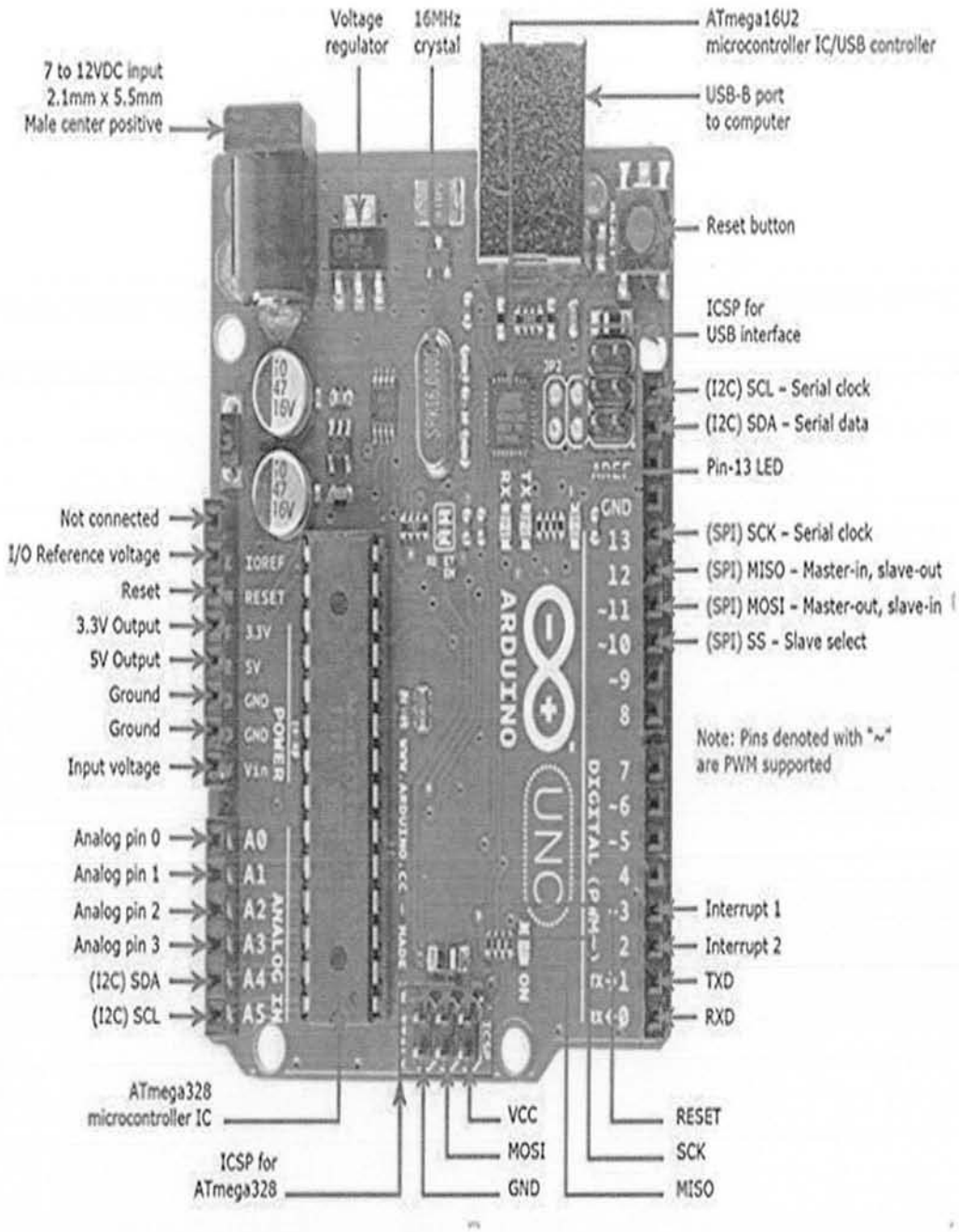

Fig 2. 2 Pins Configuration of Arduino UNO

- **Serial Pins 0 (Rx) and 1 (Tx):** Rx and Tx pins are utilized to receive and transmit TTL serial information.

- **External interference Pins 2 and 3:** These pins are used to trigger a cut at a low value, a rising or falling edge, or an adjustment in value.

- **Analog Pin 4 (SDA)** and **Pin 5 (SCA):** are additionally utilized for TWI correspondence utilizing Wire library

  - **PWM Pins 3, 5, 6, 9, and 11:** These pins give 8-bit PWM yield utilizing the analogWrite () work.
  - **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are utilized for SPI correspondence.
  - **Built-in Drove Pin 13:** Pin 13 high. If the Drove is lit and pin 13 is low, it is associated with an inside driven on the off chance that it is shut.

  Arduino (UNO) has some different pins which are as following:

  - **AREF**: This pin give reference voltage to simple contributions with analogReference() work.
  - Reset Pin: Resets the microcontroller when pin is low.
  - The operating voltage of Arduino is 5v input voltage recommended is 7-12 volts and the input voltage limits are 6-20 volts.
  - Flash memory is 32 KB of which 0.5 KB used by boot loader.
  - SRAM is of 2 KB (ATmega 328)
  - Clock speed is 16 MHz.
  - EEPROM is of 1KB.

2.4.2 Programming of Arduino UNO

Arduino UNO is programmed by its Arduino software (Arduino IDE). Arduino software is a kind of compiler which helps to write the code. This software can also burn the code in the Arduino microcontroller. Coding language is C and C++ through which we can call different functions in Arduino. This software is used to program all kind of Arduino's.

2.5 LCD

LCD stands for Liquid Crystal Display.

2.5.1 LCD with Keypad Shield

Arduino LCD Keypad shield allows users to change from the menu, to make selections etc. This keypad shield is made to provide user friendly interface. A 1602 white character contains a blue lit up LCD. This LCD keypad shield consists of 5 keys left, right, up, down and select. The keypad shield interface stores digital I/O pins by only using single ADC channel. The key value is checked with a 5-step voltage divider.

In this LCD keypad shield the connections are made by the designer where, the pin 9 is fixed for enable and the digital pins 4,5,6,7 are fixed for data lines.

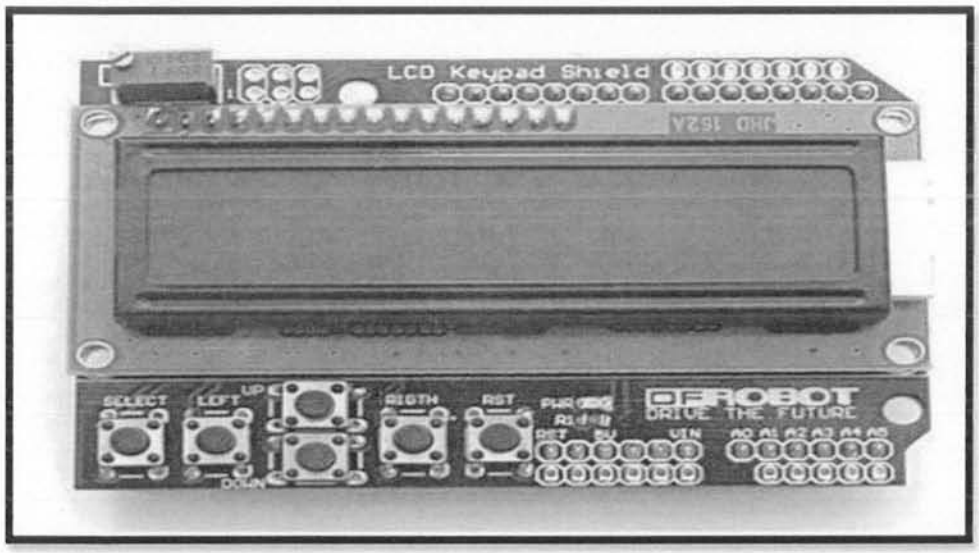

Fig 2. 3 LCD with Keypad Shield

The push buttons are connected to analog pin which is A0, and these buttons are connected with the series of resistors.

2.5.2 LCD without Keypad

When LCD without keypad shield is used one has to connect all LCD pin with appropriate Arduino.

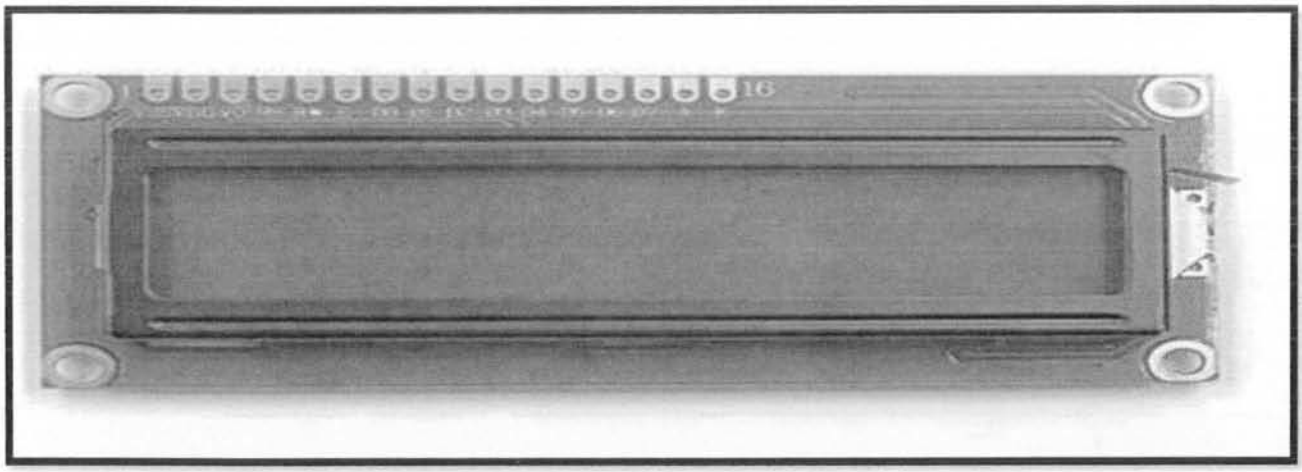

Fig 2. 4 LCD withOutKeypad Shield

Besides the keypad shied LCD without keypad LCD can also be used. For this we have to use 4 push buttons for enrolling, deleting, selecting, up and down. Push Button is shown in figure below:

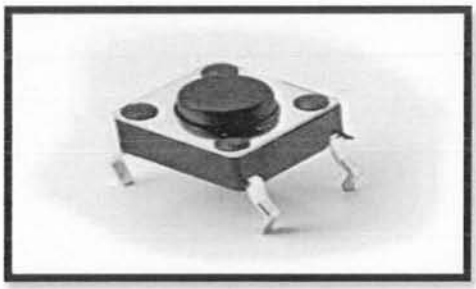

Fig 2. 5 Push Button

2.6 Fingerprint Module

High quality fingerprint sensors/scanners are used in cell phones, in banks ATM and many other technological advancements. These biometric sensors are utilized for specificity and security. Due to technology improvement the fingerprint sensors work quickly and are more precise in the

identifications. In view of all that, now we will find how the most recent fingerprint sensor/scanners work and know its features.

Optical biometric fingerprint devices use advanced features and are installed in an organization for multiple purpose. For example, control room, security store box, auto entrance locks etc.

In this project we are using fours wires of this fingerprint module white, green, red and black.

1. Green wire is used for the transmission from the sensor to Arduino.
2. White wire is used to receive signal from the Arduino to the sensor.
3. Black wire work is used as ground wire.
4. Red wire provide power to the sensor which is VCC (5V).

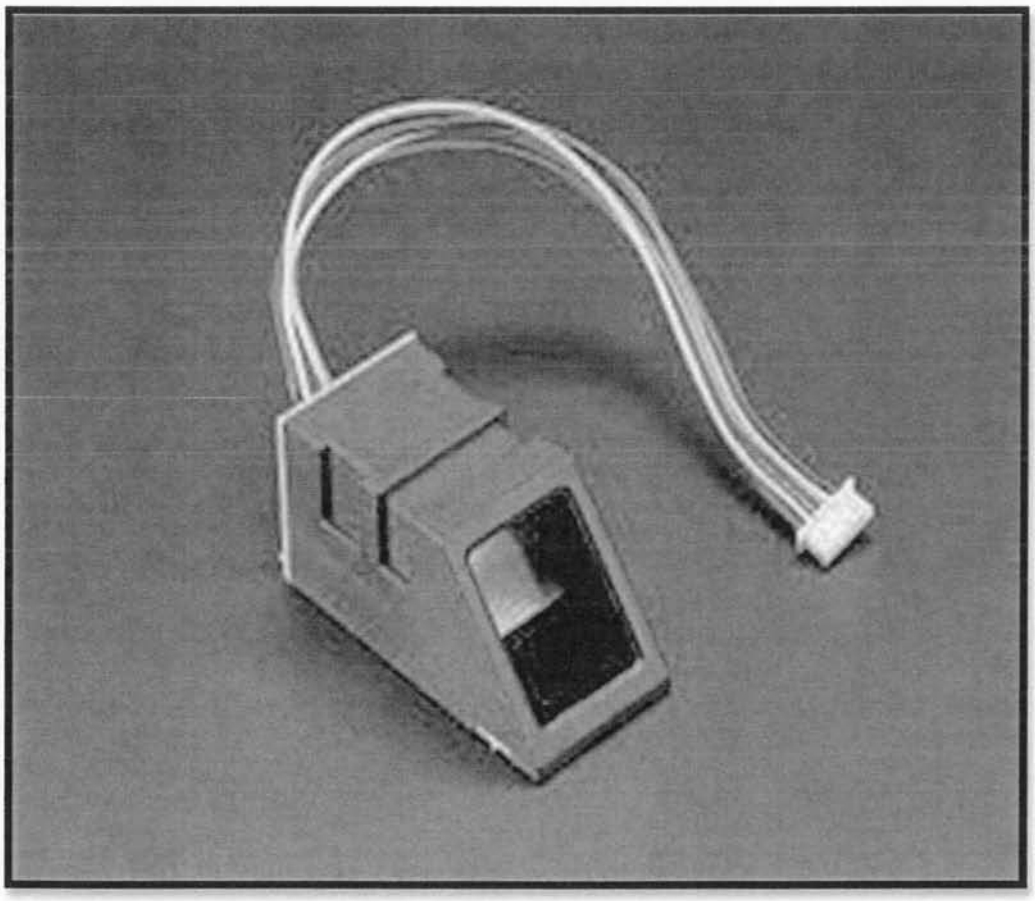

Fig 2. 6 Fingerprint Sensor with Front Image

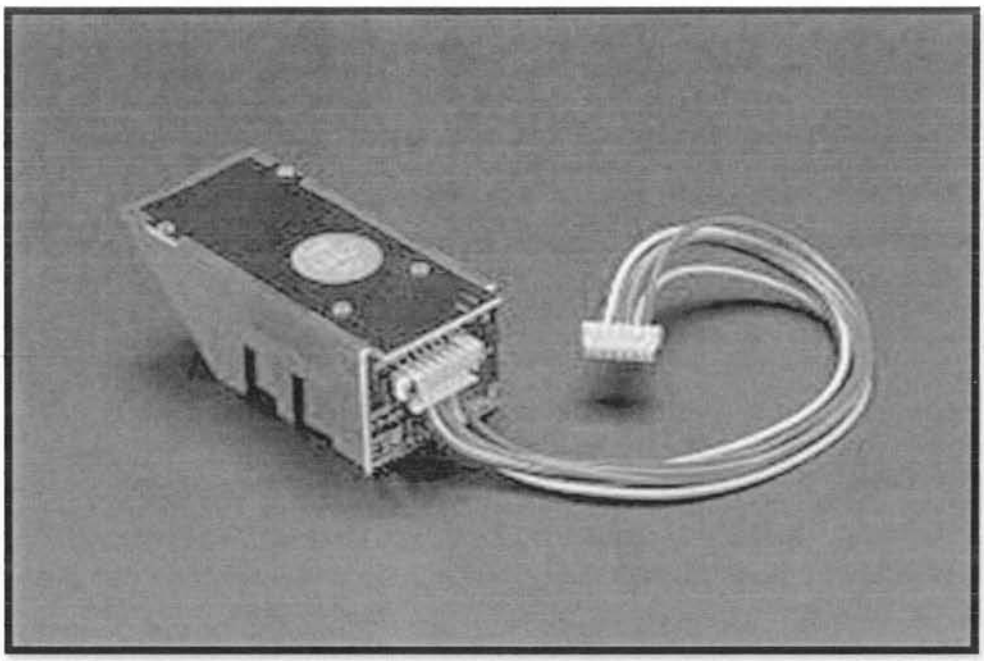

Fig 2.7 Fingerprint Sensor with Back Image

- Integrated image collecting and algorithm chip together, All-in-one
- Fingerprint reader can conduct secondary development, can be embedded into a variety of end products
- Low power consumption, low cost, small size, excellent performance
- Professional optical technology, precise module manufacturing techniques
- Good image processing capabilities, can successfully capture image up to resolution 500 dpi

Features

Specifications

- Fingerprint sensor type: Optical
- Sensor Life: 100 million times
- Static indicators: 15KVBacklight: bright green
- Interface: USB1.1/UART(TTL logical level)
- RS232 communication baud rate: 4800BPS~115200BPS changeable
- Dimension: 55*32*21.5mm
- Image Capture Surface 15—18(mm)
- Verification Speed: 0.3 sec
- Scanning Speed: 0.5 sec
- Character file size: 256 bytes
- Template size: 512 bytes
- Storage capacity: 250
- Security level: 5 (1,2,3,4,5(highest))
- False Acceptance Rate (FAR) :0.0001%

- False Rejection Rate (FRR): 0.1%
- Resolution 500 DPI
- Voltage :3.6-6.0 VDC
- Working current: Typical 90 mA, Peak 150mA
- Matching Method: 1: N
- Operating Environment Temperature: -20 to 45° centigrade

2.6.1 Working of Fingerprint Scanner

Generally there are two types of scanners which are used, one is optical scanner and second is capacitance scanner. In optical scanner images are scanned by use of light while in capacitance scanner electrical current is used to display image.

2.6.1.1 Optical Scanner

For this project we used optical scanner and our discussion here is only for optical scanner. The following image show the working of the fingerprint sensor

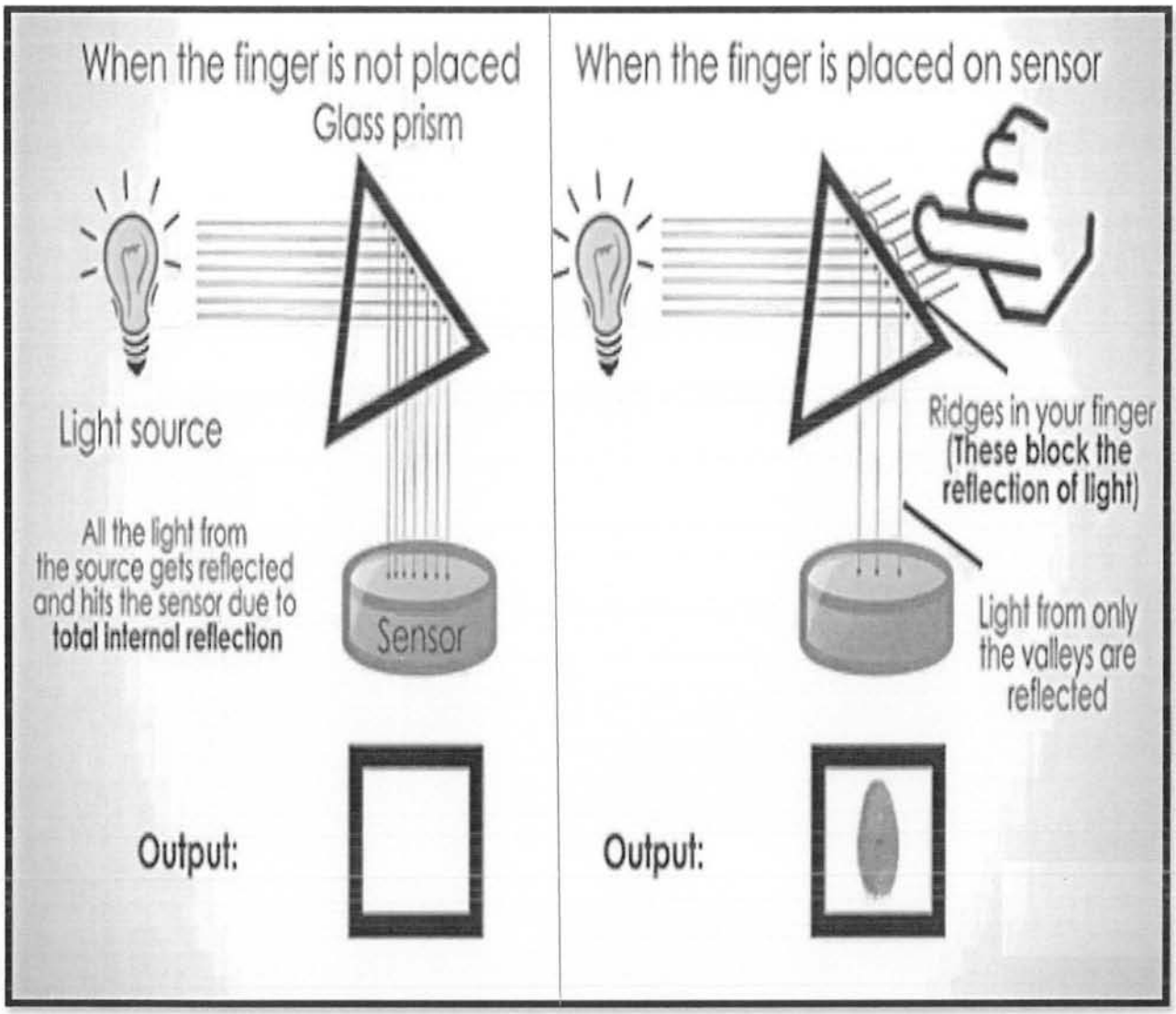

Fig 2. 8 Fingerprint Sensor Working

When a light from any light source falls on glass prism of the fingerprint module it is then totally reflected and hit the sensor directly because of the total internal reflection. When nobody places the finger on the fingerprint sensor there is no output. No output means that we do not have any image of finger which can be stored in the memory of fingerprint sensor. When the finger is placed on the glass prism the ridges in your finger block the reflection of the light and some of the light rays are blocked due to the ridges and some of the light rays are reflected because of the valleys of the finger, by this method both reflected and non-reflected rays form the image and that image can be stored in the fingerprint sensor memory.

Figure 2.9 shows valleys and ridges of the finger and these altogether give grating to the skin.

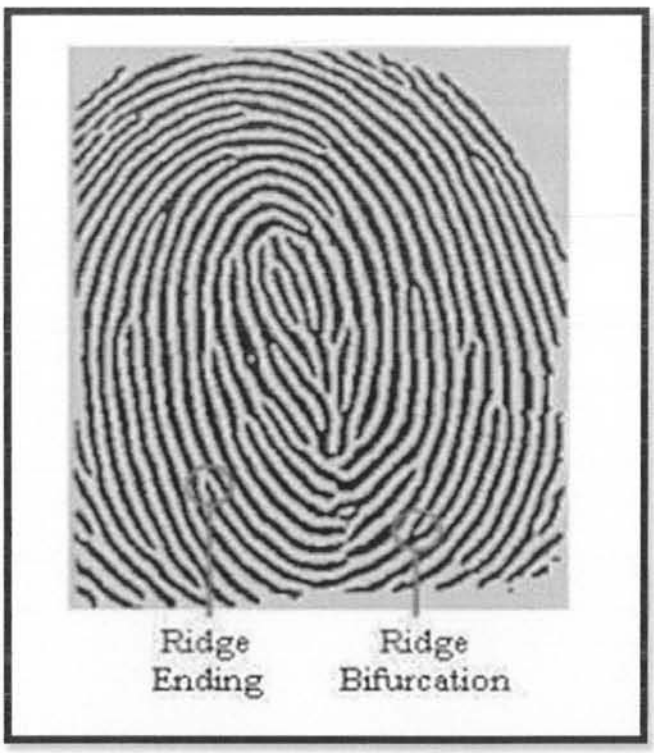

Fig 2. 9 Valleys and Ridges of the Finger

There are various characteristics of fingerprint and these characteristics are also useful for method of identification. Following diagram shows some other characteristics of fingerprint:

Fig 2.10 Characteristics of Fingerprint

Fingerprint scanner has two main functions, first it needs to capture an image of your finger, and then it needs to determine whether the pattern of ridges and valleys in this image matches the pattern of ridges and valleys in pre-scanned images.

The principle associated to the optical scanner is the Charge Coupled Device (CCD),the basic component of an optical scanner is a charge coupled device (CCD), digital cameras and camcorders also use CCD light technique. A CCD is simply a ray of light sensitive diodes which are called photosites A CCD creates an electrical signal in reaction to light photons. Each photosite keeps a pixel i.e. a small dot to show the light that hits that area. Together the dim and bright pixels form an image of the scanned scene (a finger, for example). Typically, an analog-to-digital converter in the scanner system processes the analog electrical signal to generate a digital representation of this image.

The scanning process starts when a finger is placed on a glass plate, and then a CCD camera takes a picture. The scanner has its own light source, typically an array of light-emitting diodes, to illuminate the ridges of the finger. The CCD system actually generates an inverted image of the finger with darker areas representing more reflected light (the ridges of the finger) and lighter areas representing less reflected light (the valleys between the ridges).

Before comparing the image to stored data, the scanner processor makes sure the CCD has captured a clear image. It checks the average pixel darkness or the overall values in a small sample and rejects the scan if the overall image is too dark or too light. If the image is rejected, the scanner adjusts the exposure time to let in more or less light, and then tries the scan again.

If the darkness level is adequate, the scanner system goes on to check the image definition (how sharp the fingerprint scan is). The processor looks at several straight lines moving horizontally and vertically across the image. If the fingerprint image has good definition, a line running perpendicular to the ridges is developed with alternating sections of very dark pixels and very light pixels.

If the processor finds that the image is crisp and properly exposed, it proceeds to comparing the captured fingerprint with fingerprints stored in file.

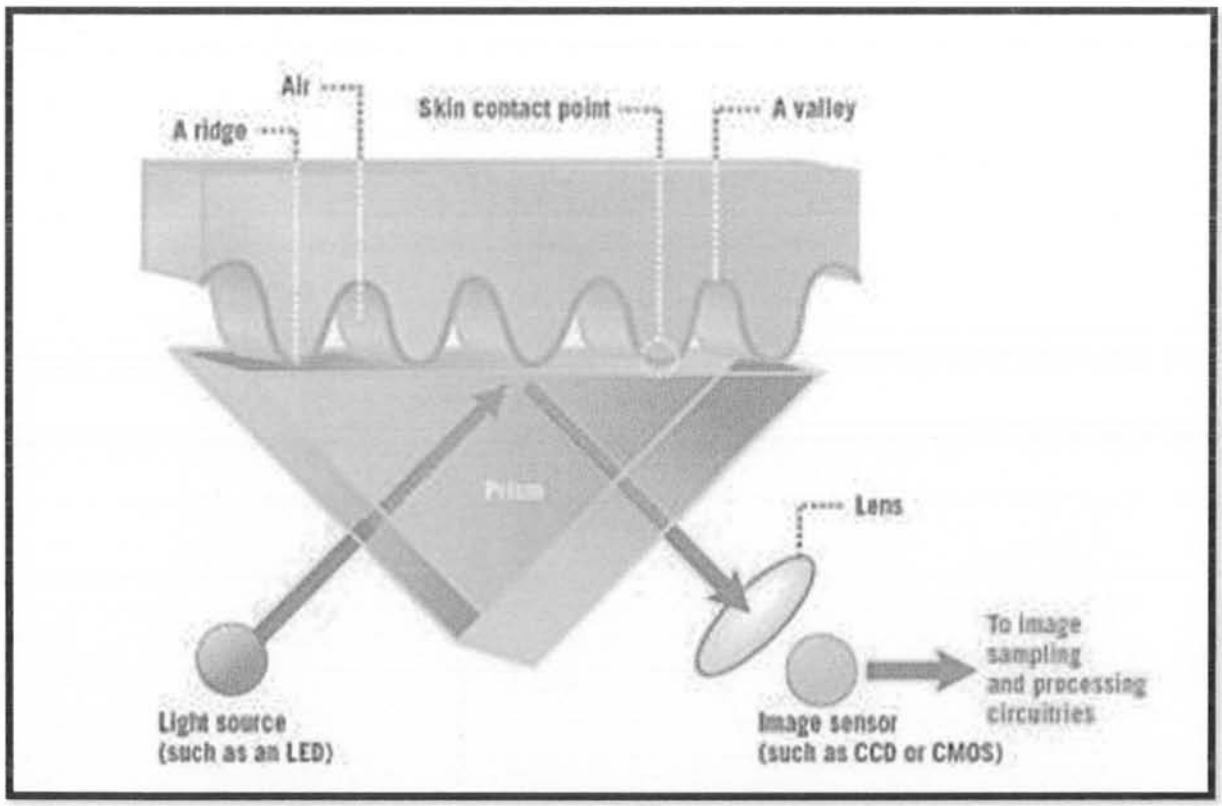

Fig 2.11 Working of Fingerprint Sensor

Every fingerprint is marked with some point or some line, these points or lines are stored in the form of binary data, as shown below:

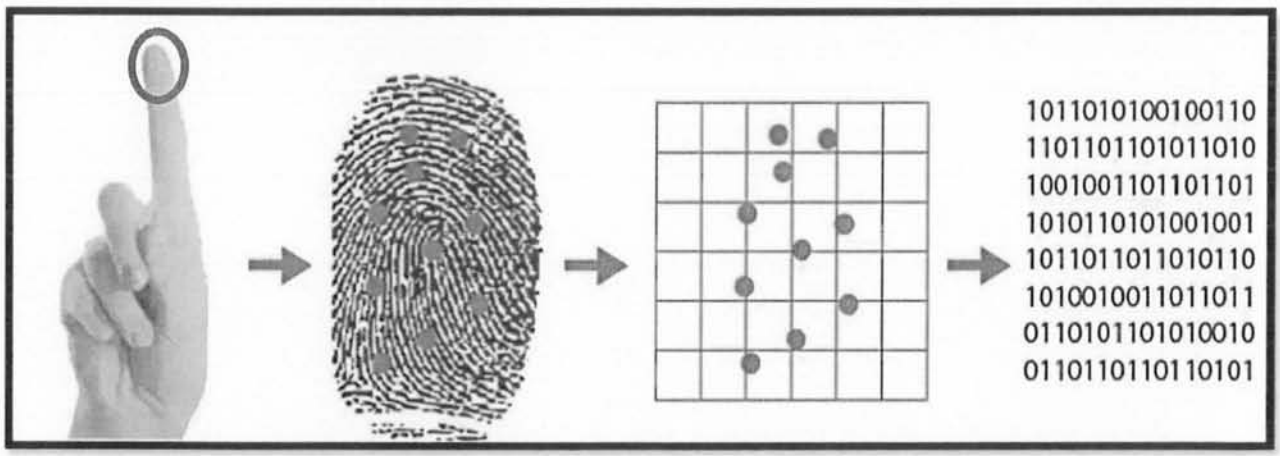

Fig 2. 12 Fingerprint is Marked with Some point or Some line, these are Stored in the Form of Binary Data.

The individuality and specificity of each fingerprint is because of the DNA of human genome in every individual. Due to unique fingerprint of every individual in the world there is no possibility that fingerprint will match with other person .So by capturing these we can keep the track of every single person.

Advantages of Biometric system

1. Physical proof of registered individual is confirmed.
2. No fake activity is possible like any sort of identity cards.
3. A password of someone can be figured but it is impossible to find out a fingerprint.
4. One may lose his identity card but cannot lose his fingerprint. A similar is the situation of a password, someone may lose his password.

Disadvantages of Biometric system

1. An optical scanner can be tricked by demonstrating a picture of a finger print as opposed to a genuine finger. Capacitance scanners may likewise be tricked by swapping a form of a finger by and in addition edges and valleys. On the off chance that it's some genuine business, a bandit may moreover remove the individual's finger and swap it on the scanner.
2. On the off chance that you have lost your identity card you'll have the new one for little money. In extreme case one may have shaped or molded a same finger print as yours, there's no real way to replace it.

2.7 RTC Module

RTC stands for Real Time Clock. The first question that arise here is why we really need to have a separate RTC for our Arduino project as the Arduino itself has built-in timekeeper. Well the purpose is that the RTC module runs on battery and may keep track of the time when the microcontroller is disconnected or reprogrammed.

A real time clock is essentially a bit like a watch, it runs on battery and keeps time for you even once there's a power outage. Using an RTC, we can keep track of long timelines, unless you reprogram your microcontroller or disconnect it from USB or an influence plug.

Most microcontrollers, as well as the Arduino, have a intrinsically(built in) timekeeper referred to as millis() and there are also timers which are designed to keep track of time longer than millis

like minutes, hours and days. Therefore why would we wish to own a separate RTC chip? Well, the most important reason is that millis() only keeps track of time since the Arduino was last battery powered, meaning that once the facility is turned on, the millisecond timer is about back to zero. The Arduino does not understand that it's 'Tuesday' or 'March 8th', all it will tell is 'It's been fourteen, 000 milliseconds since I used to be last turned on'.

What if we need to set the time on the Arduino? We must have to program within the date and time and we'll have it count from the time when it was on. However if it lost power, we'd have to be compelled to reset the time. Very similar to low cost alarm clocks.

While this type of basic activity is suitable for a few out comes, like data-loggers, clocks, etc. The RTC chip may be a specialized chip that simply keeps track of our time. It will count leap-years and is aware of days, weeks and month, however it does not beware of daylight savings time (because it changes from place to place).

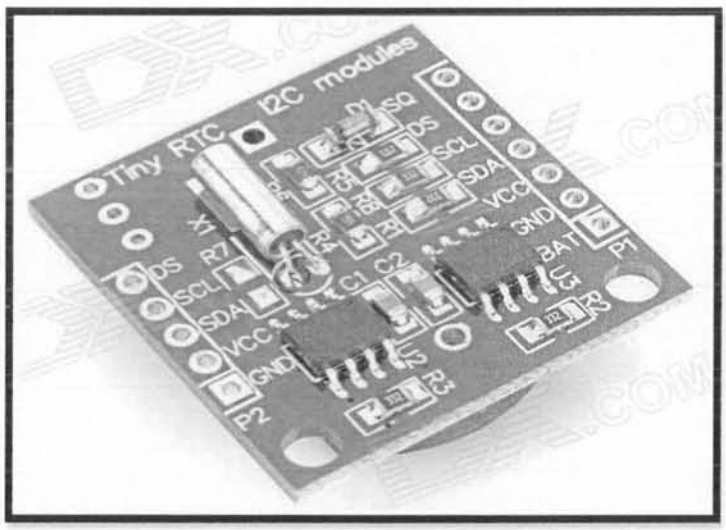

Fig 2. 13 RTC Module

Specifications

- Supply 5V DC
- Completely Manages All Timekeeping Functions
- Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of th Week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- Programmable Square-Wave Output Signal
- Simple Serial Port Interfaces to Most Microcontrollers
- I2C Serial Interface
- Low Power Operation Extends Battery Backup Run Time
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Automatic Power-Fail Detect and Switch Circuitry

2.8 Buzzer

Buzzer is used to alert the user that he/she can pass the gate.

Specifications

Item	Specification
Operating Voltage Range	200.0V~250.0VDC
Rated Current (max)	max.30mA
Sound Pressure Level	min.80dB
Frequency Of Output Signal	3,500 ± 200 Hz
Operating Temperature Range	-20℃ to +60℃
Storage Temperature Range	-30℃ to +70℃

Fig 2. 14 Buzzer

CHAPTER 3

Implementation

3.1. Implementation

Implementation knowledge with results of respective executions and the software design of the system along with the hardware components with their specifications are mentioned here. Also the entire hardware assembly is arranged and developed for its connection with the software.

3.1.1 Necessary steps to implement the project:

1. Connect Arduino with the USB port:

We select the board which is been used in our project from the given menu.

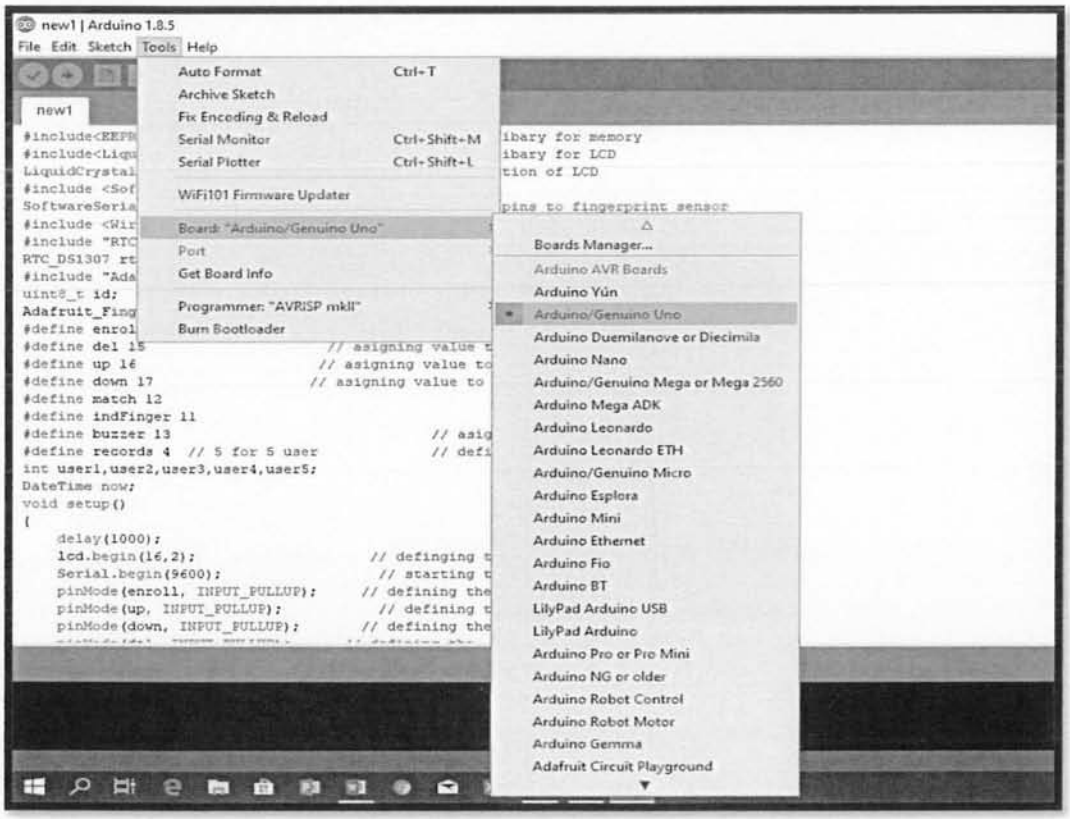

Fig 3. 1 Selecting the Board

This is important to connect our system with the computer.

2. Run Empty Code:

An empty code is burned in Arduino IDE which bypasses boot loader in UNO.

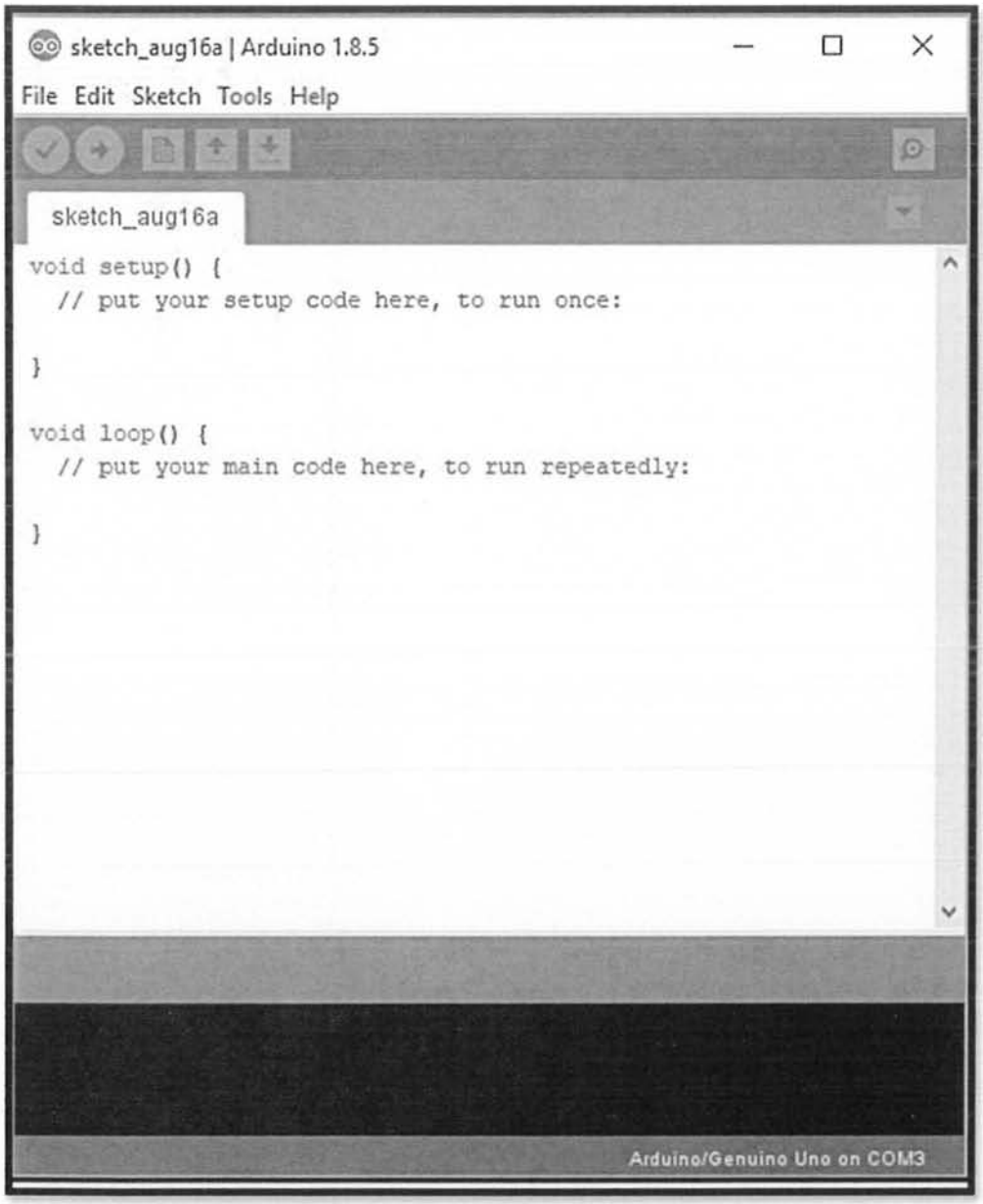

Fig 3. 2 Running the Empty Code to Check the Board

The empty code is executed whether to check Arduino is working or not. After this code is uploaded on Arduino to make it ready to work.

3. Verify if Arduino is Working or not by Applying Blink Test:

Before anything else, we have to make sure that device is working properly. In order to check it, following steps are performed:

- Plug the board into the USB port of computer and check either green light power LED illuminates or not. If it does not light up, it means that board is not receiving power. It is important to check the USB port connection.
- An orange or red LED near the center of the board "PIN 13 LED" should flash on and off when the board is powered up. The flashing LED is being controlled by the blink code running in the software.

How it works?

- Enter the 'blink' code in Arduino IDE and then verify the code to see any errors then load it. If loading is done successfully, LED 13 must be lit up showing Arduino is now in working state.

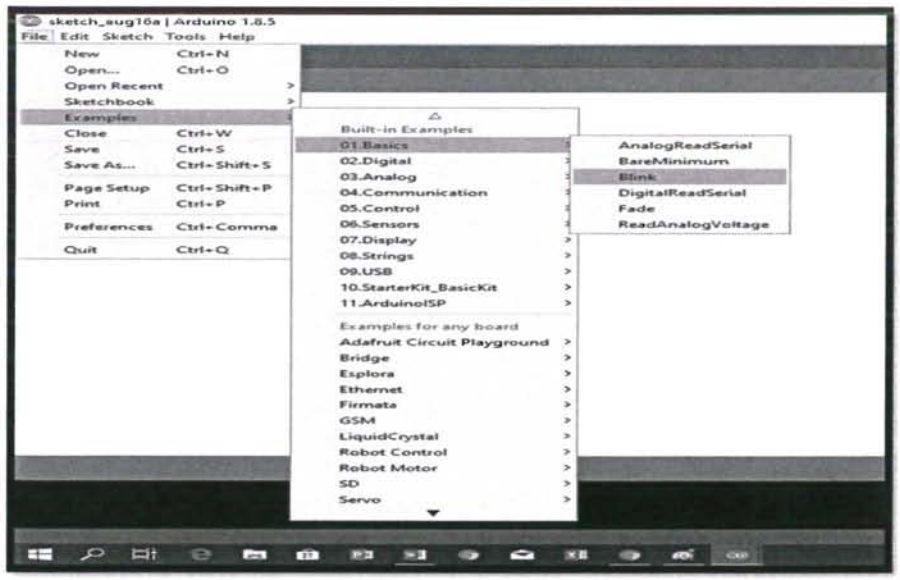

Fig 3. 3 Uploading the Example Code to Check the Working

Code for Blink:

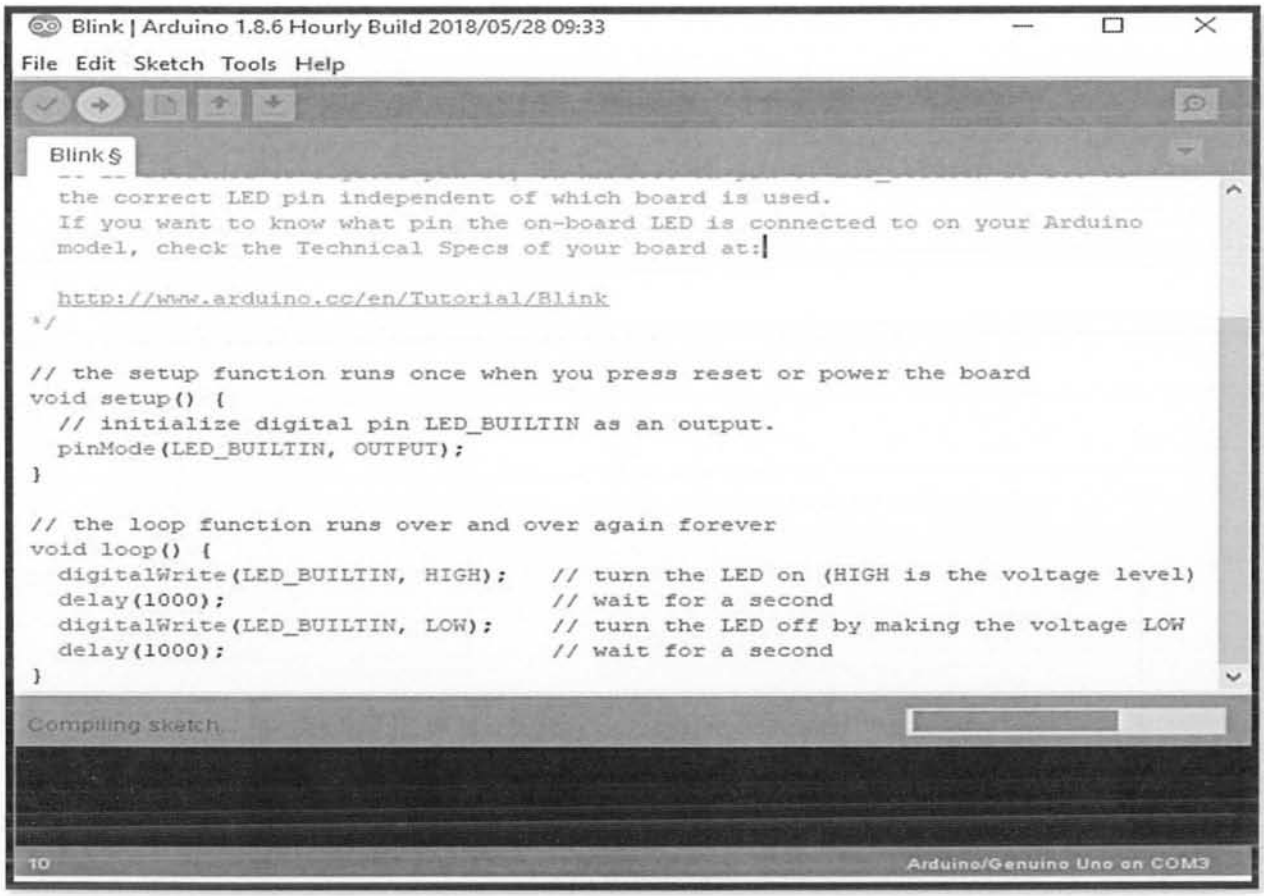

Fig 3. 4 Code for Blinking for LED

4. Disconnect Arduino from Computer

It is important to disconnect Arduino from computer before making any further connection of the required components with the Arduino.

5. The Connections of Fingerprint Sensor with Arduino

The Green wire of fingerprint sensor is connected with the digital pin 2 of Arduino, white wire of fingerprint sensor is connected with the digital pin 3, red wire of fingerprint sensor is connected with the 5V of Arduino and black wire of fingerprint sensor is connected with the ground of

Arduino. The connection is complete and now we check the fingerprint sensor working by uploading the enroll code which is used in this project.

For this we use the library which support the fingerprint sensor which is "Adafruit_Fingerprint.h".

- Check the enroll code:

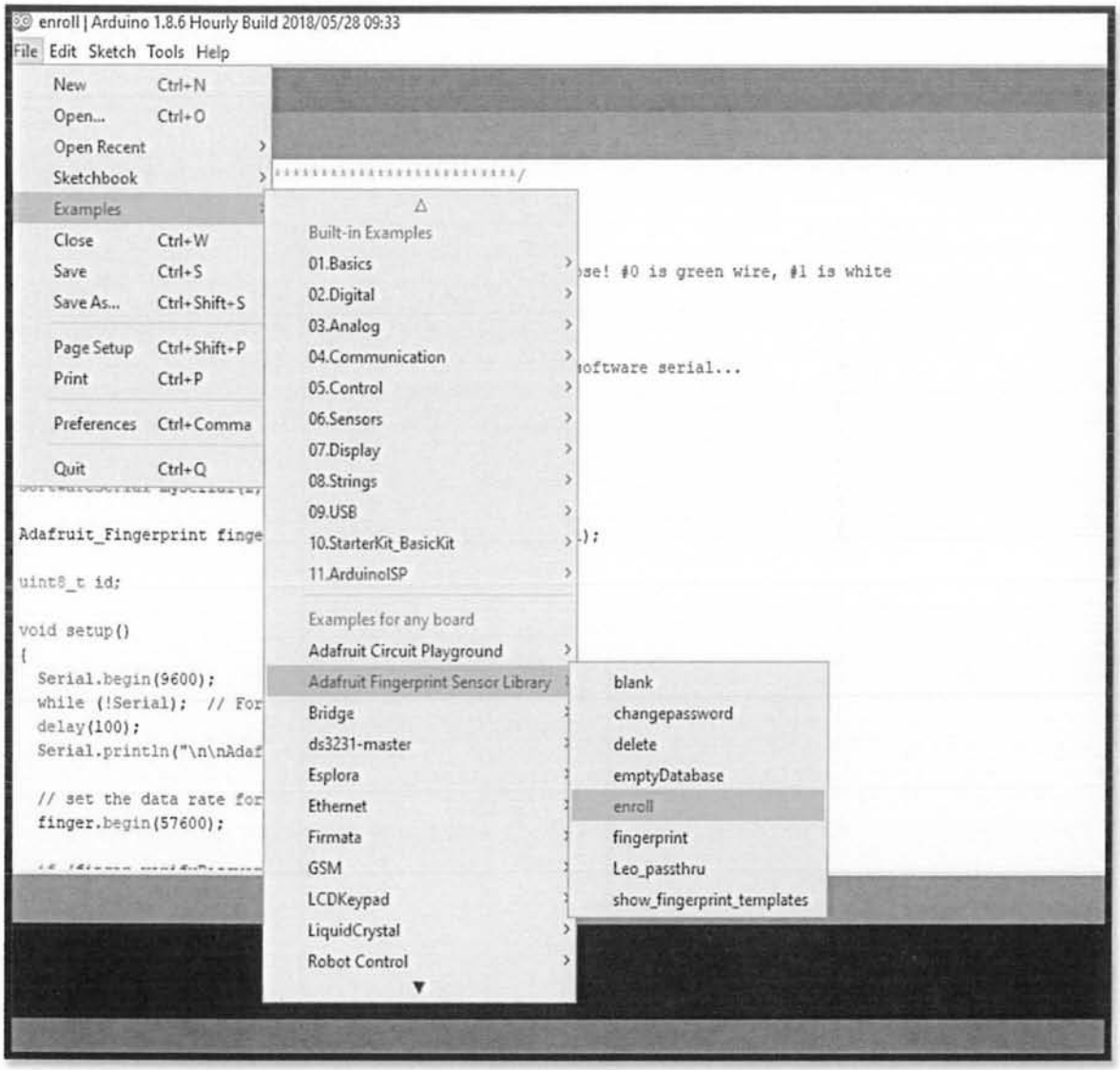

Fig 3. 5 Uploading the Code for Enrolling the Fingerprint on the Sensor

• Check the deleting code:

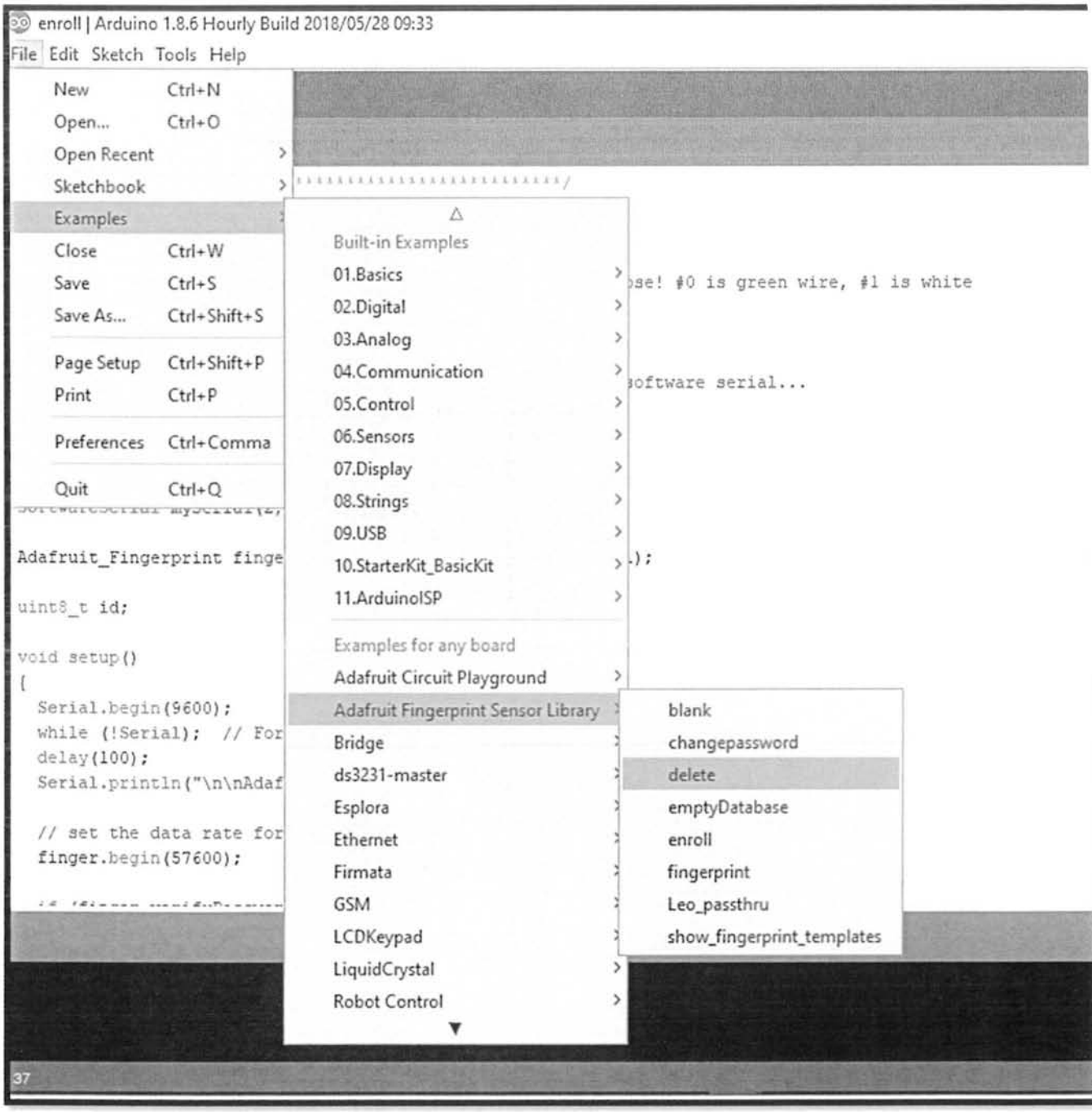

Fig 3. 6 Uploading the Code for Deleting the Fingerprint on the Sensor

• Check code for fingerprint matching:

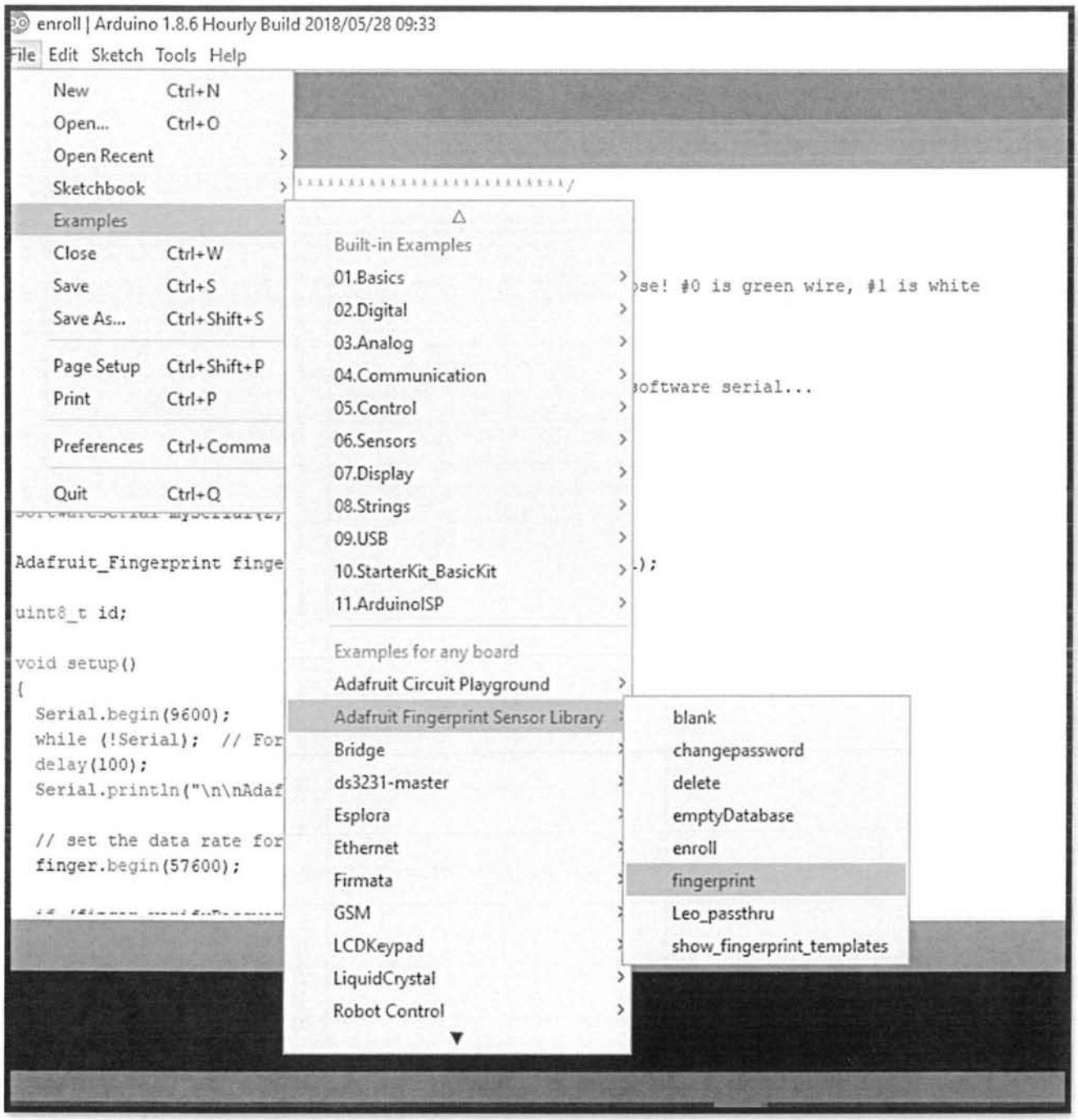

Fig 3. 7 Uploading the Code for Matching the Fingerprint on the Sensor

6. The Connections of LCD with Arduino

In this there are two connections for LCD. One is for LCD without keypad shield and the other is with keypad shield.

LCD without keypad shield

For this we just have to change the pin for Arduino like

- 11, 10, 9, 8 pins of Arduino are connected to D4, D5, D6, D7 pins of the LCD.
- 13 Pin of Arduino is connected to pin RS of the LCD.
- 12 Pin of Arduino is connected to pin enable of the LCD.

Code

```
#include<LiquidCrystal.h>                // including library for LCD

LiquidCrystal lcd(8,9,7,6,5,4);          // pin configuration of LCD
```

LCD with keypad shield

For this we just have to change the pin for Arduino like

- Pins 4,5,6,7 of Arduino has to be connected with the D4, D5, D6, and D7 of the LCD.
- Pin 8 of Arduino has to connect with the pin RS of the LCD.
- Pin 9 of Arduino has to connect with the pin enable of the LCD.
- Pin 10 of Arduino has to connect with the backlight of the LCD.

7. The connections of RTC with Arduino

RTC stands for real time clock it stores the time and date of the project.

Connections

- Pin A5 of Arduino is associated (connected) with SCL pin of the RTC module.
- Pin A4 of Arduino is associated (connected) with SCA pin of the RTC module.
- Pin ground of Arduino is associated (connected) with ground pin of the RTC module.
- Pin 5V of Arduino is to associated (connected) with the VCC pin of the RTC module.

Code:

After it, in loop function, we have read RTC time and displayed it on LCD

```
void loop()
{ now = rtc.now();
  lcd.setCursor(0,0);                  // cursor on 1st Colum and 1st row
  lcd.print("Time:");                  // Display time on LCD
  lcd.print(now.hour(), DEC);          // Set hour on RTC
  lcd.print(':');                      // display hour on LCD
  lcd.print(now.minute(), DEC);        // Set minute on RTC
  lcd.print(':');                      // Display minute on LCD
  lcd.print(now.second(), DEC);        // Set second on RTC
  lcd.print("    ");                   // Display second on LCD
  lcd.setCursor(0,1);                  // cursor on 1st Colum and 2nd row
  lcd.print("Date: ");                 // Display Date on LCD
  lcd.print(now.day(), DEC);           // Set date on RTC
  lcd.print('/');                      // Display Day on LCD
  lcd.print(now.month(), DEC);         // Set month on RTC
```

```
lcd.print('/');                    // Display month on LCD
lcd.print(now.year(), DEC);        // Set year on RTC
lcd.print("  ");                   // Display year on LCD
delay(1000);                       // wait for a second
```

8. The Connections of Buzzer with Arduino

- Pin 12 of Arduino is connected to pin positive of the buzzer.
- Pin ground of Arduino is connected to ground of the buzzer.

After it, we have to write code for downloading attendance data.

```
void setup()
{
  delay(1000);
  lcd.begin(16,2);                  // initializing the column and rows
  Serial.begin(9600);               // starting the serial monitor
  pinMode(enroll, INPUT_PULLUP);    // selecting the pin mode as input
  pinMode(up, INPUT_PULLUP);        // selecting the pin mode as input
  pinMode(down, INPUT_PULLUP);      // selecting the pin mode as input
  pinMode(dele, INPUT_PULLUP);      // selecting the pin mode as input
  pinMode(match, INPUT_PULLUP);     // selecting the pin mode as input
  pinMode(buzzer, OUTPUT);          // selecting the pin mode as output
  pinMode(finger, OUTPUT);          // selecting the pin mode as output
  digitalWrite(buzzer, LOW);        //initializing the buzzer as low
  if(digitalRead(enroll) == 0)
  {
```

```
digitalWrite(buzzer, HIGH);          // Turing on the buzzer
delay(500);                          // wait for half second
digitalWrite(buzzer, LOW);           // Turing off the buzzer
lcd.clear();                         // clearing the LCD
lcd.print("Please wait");
lcd.setCursor(0,1);                  // cursor on 1st Colum and 2nd row
lcd.print("Downloding Data");
```

Code for clearing attendance data from EEPROM.

```
if(digitalRead(dele) == 0)
{
 lcd.clear();
 lcd.print("Please Wait");
 lcd.setCursor(0,1);
 lcd.print("Reseting.....");
 for(int i=1000;i<1005;i++)  // count from 1 to 4
 EEPROM.write(i,0);          // storing in EEPROM
 for(int i=0;i<841;i++)
 EEPROM.write(i, 0xff);
 lcd.clear();
 lcd.print("System Reset");
 delay(1000);
```

Waiting for the finger print to take input and compare captured image ID with stored IDs. If a match occurs then proceed with next step. Check enroll delete keys as well

```
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("Welcome ");
        lcd.setCursor(0,1);
        lcd.print("You May Pass");
        delay(1000);
    digitalWrite(indFinger, HIGH);
    return;
 }
```

Storing attendance time and date in the allotted slot of EEPROM

```
void attendance(int id)
{
```

CHAPTER 4

Working

The working of biometric parking system is quite simple. Firstly, the fingerprints of new user is registered with the help of push buttons. For this purpose, user needs to press ENROLL key and then command appears on LCD to enter ID for the fingerprint, then fingerprints are saved in data memory by ID name.

For registration user needs to enter ID by using up/down keys afterwards a message on LCD is displayed in which user is asked to remove finger and place it again on fingerprint unit. The unit takes an image again and change it into templates and store it in finger print memory unit by ID of user. Now, the user is registered and can enter attendance by only placing it on fingerprint unit. By this small procedure all the users are registered in the system.

After registration user just needs to enter ID by using up/down keys. User needs to select ID, then press OK key (DEL key). Now a new message appears on LCD to place finger on the fingerprint module, then the unit takes image of the finger.

If the user wishes to remove or delete any of the stored ID or fingerprint, it is done by pressing DELETE key. For deleting the ID, the system asks the user to select the ID to be deleted and then press OK key (DEL key). Now message on LCD shows that fingerprint has been deleted effectively.

Here we have included an LED which demonstrates that fingerprint module is prepared to take a picture of the finger. A buzzer is additionally utilized for audio signal. Arduino is the fundamental part of this system, its function is to control the entire system

Along with the fingerprint unit, RTC module is utilized for date and time which is running continuously. Whenever a registered user places his/her finger on fingerprint sensor then Arduino records exact time and date of arrival of user and stores it in its EEPROM (memory).

Downloading Data

Here we have made 4 user space in this project for 30 days. When reset and enroll buttons are pressed simultaneously the system will start downloading data over serial monitor from the Arduino EEPROM memory. It looks like as

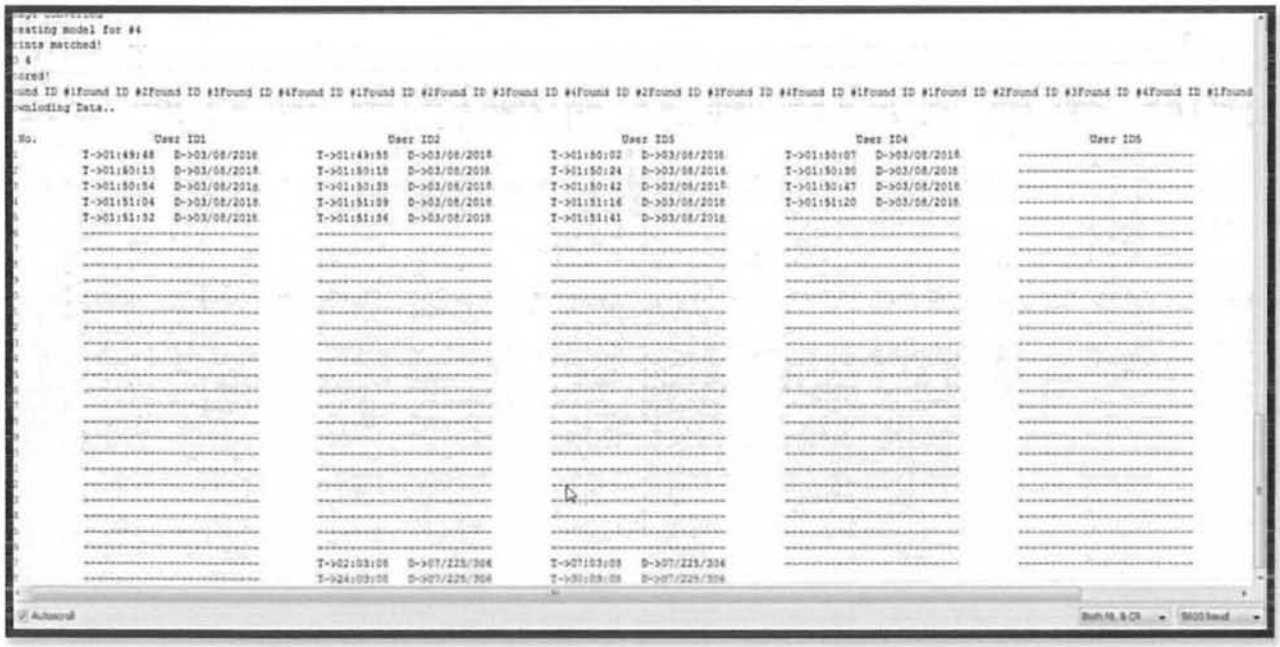

Fig 4. 1 Downloading Data over Serial Monitor to Check Time and Date of Each User

User ID Records:

Fig 4. 2 Record of Each ID

There is record for four user and for all the id's.

Clearing Downloaded Data

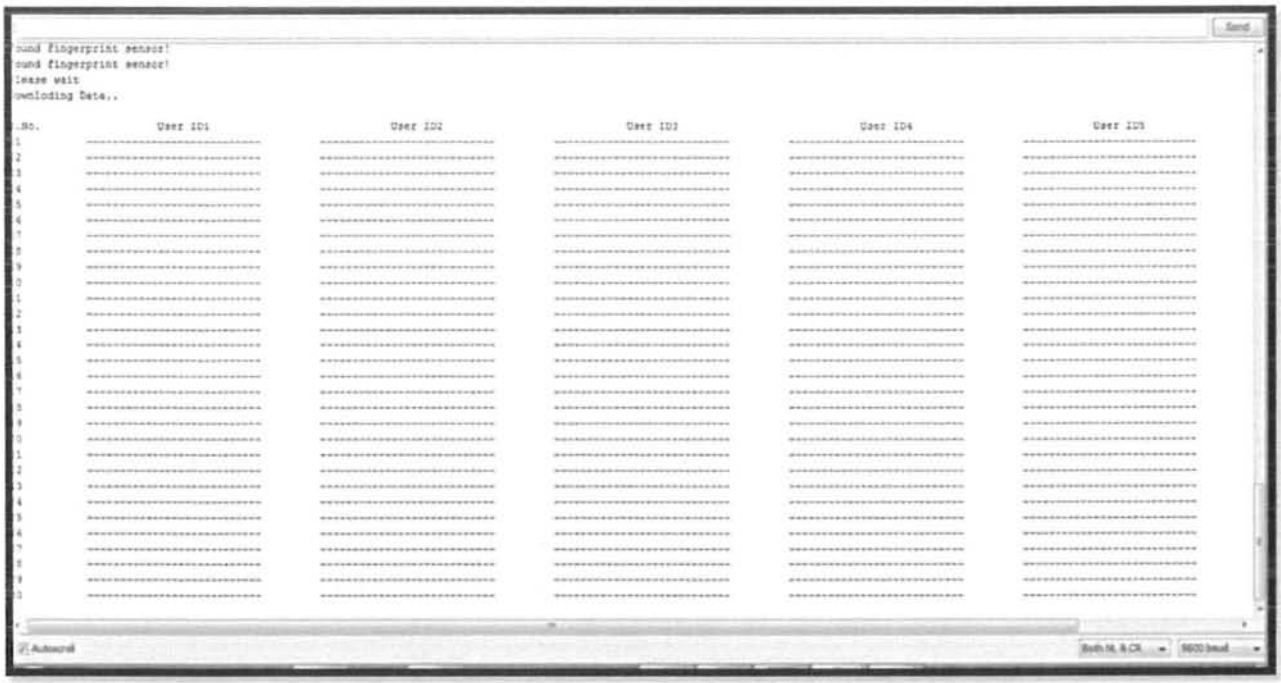

Fig 4. 3 Clearing Data from Serial Monitor

Memory management

- Total 1023 byte memory in Arduino UNO.
- We have recorded the data of 4 user for 30 days which utilize 1018 byte memory in Arduino UNO.
- It records the time and date which requires 7 Byte.

CHAPTER 5

Conclusion and Recommendation

Conclusion

The thought of biometric security system for car parking, is as fingerprints are specific for every individual hence there are minimum possibilities of security compromise. In this project we have designed biometric parking system which can serve two functions at the same time; security and attendance record. System is created using Arduino UNO, fingerprint sensor, RTC module, BUZZER, LED, LCD and Breadboard. The project has been successfully implemented and demonstration has been given in depth. The problem which is faced in the use of Arduino UNO is that, its memory is small which cannot store large record of users. This problem may be resolved by use of Arduino Mega in future projects.

Recommendation for Future Work

The security and compliance of users may be improved by adding camera or retinal scan as security control. Retinal scanner or camera for face detection can be placed at face level, which will be convenient for user, as there will be no need to come out of the car or take out hand to swap the finger on finger print sensor to ensure the presence.

References:

http://www.hobbytronics.co.uk/arduino-uno-r3

http://www.circuitstoday.com/working-of-fingerprint-scanner-2

http://www.electronics-lab.com/project/ds1307-rtc-module/