BIOMETRIC ACCESS FOR SECURE SMART ROBOT

A PROJECT REPORT

Submitted in partial fulfillment for the award of the degree of

B.TECH

in

Information Technology

by

NIKHIL KESHRI	**13BIT0004**
SHUBHAM PAMECHA	**13BIT0057**
RAVEESH KHATTAR	**13BIT0140**

Under the Guidance of
Prof. RAGHAVAN R

VIT UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

School of Information Technology & Engineering

MAY 2017

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled **"BIOMETRIC ACCESS FOR SECURE SMART ROBOT"** submitted by me to VIT University, Vellore in partial fulfillment of the requirement for the award of the degree of **B.Tech.(Information Technology)** is a record of bonafide project work carried out by me under the guidance of **Prof. RAGHAVAN R**. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Signature of the Candidate

Date:8TH MAY 2017

NIKHIL KESHRI
SHUBHAM PAMECHA
RAVEESH KHATTAR

School of Information Technology & Engineering [SITE]

CERTIFICATE

This is to certify that the project report entitled "**BIOMETRIC ACCESS FOR SECURE SMART ROBOT"** submitted by **NIKHIL KESHRI (13BIT0004) , SHUBHAM PAMECHA (13BIT0057) , RAVEESH KHATTAR (13BIT0140)** to VIT University, Vellore, in partial fulfillment of the requirement for the award of the degree of **B.Tech** in **Information Technology** is a record of bonafide work carried out by him/her under my guidance. The project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature
Prof. Dr. M. Dinakaran ,Program Chair
Prof. Raghavan R, Internal Guide

The project work is satisfactory / unsatisfactory

Internal Examiner External Examine

CERTIFICATE BY THE EXTERNAL GUIDE

This is to certify that the project report entitled "**BIOMETRIC ACCESS FOR SECURE SMART ROBOT** " submitted by **NIKHIL KESHRI (13BIT0004) , SHUBHAM PAMECHA (13BIT0057) ,RAVEESH KHATTAR (13BIT0140)** to VIT University, Vellore in partial fulfillment of the requirement for the award of the degree of **BTECH** in **INFORMATION TECHNOLOGY** is a record of bonafide work carried out by him/her under my guidance. The project fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

EXTERNAL SUPERVISOR

ACKNOWLEDGEMENT

I wish to express my heartfelt gratitude to **Dr.G.Viswanathan**, Chancellor, VIT University, Vellore for providing facilities for the final year capstone project. I am highly grateful to our Vice Presidents, **Shri. Sankar Viswanathan, Dr. Sekar Viswanathan** and **Shri. G.V.Selvam**, Vice Chancellor **Dr. Anand A. Samuel**, Pro-Vice Chancellors **Dr. S. Narayanan** and **Dr. V. Raju** for providing necessary resources.

My sincere gratitude to **Dr. Aswani Kumar Cherukuri**, Dean, School of Information Technology and Engineering for giving us the opportunity to undertake the project.

I wish to express my sincere gratitude to **Dr. M.Dinakaran**, Associate Professor, Head of Department of Information Technology and the Project Coordinator **Prof. Thippa Reddy G**, Assistant Professor (Senior), School of Information Technology and Engineering for providing me an opportunity to do my project work in the **VIT University.**

I would like to express my special gratitude and thanks to my internal guide **Prof. Raghavan R,** Senior Professor, School of Information Technology and Engineering whose esteemed guidance and immense support encouraged to complete the project successfully.

I thank the Management of VIT University for permitting me to use the library resources. I also thank all the faculty members of VIT University for giving me the courage and strength I needed to complete my goals. This acknowledgement would be incomplete without expressing my whole hearted thanks to my family and friends who motivated me during the course of the work.

Nikhil Keshri

Place : Vellore

Shubham Pamecha

Date : 8th May'17

Raveesh Khattar

Executive Summary

Biometrics, an important research field emerging fast, has growing challenges related to the secured access and security of the systems protected. In this paper, the secured access to a smart robot is proposed using biometrics. The robot can be accessed by recognizing the user's finger print, acting as a password, and also operates on the voice-commands given by the speaker. The voice-commands are processed in real-time. The commands are processed in real-time using an on-line cloud server. The voice-commands converted to text form are sent via the Bluetooth network to a micro-controller on the robot, where these are interpreted and the smart robot performs the desired operation accordingly. The performance evaluation results of the finger print scanner and the voice commands signal processing are encouraging. Combining both the features makes the robot more robust.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

API	Application Program Interface
IDE	Integrated Development Environment
LED	Light Emitting Diode
TX	Transmission
RX	Receiver

TABLE OF CONTENTS

1.Introduction

In this report, as a part of our final year project, we present smart robot which is accessed using user voice using voice recognition technique. It is also secured by biometric fingerprint scanner. In this chapter, we will give a brief description of this thesis.

1.1 Background

Biometrics, an important research field emerging fast, has growing challenges related to the secured access and security of the systems protected. In this paper, the secured access to a smart robot is proposed using biometrics. The robot can be accessed by recognizing the user's finger print, acting as a password, and also operates on the voice-commands given by the speaker. The voice-commands are processed in real-time. The commands are processed in real-time using an on-line cloud server. The performance evaluation results of the finger print scanner and the voice commands signal processing are encouraging. Combining both the features makes the robot more robust.

1.2 Problem Statement

The project aims in designing a smart robot which is accessed using user voice using voice recognition technique which is secured using biometric fingerprint scanner. It can be used in many of the automobiles which will be controlled through our voice .It can be a very low cost system used to monitor automobiles .

1.3 Importance

Even though, the use of possible 'Smart Robots' is on the rise, there is a concern over the possible misuse of these robots . 'Smart Robots' deployed at a larger scale for the applications such as military operations or in case of unmanned tracking can be easily exploited if hijacked and not handled with proper care and security. Hence, security in terms of recognizing the user, can be very beneficial in reducing such abuse. Biometrics can be a key is such scenarios, to provide a secured access to these Smart Robots. Biometrics examples are fingerprint , iris, texture, DNA, ear analysis. Use of such biometric techniques can possible provide such secured access, where this access can be given to a group of selected people, in order to minimize the mis-utilization of these precious resources

1.4Organisation of the Report

The complete report of this proposed system have been organised in seven chapters. Chapter 1 mainly deals with the introduction part. In this chapter we have explained the main aims and objectives of the system along with the background and motivation for this project. The problem statement is also described in this chapter. Chapter 2 deals with the theoretical background and the literature review part. It explains the theoretical background and technologies that will be used in this proposed system. It gives a brief description about various network technologies. Chapter 3 presents the hardware and software requirement of the system. The main hardware with the pin diagrams have been described in this system together with the software requirements. Chapter 4 explains the main system architecture and structure. It explains about the database used, sensor data transmission etc. Chapter 5 demonstrates the testing and validation part. Various test cases have been designed in order to test and validate the system. Chapter 6 shows the discussions related to this system. It explains how the data is processed. Chapter 7 is the last chapter and it explains the main conclusion and the future work that can be done in this system.

2. OVERVIEW AND PLANNING

In this chapter we discuss about the planning which we went through in making this project. It explains the system overview and assumptions made in this proposed system. It gives a brief description about various hardware and software requirements.

2.1 Proposed System Overview

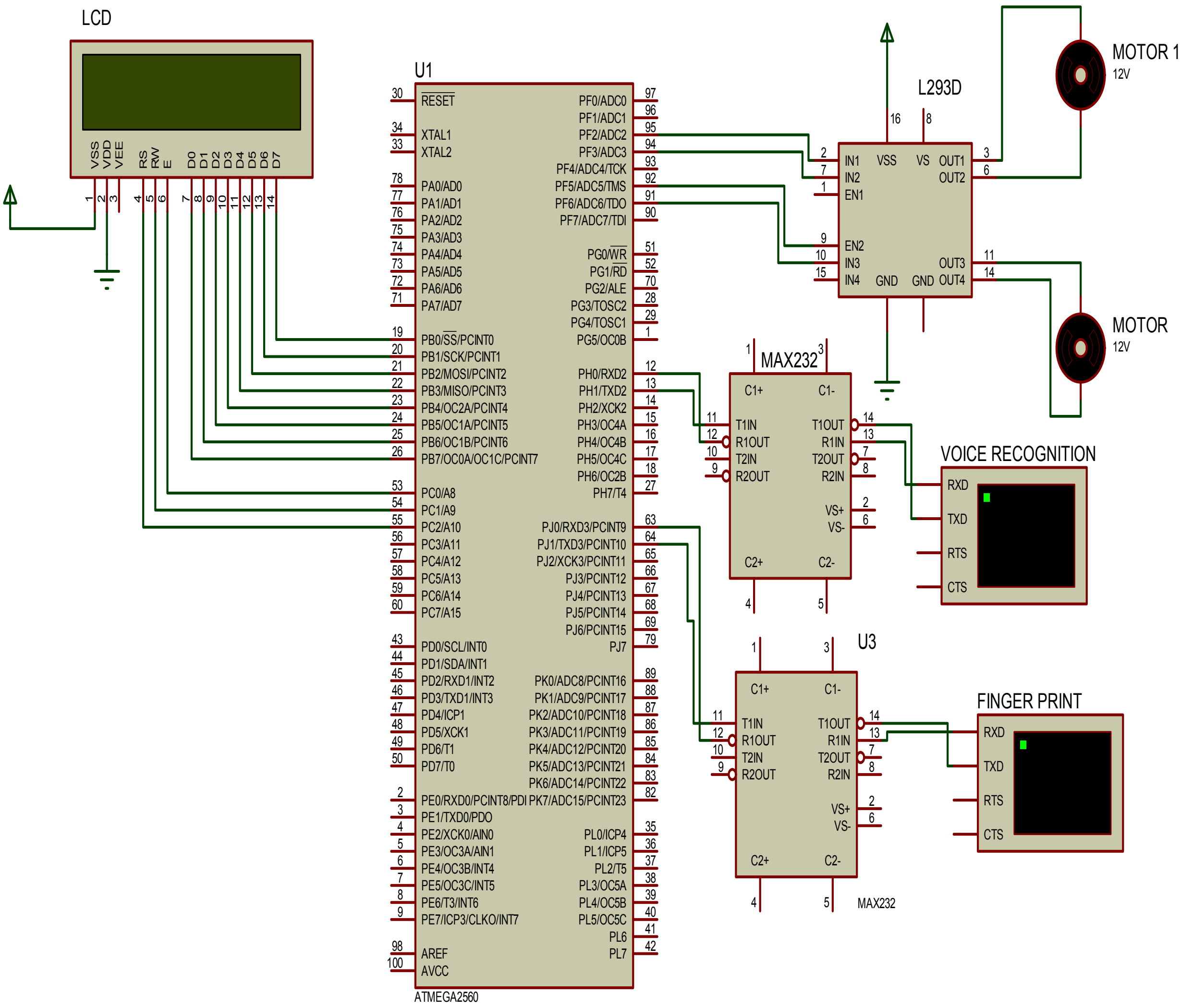


Fig 2.1.1 Proposed System Design

This diagram outlines how our various assets communicate with each other. Arduino Mega is in the middle show how all the components are connected to its different pins

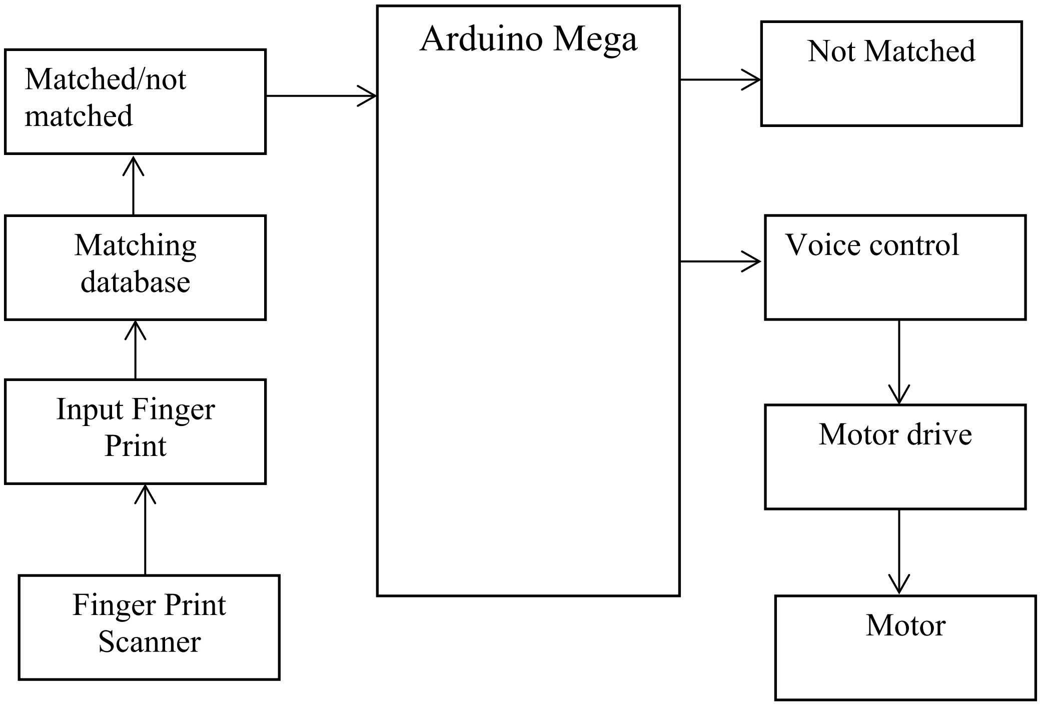
Fingerprint scanner, voice recognition module , motor and LCD all are connected to this Arduino Mega to different pins and sending signals to each other accordingly.

2.2 Assumptions

- The Arduino Mega is used to control the operations. It will process the data which are collected by different sensors thus the results will be easy and clear for reading and understanding.
- All the sensors used are of small scale and cannot be directly put to test in real life scenario.
- The project has been made on a prototype model and can be further extended to a large scale when put to use in real life scenario.

2.3 Architecture

Fig 2.3.1 System Architecture



The figure gives an overview of the reference architecture. The architecture is decomposed into a number of modules with core functionalities. These modules are described in more detailed in other section. Several of these modules need to be implemented in devices while others on a breadboard or the board. The fingerprint scanner and the voice recognition module are separate components and will be completely implemented on a board. Furthermore, we also distinguish different modules based on their importance.

2.4 Hardware Requirements

- Arduino Mega
- Voice control unit
- Finger Print Scanner
- Motor Drive
- Motor
- Arduino LCD Board
- Battery

2.5 Software Requirements

- Arduino IDE

2.6 Project Schedule

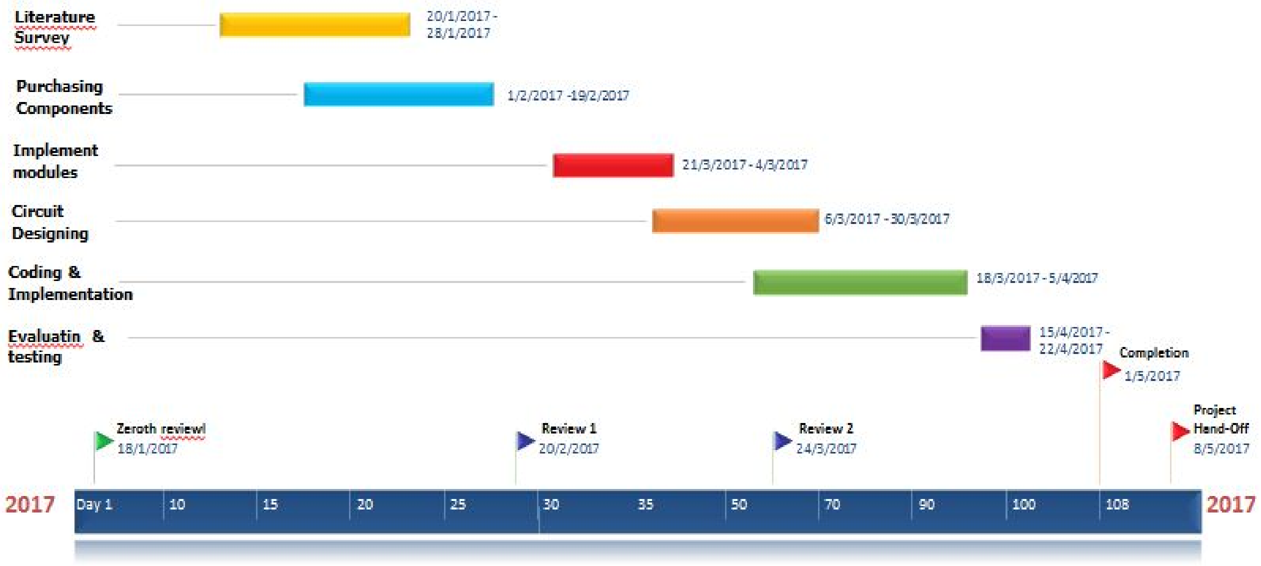


Fig 2.6.1 Gantt Chart

2.8 Work Breakdown Structure

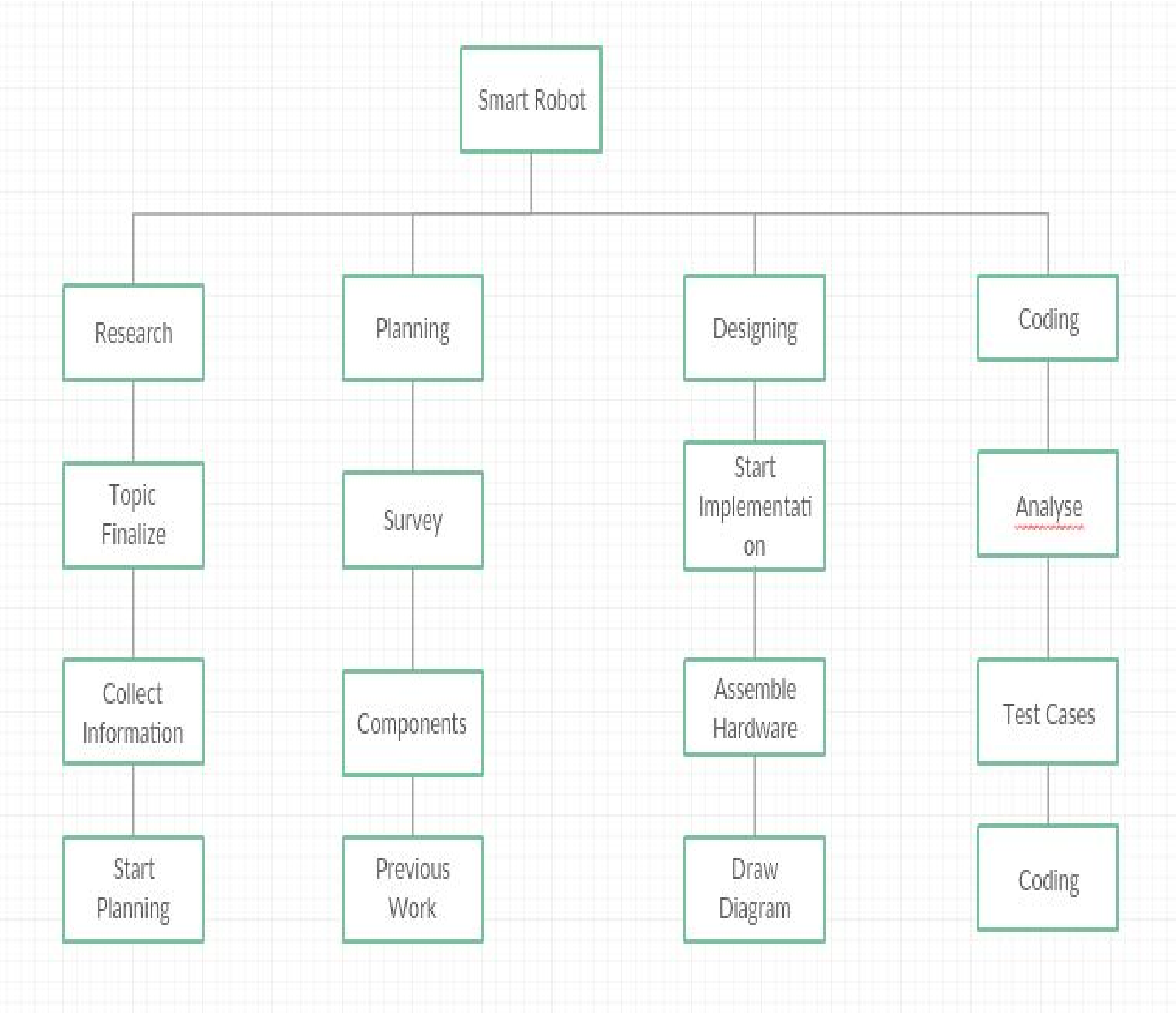


Fig 2.8.1 Work Breakdown Structure

3. Literature Survey and review

3.1 Literature Survey

Biometric access for secure smart robot

Author :- Kanv Kumar Indian Institute of Information Technology, Chittoor, Sri City, A.P, India

Kishan Kumar Indian Institute of Information Technology, Chittoor, Sri City, A.P, India

Shyam Nandan Indian Institute of Information Technology, Chittoor, Sri City, A.P, India

Abstract

Biometrics, an important research field emerging fast, has growing challenges related to the secured access and security of the systems protected. In this paper, the secured access to a smart robot is proposed using biometrics. The robot can be accessed by recognizing the user's face image, acting as a password, and also operates on the voice-commands given by the speaker. The face recognition technique uses `Principal Component Analysis' and `Eigenfaces'. The voice-commands are processed in real-time. The commands are processed in real-time using an on-line cloud server. The voice-commands converted to text form are sent via the Bluetooth network to a micro-controller on the robot, where these are interpreted and the smart robot performs the desired operation accordingly. The performance evaluation results of the face image recognition and the voice-commands signal processing are encouraging. Combining both the features makes the robot more robust. Few limitations of both the techniques have also been discussed with proposed solutions.

3.2 Literature Summary

The wide range of applications of robots and the field robotics include industrial purposes, military applications and household activities etc [1]. Even though, the use of possible 'Smart Robots' is on the rise, there is a concern over the possible misuse of these robots. ‘Smart Robots’ deployed at a larger scale for the applications such as military operations or in case of unmanned tracking can be easily exploited if hijacked and not handled with proper care and security. Hence, security in terms of recognizing the user, can be very beneficial in reducing such abuse. Biometrics can be a key is such scenarios, to provide a secured access to these Smart Robots. Biometrics examples are fingerprint, iris, texture, DNA, ear analysis. Use of such biometric techniques can possible provide such secured access, where this access can be given to a group of selected people, in order to minimize the mis-utilization of these precious resources [2].Computer Vision Department of the University of Oulu is working on Robust bi-modal biometrics (face and skin texture) under spoofing attacks [3]. The aim is to study, develop and evaluate novel audio-visual biometric authentication solutions to enhance information security, especially under spoofing attacks. Clemson University’s Biometric and Pattern Recognition Lab is working on finger print scanner [4]. The three ongoing projects in this lab are: (a) Periocular Based Biometric Recognition, (b) Aging effects on Facial Recognition and (c) Forensic Detection [4].

4.System Design

In this chapter we discuss about the architecture of our system, the design specifications of our model, the configuration settings of our modules, the circuit design of remote sensor boards with respect to Arduino Uno.

4.1 High Level Design

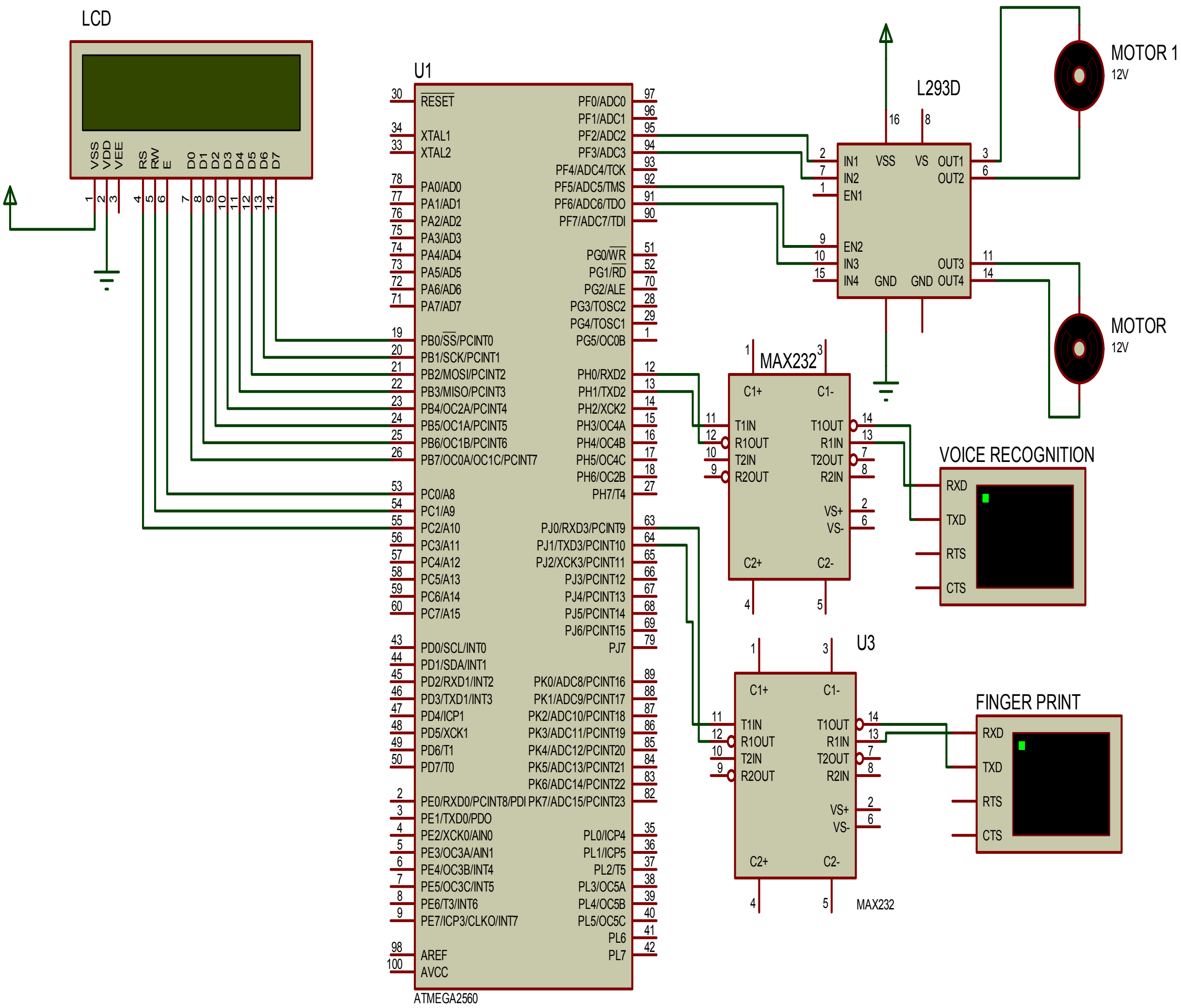


Fig 4.1.1 Diagram of different parts in the system

4.2 Low Level Design

4.2.1 Module 1:Arduino Mega

Mega2560-CORE is a small, complete and breadboard-friendly board base on the ATMega2560. Its design is based on the Arduino Mega2560, so we can use it as a Arduino Mega2560 development board. In a different place, it lacks only a 6-foot download port and a reset switch. Reducing the hardware circuit that can we reduce the power consumption and the cost. Mega2560-CORE has a matching download line and the other one end of the download cable is a USB interface, so it is very convenient for use. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. PRODUCT DESCRIPTION The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 (datasheet). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

The Arduino Mega is a microcontroller board based on the ATmega1280 . It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Microcontroller	ATmega1280
Operating Voltage	5V

Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

FEATURES

- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V
- Digital I/O Pins: 54 (of which 14 provide PWM output)
- Analog Input Pins: 16
- DC Current per I/O Pin: 40 mA

APPLICATIONS

- Multiple projects requires high speed operations

- Multiple DIY projects having larger code size
- Atmega32 based robot
- Real time biometrics

Fig 4.2.1.1 Arduino Mega

4.2.2 Module 2: VOICE RECOGNITION BOARD DESCRIPTION

The voice recognition board is a completely assembled and easy to use programmable …speech recognition circuit. The commands that you need can be programmed to recognize. It has 8 bit data out which can be interfaced with 16-bit PIC microcontroller . The audio input from the microphone can be given through the audio jack assembled in this board . The input command with their corresponding characters can be displayed in LCD.

PRODUCT DESCRIPTION

Speech samples are acquired by a Microphone on board .This board analyzes the analog signal received compares with the data stored in external RAM and finally outputs a corresponding 8 bit Data . This 8 bit data can be directly connected to a port of microcontroller for further action .The board requires 5v DC supply.

A 12v adapter can be used as a power source , as the board has in built 5v regulator with heat sink . This board requires initial configuration or training of words . During training process user trains the IC by speaking words into the microphone and assigning a particular value for thatword.

Fig 4.2.2.1 Voice Recognition Board

FEATURES:-

☐Supply voltage : 12v DC

☐Output : status 2x16 LCD display

☐Recognize upto 15 predefined voices

☐Compact size

APPLICATIONS

☐Voice recognition system

☐Industrial applications

4.2.3 Module 3: Finger Print Scanner

GENERAL DESCRIPTION

SM-621 is RS232 /UART fingerprint module scanner for the demand of access control system, door lock, T&A and safety box OEM POS Consisting of high function DSP, large capacity FLASH and color CMOS, etc, SM621 optical fingerprint module can conduct fingerprint enrollment, image processing, templates storage, fingerprint matching and fingerprint searching. This Optical biometric fingerprint reader is with great features and can be embedded into a variety of end products, such as: access control, attendance, safety deposit box, car door locks.

PRODUCT DESCRIPTION

R305 fingerprint module is fingerprint sensor with TTL UART interface for direct connections to microcontroller UART or to PC through MAX232 / USBSerial 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 FP module can directly interface with 3v3 or 5v Microcontroller. A level converter (like MAX232) is required for interfacing with PC serial port

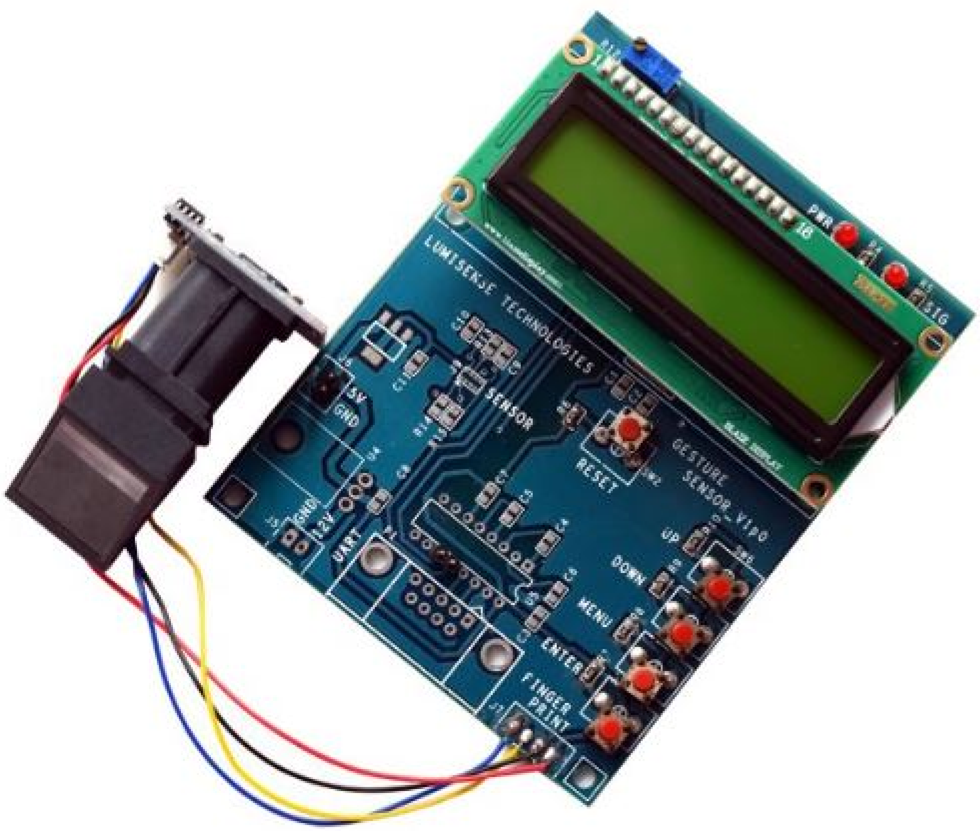


Fig 4.2.3.1 Fingerprint Scanner

FEATURES

- Input voltage: 5v
- Interface: RS232.
- Matching Mode: 1:1 and 1:N.
- Baud rate: 9600 – 115200.
- Storage Capacity: 256.

APPLICATIONS

- Attendance system.
- Safety deposit box system.
- Car door locking system

4.2.4 Module 4 : Arduino LCD Board

GENERAL DESCRIPTION

LCD board is specially designed for interfacing with Arduino. This board consists of two sections namely data and control sections. This board is provided with a potentiometer to adjust the LCD. The supply voltage to the board is given from the controller or from a separate source.

PRODUCT DESCRIPTION

The communication extension board provides a set of communication features and pin outs as well as an external power connector. The LCD-Board is ideal as an extension board for the Hitex LPC2478-Stick or the Hitex LPC3250- Stick. LCD module [STM32F4DIS-LCD] - consists of 3.5 inch LCD and driver board. This module is designed for Embest STM32F4DIS-BB board.

Fig 4.2.4.1 Arduino Display Board

FEATURES

- Supply voltage: 5VDC
- Separate data lines
- Separate control lines

APPLICATONS

- Display applications for Arduino and other controller•

4.2.5 Module 5 : DC Motor

A DC motor is designed to run on DC electric power. Two examples of pure DC designs are Michael Faraday's homopolar motor (which is uncommon), and the ball bearing motor, which is (so far) a novelty.

By far the most common DC motor types are the brushed and brushless types, which use internal and external commutation respectively to create an oscillating AC current from the DC source—so they are not purely DC machines in a strict sense.

We in our project are using brushed DC Motor, which will operate in the ratings of 12v DC 0.6A which will drive the flywheels in order to make the robot move.

Fig 4.2.5.1 DC MOTOR

4.2.6 Module 6 : Battery 12v

GENERAL DESCRIPTION

These batteries have internal electronics to allow them to be used as a dropin replacement for the equivalent sealed lead acid battery. They can be charged with lead acid chargers, and the 12.8 volt potential is very close to the 12.9V open circuit voltage of the lead acid chemistry. To be even more cost-effective, LiFeP batteries are among the longest lived batteries ever developed. Test data in the laboratory show up to 2000 charge/discharge cycles. Our cells typically have more than 1000 cycles in service. These batteries are excellent for use in motorcycles, snowmobiles, jet skis, etc., because they do not need maintenance during the off season.

PRODUCT DESCRIPTION

Batteries can be charged manually with a power supply featuring user-adjustable voltage and current limiting. 12v 7.5Ah lead acid battery is a rechargeable battery that supplies electrical energy. These batteries are designed to release a high burst of current and then quickly recharged. Six cells are connected in series in this battery. Another advantage is that running the battery completely down (i.e. leaving the parking light on for a week) will not damage the battery at all, unlike the SLA and Gel batteries, since it has a low-voltage disconnect circuit.

FEATURES

- Output voltage: (12-12.6) v DC
- Current capacity: 7.5Ah
- Low self discharge
- Long life
- 7.5AH Large lithium iron phosphate rechargeable battery

APPLICATIONS

- Automobiles
- UPS
- Perfect for motorcycle starter, 12VDC power, medical equipment, LED light
- Power supply, solar power This is plenty of power to start most small engines.

Fig 4.2.6.1 Battery

4.3 Code and Standards

4.3.1 Enroll code

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
uint8_t id;
uint8_t getFingerprintEnroll();
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
void setup()
{
  while (!Serial);
 delay(500);
  Serial.begin(9600);
  Serial.println("Adafruit Fingerprint sensor enrollment");
  finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1);
  }
}
uint8_t readnumber(void) {
uint8_t num = 0;
boolean validnum = false;
  while (1) {
    while (! Serial.available());
    char c = Serial.read();
    if (isdigit(c)) {
       num *= 10;
```

```
      num += c - '0';
      validnum = true;
    } else if (validnum) {
      return num;
    }
  }
}
void loop()
{
  Serial.println("Ready to enroll a fingerprint! Please Type in the ID # you want to save this finger as...");
  id = readnumber();
  Serial.print("Enrolling ID #");
  Serial.println(id);
    while (!  getFingerprintEnroll() );
}
uint8_t getFingerprintEnroll() {
  int p = -1;
  Serial.print("Waiting for valid finger to enroll as #");
Serial.println(id);
  while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println(".");
      break;
    case FINGERPRINT_PACKETRECIEVEERR:
```

```
    Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
  default:
    Serial.println("Unknown error");
    break;
  }
}
p = finger.image2Tz(1);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
```

```
}
  Serial.println("Remove finger");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image taken");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.print(".");
    break;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
  default:
    Serial.println("Unknown error");
    break;
  }
}
```

```
p = finger.image2Tz(2);
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:
      Serial.println("Image too messy");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
      Serial.println("Could not find fingerprint features");
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      Serial.println("Could not find fingerprint features");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }
  Serial.print("Creating model for #");  Serial.println(id);
    p = finger.createModel();
  if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
```

```
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }
  Serial.print("ID "); Serial.println(id);
  p = finger.storeModel(id);
  if (p == FINGERPRINT_OK) {
    Serial.println("Stored!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
  } else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }
}
```

4.3.2 Fingerprint code

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
int getFingerprintIDez();
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
void setup()
{
  while (!Serial);  // For Yun/Leo/Micro/Zero/...
    Serial.begin(9600);
  Serial.println("Adafruit finger detect test");
finger.begin(57600);
    if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1);
  }
  Serial.println("Waiting for valid finger...");
}
void loop(){
  getFingerprintIDez();
  delay(50);
 }
uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
```

```
    break;
  case FINGERPRINT_NOFINGER:
    Serial.println("No finger detected");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    return p;
  default:
    Serial.println("Unknown error");
    return p;
}
p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
  case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
```

```
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }
  Serial.print("Found ID "); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
}
int getFingerprintIDez() {
  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.println(finger.fingerID);
  return finger.fingerID;
```

```
}
```

4.3.3 Motor Drive code:

```
#include <NewPing.h>
#include <LiquidCrystal.h>
#define PING_PIN  13
#define PING_MAX_DISTANCE 200 /
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
 NewPing sonar(PING_PIN, PING_PIN, PING_MAX_DISTANCE);
#define MAX_DISTANCE_FROM_OBSTACLE_IN_CMS 100
#define BUFFSIZ 90
char buffer[BUFFSIZ];
char *parseptr;
char buffidx;
int row = 0;
char tagOk[] ="FRON";
char tagOk1[] ="LEFT";
char tagOk2[]="RIGH";
char tagOk3[]="STOP";
char msg = ' ';
int count = 0;
char inputTag[04];
uint8_t hour, minute, second, year, month, date;
uint32_t latitude, longitude;
uint8_t groundspeed, trackangle;
char latdir, longdir;
char status;
int motor_left[] = {6, 7};
int motor_right[] = {8, 9};
int mine=A1;
```

```
int motor=23;
int pir=22;
int temp1;
float temp2;
int temp3;
void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
   while (!Serial) {
    ;
}
  for(int i = 0; i < 2; i++){
  pinMode(motor_left[i], OUTPUT);
  pinMode(motor_right[i], OUTPUT);
  }
  pinMode(pir,INPUT);
  pinMode(mine,INPUT);
  pinMode(motor,OUTPUT);
Serial.println("GPS get start");
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("VECHICLE SECURITY");
  lcd.setCursor(0, 1);
  lcd.print("SYSTEM");
  delay(3000);
  lcd.clear();
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
   lcd.print("WAITING FOR");
  lcd.setCursor(0, 1);
```

```
lcd.print("FINGER");
delay(3000);
l1:if (Serial.available()) {
  char inChar = (char)Serial.read();
  if(inChar=='0'){
     lcd.clear();
     lcd.print("FINGER NOT VERIFIED");
     Serial.println("FINGER NOT VERIFIED");
   delay(3000)
  }
  else if(inChar=='1'){
   delay(1000);
delay(50);
unsigned int distanceInCms = sonar.ping_cm();
if(distanceInCms != NO_ECHO && distanceInCms <
MAX_DISTANCE_FROM_OBSTACLE_IN_CMS) {
}  else  {};
lcd.clear();
lcd.setCursor(0, 1);
lcd.print("FINGER VERIFIED");
Serial.println("FINGER VERIFIED");
digitalWrite(motor,HIGH);
delay(1000);
lcd.clear();
delay(100);
lcd.setCursor(0, 0);
lcd.print("WAITING FOR");
lcd.setCursor(0, 1);
lcd.print("VOICE");
delay(3000);
```

```
 }
  }
 else
 {
  goto l1;
 }
}

void loop() {
if(Serial1.available())
  {
   msg=(Serial1.read());
if(msg=='7')
 {
    lcd.setCursor(0, 0);   lcd.print("FORWARD       ");
 drive_forward();
lcd.clear();
  }
 if(msg=='8')  {
    lcd.setCursor(0, 0);
   lcd.print("LEFT      ");
 turn_left();
}
   if(msg=='B')
 {
    lcd.setCursor(0, 0);
   lcd.print("RIGHT    ");
 turn_right();
}
   if(msg=='F')
```

```
 {
    lcd.setCursor(0, 0);
   lcd.print("STOP   ");
 motor_stop();
}
void drive_forward() {
  digitalWrite(motor_left[0], HIGH);
  digitalWrite(motor_left[1], LOW);
  digitalWrite(motor_right[0], HIGH);
  digitalWrite(motor_right[1], LOW);
}
void drive_backward() {
  digitalWrite(motor_left[0], LOW);
  digitalWrite(motor_left[1], HIGH);
  digitalWrite(motor_right[0], LOW);
  digitalWrite(motor_right[1], HIGH);
}
void turn_left() {
  digitalWrite(motor_left[0], LOW);
  digitalWrite(motor_left[1], HIGH);
  digitalWrite(motor_right[0], HIGH);
  digitalWrite(motor_right[1], LOW);
}
void turn_right() {
  digitalWrite(motor_left[0], HIGH);
  digitalWrite(motor_left[1], LOW);
    digitalWrite(motor_right[0], LOW);
  digitalWrite(motor_right[1], HIGH);
}
void motor_stop(){
```

```
  digitalWrite(motor_left[0], LOW);
  digitalWrite(motor_left[1], LOW);
  digitalWrite(motor_right[0], LOW);
  digitalWrite(motor_right[1], LOW);
}
```

4.4 Constraints and Tradeoffs

- ☐ The application used is compatible enough for small test processes but it can't be directly put into deployment where it can be over fed with huge volume of data.
- ☐ Fingerprint scanner will react gently to the finger print of the user and then the user will be able to access smart robot.

4.5 Test Case Generation

Various different possible situations are developed that can happen in a situation. It includes general basic scenarios that we see on a normal basis. In addition to that, most of the extreme cases have also been taken into consideration which should also be handled by the system.

5.System Design

In this chapter we present the testing results of our designed robotic boat in three different scenarios which are shown below:

1. To connect Arduino Mega to gateway PC through USB.

2. To process and run each module separately.

3. To integrate all the modules and test altogether.

For the first test, Arduino Mega is connected to the adaptor. The results show that system requirements are successfully achieved. The second scenario is the most important one as each and every component is successfully tested in a real condition experiment. In the third scenario this scenario all the modules are integrated and are tested. The result shows that all modules are integrated successfully.
The first two scenarios describe the Unit Testing whereas the third scenario describes the Integration Testing.

5.1 Unit Testing

5.1.1 Fingerprint Scanner

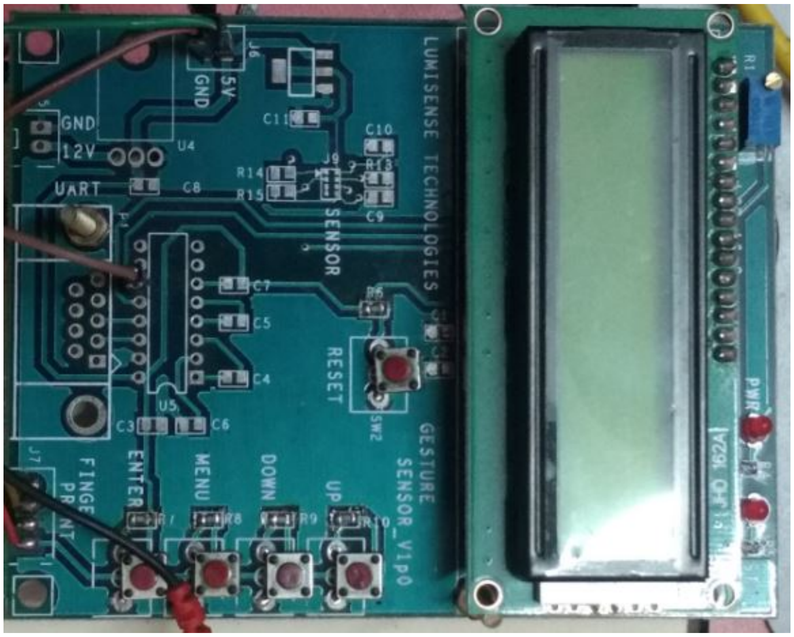


Fig 5.1.1.1: Initial Module of Fingerprint Scanner

Fig 5.1.1.2 Fingerprint verified

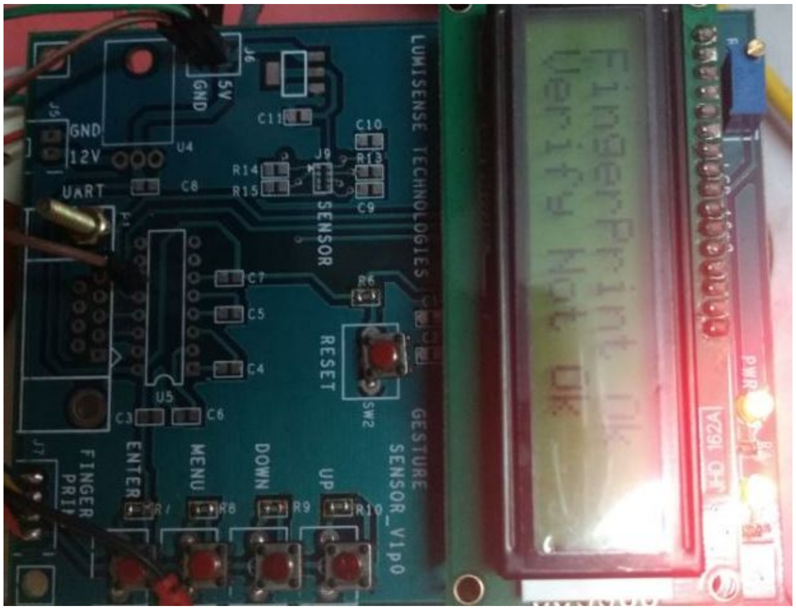


Fig 5.1.1.3Fingerprint not verified

5.1.2 Arduino Mega

Fig 5.1.2.1: Arduino Mega when power is off

Fig 5.1.2.2: Arduino Mega when power is on

5.1.3 Display Board

Fig 5.1.3.1 Display Board Off

Fig 5.1.3.2Display Board displaying message vechicle security system

Fig 5.1.3.3Display Board displaying message waiting for finger

Fig 5.1.3.4Display Board displaying message waiting for voice

Fig 5.1.3.5Display Board displaying message to move right

Fig 5.1.3.6Display Board displaying message to move left

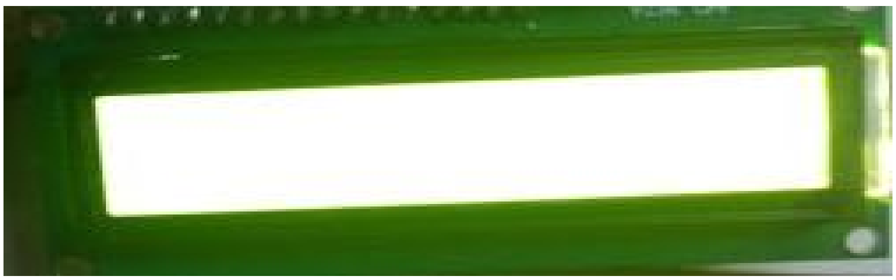


Fig 5.1.3.7Display Board displaying message to move forward

Fig 5.1.3.8Display Board displaying message to stop

5.1.4 Voice Recognition Board

Fig 5.1.4.1 Voice Recognition Board off

Fig 5.1.4.2 Voice Recognition Board on

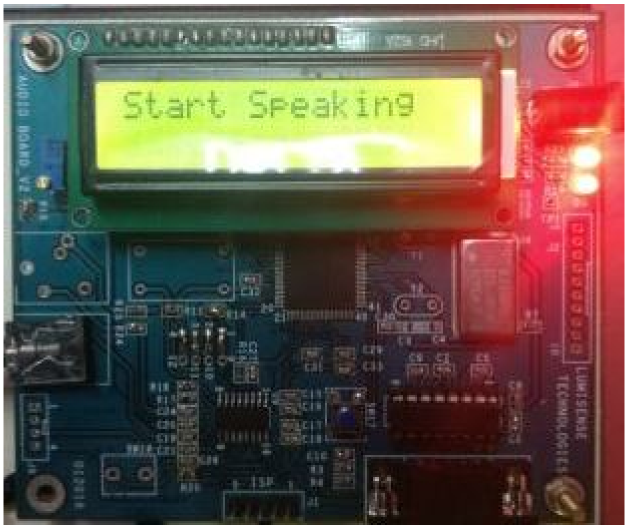


Fig 5.1.4.3 Voice Recognition Board messages to start speaking

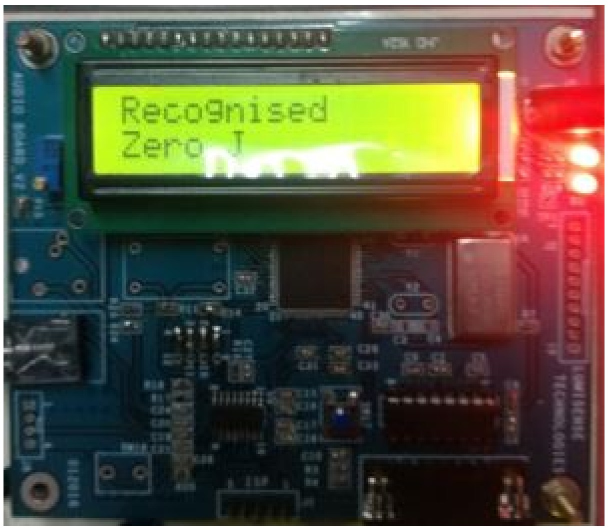
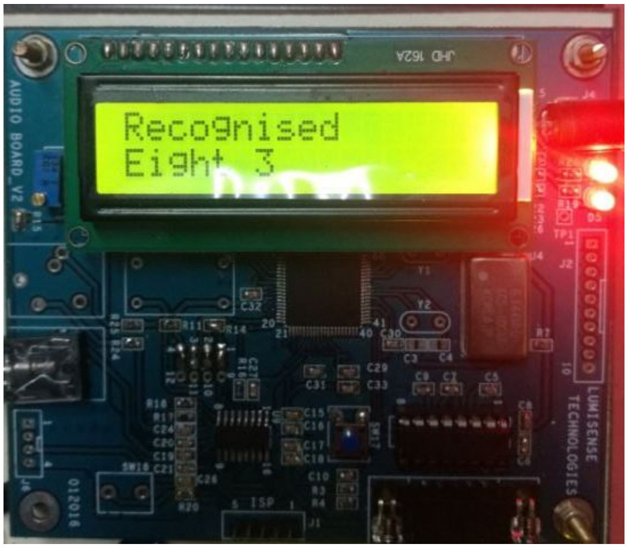


Fig 5.1.4.4 Voice Recognition Board messages zero ie to stop

Fig 5.1.4.5 Voice Recognition Board messages up ie to move forward

Fig 5.1.4.6 Voice Recognition Board messages eight ie to move right



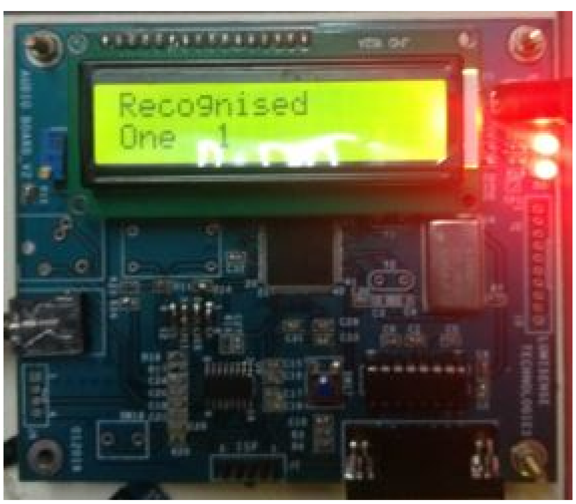


Fig 5.1.4.7 Voice Recognition Board messages one ie to move leftt

5.1.5 DC Motor

Fig 5.1.5.1 DC Motor moving

Figure No	Component Name	Component Description	Result
Fingerprint Scanner			
	Authorized Fingerprint	When authorized user uses fingerprint scanner then its shows finger print is verified.	Fingerprint scanner is properly in working condition
	Unauthorized Fingerprint	When unauthorized user uses fingerprint scanner then its shows finger print is not verified.	Fingerprint scanner is properly in working condition
Arduino Mega			
	Arduino Mega not connected with external power source	TX/RX inbuilt Led doesn't glow	Arduino Mega is inactive
	Arduino Mega connected with external power source	TX/RX inbuilt Led glow	Arduino Mega is active
Voice Recognition Board			
	Inbuilt commands are given	Recognize the command and gives the desire output	Voice recognition board working properly.

	Inbuilt commands are not given	Dont give any output	Voice recognition board working properly.
DC Motor			
	Not connected with PIC	Inactive	Not connected
	Connected with PIC and proper pins connected.	Active	Rotation of blades starts. Hence dc motor is in working properly

Table5.1.6.1:Unit testing and their results

5.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. After completing the connection and supplying the device with power all the individual components are functioning properly. Each individual component is providing with its respective values.

6. Results and Analysis

In this chapter we discuss about the results and outcomes of the test cases performed of each module and analyse them.

6.1 Output/Results

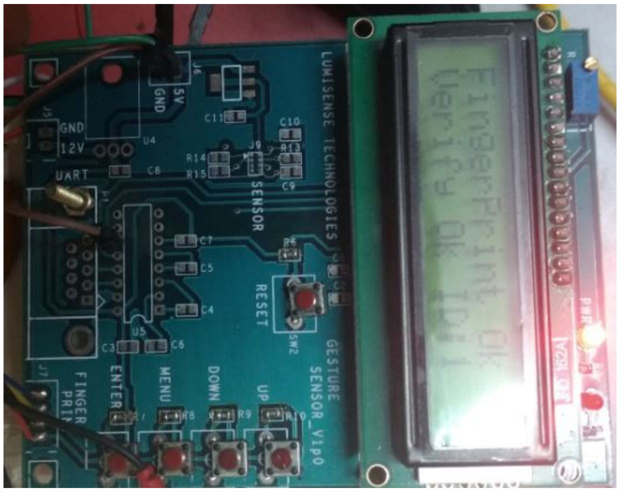


Fig 6.1.1 Fingerprint gets verified

Fig 6.1.2 Voice Recognition Board enables when finger print matches

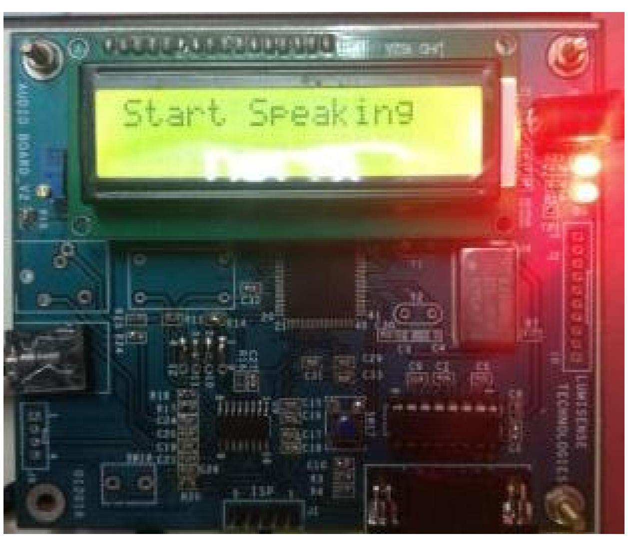


Fig 6.1.3 Voice Recognition Board ask user to give command to robot

.

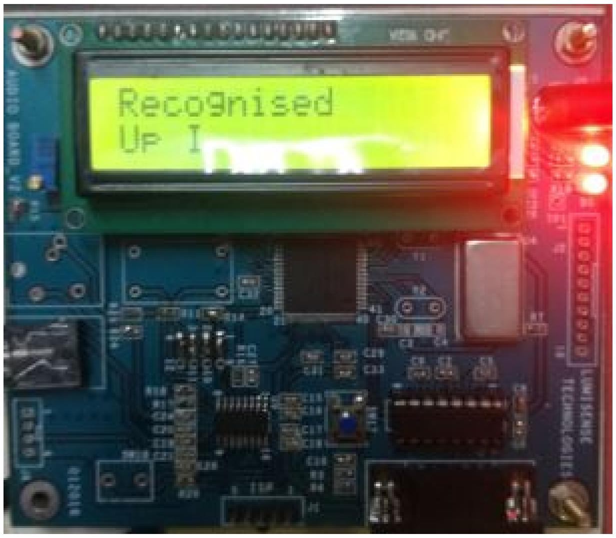


Fig 6.1.4 When user gives up as voice command robot gets start moving.

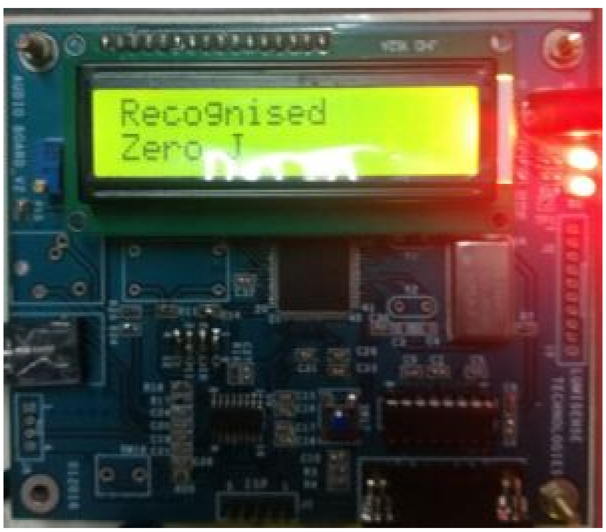


Fig 6.1.5 When user gives zero as voice command

Fig 6.1.6 Robot gets stop.

Fig 6.1.7 When user gives one as voice command.

Fig 6.1.8 Robot moves left

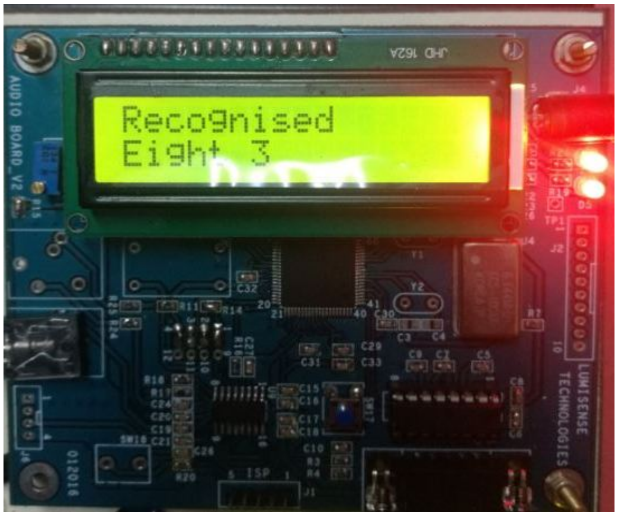


Fig 6.1.9 When user gives eight as voice command.

Fig 6.1.10 Robot moves right.

6.2 Result Analysis

S.No	Components	Quantity	Expected Output
1.	Fingerprint Scanner	1	Should matches the authorized fingerprint

2.	Voice Recognition board	1	Should give correct input to robot so that it moves correctly
3.	Motor driver	1	Motor should drive the motor coorectly.
4.	DC Motor	1	DC motor should run with Arduino Mega motor driver.
5.	Arduino Display Board	1	It should display correct output to user.
6.	Arduino Mega	1	It should manage all the connection properly.
7.	Battery	1	Battery should work when connected to Arduino and DC motors.

Table 6.2.1 Components and their outcomes

7. Conclusion

This report summarizes our final year project work done for the degree of B.Tech in Information Technology. The project has reached its conclusion and left the team with many lessons learned.

7.1 Important Accomplishments

The team strived to have a prototype to be proud of and demonstrate during the Final Review of the Project Work. The team accomplished this and was able to test the system well before in order to adjust controls. The team attributes this accomplishment to good time management and hard work put in by each member of the team and the guide throughout the project work.

7.2 Lessons Learned

Throughout this project the team learned several lessons about product development and project management. The first lesson learned was that it is important to manage time well and front load a project as much as possible. The team believes that they did a good job loading up work in the first month of the project developing an early prototype in February.

A lesson learned by the team members is how much effort goes into coordinating a team of people. The task of managing the project was larger than anticipated and as a result we all worked to assist each other in all areas of the project.

Additionally the team learned that it is important to develop a bill of materials early in the project so that all parts can be ordered in plenty of time for implementation. The most important lesson learned is to listen to the suggestions of your guide time to time. All the suggestions and changes made by our guide really helped us in building a successful project.

7.3 Possibility of Future Work

There are several areas that the team sees as potential development. One big change expected in coming years is the introduction of new technology for use in car security systems. Even more advanced is the face-recognition camera by Netatmo, also introduced in 2015. This camera can be programmed to recognize certain faces, like friends and family members, and will able to access the automobile or any robot. And if an unrecognized face arrives at your front door, you will also receive an alert and be able to watch a live feed from your smart phone.

This project proved to be a challenging, yet rewarding experience allowing us to gain a breadth of knowledge in both designing and implementing smart robot.

Though the prototype would need to be redesigned for mass use, it would certainly be a feasible and economical solution to anyone’s security and reconnaissance needs.

8. References

Journal Articles and Reports

1] A. J. Ijspeert, “Biorobotics: Using robots to emulate and investigate agile locomotion,” Science, vol. 346, no. 6206, pp. 196–203, Oct. 2014.

[2] Y. Kang, H. Kim, S. H. Ryu, N. L. Doh, Y. Oh, and B. J. You, “Dependable humanoid navigation system based on bipedal locomotion,” IEEE Trans. Ind. Electron., vol. 59, no. 2, pp. 1050–1060, Feb. 2012.

[3] N. Motoi, T. Suzuki, and K. Ohnishi, “A bipedal locomotion planning based on virtual linear inverted pendulum mode,” IEEE Trans. Ind. Electron., vol. 56, no. 1, pp. 54–61, Jan. 2009.

[4] K. Loffler, M. Gienger, F. Pfeiffer, and H. Ulbrich, “Sensors and control concept of a biped robot,” IEEE Trans. Ind. Electron., vol. 51, no. 5, pp. 972–980, Oct. 2004.

[5] S. Grillner, T. Deliagina, O. E. Manira, R. Hill, G. Orlovsky, P. Wall´en,

O. Ekeberg, and A. Lansner, “Neural networks that co-ordinate locomotion and body orientation in lamprey,” Trends Neurosci., vol. 18, no. 6, pp. 270 – 279, Jun. 1995.